LowResNLP 2025

Proceedings
of the 1st Workshop
on Advancing NLP for Low-Resource Languages

associated with

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

Edited by Ernesto Luis Estevanell-Valladares, Alicia Picazo-Izquierdo, Tharindu Ranasinghe,
Besik Mikaberidze, Simon Ostermann, Daniil Gurgurov, Philipp Müller,
Kurt Micallef, Claudia Borg, Michal Gregor and Marián Šimko

13 September, 2025
Varna, Bulgaria

The 1st Workshop
on Advancing NLP for Low-Resource Languages
Associated with the International Conference
Recent Advances in Natural Language Processing
RANLP'2025

PROCEEDINGS

Varna, Bulgaria
13 September 2025

Online ISBN 978-954-452-100-4

Designed by INCOMA Ltd.
Shoumen, BULGARIA

Preface

LowResNLP is a workshop dedicated to advancing NLP research for low-resource languages, fostering methods, resources, and evaluation practices that address the unique challenges faced by underrepresented languages.

This year we received **24 submissions**, of which **18 were accepted**: 9 papers for **oral + poster presentation** and 9 for **poster-only presentation**. We are excited by the variety and quality of contributions, which highlight new directions in cross-lingual transfer, data augmentation, model efficiency, and evaluation for low-resource settings.

We thank all authors for submitting, and we congratulate the accepted teams. We also express our gratitude to the program committee and our supporters for their valuable efforts and support.

The LowResNLP 2025 Organizers

Organizing Committee:

Ernesto Luis Estevanell-Valladares (University of Alicante, Spain; University of Havana, Cuba)

Alicia Picazo-Izquierdo (University of Alicante, Spain)

Tharindu Ranasinghe (Lancaster University, UK)

Besik Mikaberidze (Muskhelishvili Institute of Computational Mathematics, Georgian Technical University, Georgia)

Simon Ostermann (German Research Center for Artificial Intelligence, Germany)

Daniil Gurgurov (German Research Center for Artificial Intelligence, Germany)

Philipp Müller (German Research Center for Artificial Intelligence, Germany)

Kurt Micallef (University of Malta, Malta)

Claudia Borg (University of Malta, Malta)

Michal Gregor (KINIT, Slovakia)

Marián Šimko (KINIT, Slovakia)

Invited Speakers:

Jesujoba Oluwadara Alabi

Programme Committee:

Nora Aranberri (University of Basque Country)

Sudhansu Bala Das (School of Languages, Literatures & Cultures and Insight SFI Research Centre for Data Analytics, University of Galway, Ireland)

Ana-Maria Bucur (University of Bucharest)

Annie Lee En-Shiun (Ontario Tech University and University of Toronto)

Sofía García González (imaxin software, University of the Basque Country)

Albert Gatt (Utrecht University)

Teresa Lynn (Mohamed bin Zayed University of Artificial Intelligence)

Basab Nath (Assam University)

Patrizia Paggio (University of Malta)

Dhrubajyoti Pathak (National Forensic Sciences University)

Fabian Schmidt (University of Würzburg)

Marijn Schraagen (Utrecht University)

A. Seza Doğruöz (University of Ghent)

Marc Tanti (University of Malta)

Sunita Warjri (University of South Bohemia)

Table of Contents

Bridging the Gap: Leveraging Cherokee to Improve Language Identification for Endangered Iroquoian Languages

Liam Eggleston, Michael Cacioli, Jatin Sarabu, Kevin Zhu
Algoverse AI Research
kevin@algoverse.us

Abstract

Language identification is a foundational task in natural language processing (NLP), yet many Indigenous languages remain entirely unsupported by commercial language identification systems. In this study, we assess the performance of Google LangID on a 5k Cherokee dataset and find that every sentence is classified as "undetermined", indicating a complete failure to even misidentify Cherokee as another language. To further explore this issue, we manually constructed the first digitalized Northern Iroquoian dataset, consisting of 120 sentences across five related languages: Onondaga, Cayuga, Mohawk, Seneca, and Oneida. Running these sentences through Google LangID, we examine patterns in its incorrect predictions. To address these limitations, we train a random forest classifier to successfully distinguish between these languages, demonstrating its effectiveness in language identification. Our findings underscore the inadequacies of existing commercial language identification models for Indigenous languages and highlight concrete steps toward improving automated recognition of low-resource languages.

1 Introduction

Language identification is fundamental to natural language processing (Kargaran et al., 2023), enabling applications like machine translation, speech recognition, and text classification (Qi et al., 2019). While commercial language technologies such as Google's LangID perform well for high-resource languages, they provide no support for Native American languages (Caswell et al., 2020; Yang et al., 2025b,e). This lack of recognition contributes to digital marginalization and excludes speakers from technological advancements (Bali et al., 2019; Kukulska-Hulme et al., 2023). Cherokee, a Southern Iroquoian language, exemplifies this gap, as it remains computationally underrepresented despite active revitalization efforts

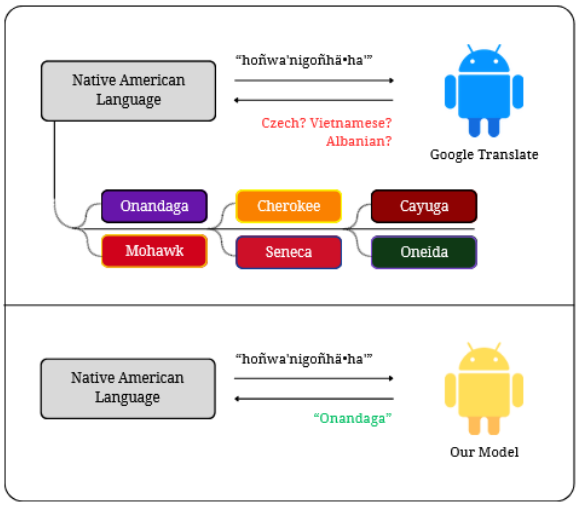


Figure 1: A stylized rendition of our language identification system for endangered Iroquoian languages.

(White, 1962; Peter and Hirata-Edds, 2006; Cushman, 2019).

To investigate this issue, we examined Google LangID's handling of Cherokee and five Northern Iroquoian languages, Onondaga, Cayuga, Mohawk, Seneca, and Oneida, using a manually curated dataset of 120 sentences evenly distributed across languages classes. While Cherokee was consistently misclassified as "undetermined", the other Northern Iroquoian languages were assigned unrelated languages. As shown in Figure 1, we then trained a random forest classifier on Cherokee and these misidentified languages, demonstrating that even with limited data, high classification accuracy is achievable. Our contributions include (1) a novel dataset, (2) an empirical evaluation of Google LangID's misclassification tendencies, and (3) an efficient classification model that outperforms existing approaches.

2 Related Work

Recent NLP research on Indigenous languages has increasingly focused on language identification,

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 1–6, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-001

cross-lingual generalization, and synthetic data generation to mitigate data scarcity. While modern LangID models support hundreds of languages (Kargaran et al., 2023; Milind Agarwal, 2023), they frequently overlook or fail for Indigenous languages due to insufficient training data (Cavalin et al., 2023). One promising approach is family-aware classification, where related languages are incorporated into training. Cavalin et al. (2023) demonstrated this by improving LangID performance for Brazilian Indigenous languages through linguistic family modeling. Similarly, leveraging phonological, morphological, and script-based cues has been proposed as a strategy for improving classification of Cherokee and Northern Iroquoian languages (Kargaran et al., 2023). However, Cherokee's unique syllabary introduces additional challenges compared to the Latin-based scripts used by its linguistic relatives.

Cross-lingual generalization offers a promising approach to improving LangID in low-resource settings. Multilingual models like mBERT can transfer knowledge across related languages (Pires et al., 2019), with pretraining on linguistically similar languages boosting classification accuracy (Bafna et al., 2023). While Cherokee belongs to the Southern Iroquoian branch (Zhang, 2022), it shares structural features with Northern Iroquoian languages, suggesting potential for generalization. However, differences in writing systems may hinder direct transfer, requiring transliteration or character-level modeling (Zhang et al., 2020). Given the scarcity of annotated data, synthetic techniques such as back-translation and morphological augmentation have been explored to enhance NLP models for endangered languages (Feldman and Coto-Solano, 2020; Zhang et al., 2020; Yang et al., 2025c,a). While synthetic data can improve classifier robustness, community validation remains crucial to mitigating risks associated with artificial augmentation (Zhang et al., 2022). Applied thoughtfully, these methods could strengthen language identification for Cherokee and Northern Iroquoian languages.

3 NatAm Language Landscape

The Cherokee language, known as Tsalagi Gawonihisdi (King, 1975), belongs to the Iroquoian language family and is classified under the Southern Iroquoian branch. As shown in Figure 2, it is the only surviving language of this branch (Rountree, 1987), with its closest linguistic relatives found in the Northern Iroquoian group, including Mohawk, Seneca, Oneida, Onondaga, and Cayuga. Linguistic evidence suggests that Cherokee diverged from Northern Iroquoian languages approximately 3,500 to 3,800 years ago (Barrie and Uchihara, 2019), leading to substantial differences in phonology, morphology, and writing systems. Unlike the Northern Iroquoian languages, which primarily rely on oral traditions and Latin-based orthographies (Birch, 2015), Cherokee developed a unique syllabary in the early 19th century (Cushman, 2012), further distinguishing it from its linguistic relatives.

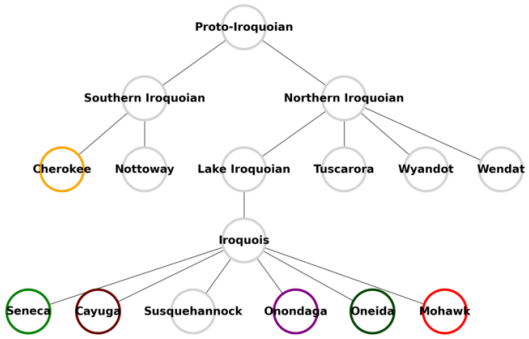


Figure 2: Language family tree for Proto-Iroquoian languages, with Cherokee, Seneca, Cayuga, Onondaga, Oneida and Mohawk highlighted through color.

Mohawk, Seneca, Oneida, Onondaga, and Cayuga, spoken in the northeastern United States and Canada, are members of the Northern Iroquoian branch and share many grammatical and phonological features. Mohawk, one of the most widely spoken Northern Iroquoian languages (Hoover, 1992), has benefited from revitalization programs and digital resources. Seneca and Cayuga, though critically endangered, continue to be taught in community-based initiatives (Chafe, 2015; Dyck and Kumar, 2012). Oneida and Onondaga, while also endangered, have seen growing interest in language preservation efforts through educational programs (Lu et al., 2024; Michelson, 2021). Despite their historical and linguistic connections, these languages exhibit distinct phonetic and syntactic structures (Kilarski, 2021), which may contribute to challenges in language classification. Furthermore, all Iroquoian languages have faced severe endangerment due to colonization and language suppression policies (Richter, 2011), necessitating ongoing revitalization efforts.

4 Data

To assess Google LangID's performance on Cherokee and other Northern Iroquoian languages, we manually collected text samples from publicly available sources[1]. For Cherokee, we were able to refer to an existing 5k dataset (Zhang et al., 2020). Given the scarcity of textual data for the other Northern Iroquoian languages, we manually curated our own digitalized dataset with community-driven language archives, linguistic documentation projects, and publicly available transcripts of Indigenous language programs. Each language was represented by about 20 sentences carefully selected to reflect a range of grammatical structures and vocabulary diversity.

Our decision to rely on manually curated data was driven by the lack of large-scale, digitized corpora for these languages. Automatic web scraping approaches proved ineffective due to the limited online presence of Indigenous languages and difficulty in accurately identifying them, necessitating a more targeted approach to ensure linguistic accuracy and representativeness. Additionally, we prioritized sources produced or validated by native speakers to maintain authenticity and avoid potential biases introduced by machine-generated translations. This novel dataset serves as a foundational resource for evaluating LangID models on Iroquoian languages and underscores the broader challenges of building NLP tools for endangered languages.

5 Language Identification

5.1 Google LangID

To evaluate Google LangID's handling of Cherokee, we passed the 5k Cherokee dataset through the Google Translate API. Surprisingly, every sentence was classified as *undetermined*, meaning the system did not even *attempt* to associate Cherokee with any known language. While Google LangID does not officially support Cherokee, it should at least misidentify it rather than fail to classify it altogether. Prior research on low-resource language identification has shown that unsupported languages are typically misclassified as typologically or phonetically similar ones. For instance, in a recent study on Navajo (Yang et al., 2025d), a 10k dataset was run through Google LangID, and while the results were incorrect, each sentence was still assigned to an existing language. The fact that Cherokee received no such assignment suggests a fundamental failure—not just in recognizing the language, but in engaging with the data at all.

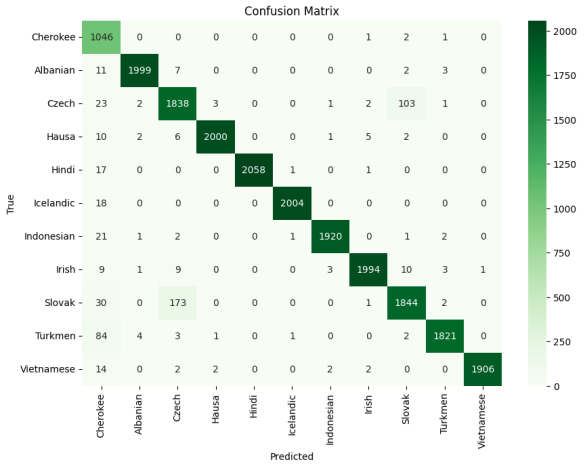


Figure 3: Classification results for Cherokee and 10 other languages, presented as a confusion matrix.

To further investigate, we ran our manually curated dataset of 120 Northern Iroquoian sentences through Google LangID. Unlike Cherokee, these sentences were assigned specific, though incorrect, language labels, indicating that the system at least attempted classification. This stark contrast in performance underscores a deeper issue; while many Indigenous languages are misidentified, Cherokee is uniquely *absent* from the model's processing pipeline, raising concerns about how commercial language technologies handle languages with distinct scripts, such as the Cherokee syllabary.

5.2 Classifier

To address the shortcomings of existing language identification models for Indigenous languages, we developed a custom classifier to distinguish Cherokee from other languages in our dataset. Given the limited availability of labeled data, we selected a Random Forest classifier (Hastie et al., 2009) for its robustness, interpretability, and effectiveness in handling small datasets with high-dimensional features. We employed a TF-IDF vectorizer to transform text into numerical representations, capturing key lexical distinctions. Tokenization was performed at the word level, and the feature space was restricted to the 5,000 most frequent terms to balance specificity and generalization.

The dataset included Cherokee alongside the ten most commonly misidentified languages, such as Albanian, Czech, Hausa, Hindi, Icelandic, Indone-

[1]Full citations are included in the GitHub.

Language	Precision	Recall	F1-Score
Cherokee	0.82	1.00	0.90
Albanian	1.00	0.99	0.99
Czech	0.90	0.93	0.92
Hausa	1.00	0.99	0.99
Hindi	1.00	0.99	1.00
Icelandic	1.00	0.99	0.99
Indonesian	1.00	0.99	0.99
Irish	0.99	0.98	0.99
Slovak	0.94	0.90	0.92
Turkmen	0.99	0.95	0.97
Vietnamese	1.00	0.99	0.99
Accuracy		0.97	
Macro Avg	0.97	0.97	0.97
Weighted Avg	0.97	0.97	0.97

Table 1: Multi-classifier Performance Metrics.

sian, Irish, Slovak, Turkmen, and Vietnamese. Text samples were manually curated and preprocessed to remove extraneous whitespace before vectorization. A stratified 80-20 train-test split ensured balanced representation across all classes. Training was conducted with 100 decision trees, using a fixed random state for reproducibility. Evaluation metrics (precision, recall, and F1-score) demonstrated strong differentiation between Cherokee and the other languages, though minor misclassifications occurred, particularly among typologically similar languages. The confusion matrix in Figure 3 highlights these cases, emphasizing the challenge of distinguishing languages with overlapping linguistic structures. The effectiveness of TF-IDF features in capturing distinguishing characteristics while filtering out noise from infrequent words is further reflected in Table 1.

Further analysis of the model's binary classification performance in Table 2 shows high accuracy in distinguishing Cherokee from all other languages. The precision and recall scores confirm the classifier's reliability in identifying Cherokee while correctly classifying non-Cherokee languages. Our results demonstrate that even with limited training data, a random forest classifier can effectively differentiate Indigenous from non-Indigenous languages, addressing gaps in commercial language identification. Future work could expand the dataset through community-driven contributions, incorporate additional Indigenous languages, and refine feature selection to enhance classification. Exploring deep learning approaches may further improve performance, fostering the development of more inclusive NLP tools for endangered languages.

Class	Precision	Recall	F1-Score
Cherokee	1.00	0.82	0.90
Non-Cherokee	0.99	1.00	1.00
Accuracy		0.99	
Macro Avg	1.00	0.91	0.95

Table 2: Binary Classification Performance Metrics.

6 Future Work

Future research will include interviews with Indigenous community members to gain cultural insights into language classification challenges. We have already scheduled two interviews with an Omaha Tribe member and a member of the Okanagan/Wenatchi community, ensuring direct engagement with native speakers. Expanding the dataset to incorporate additional Indigenous languages and exploring deep learning models will further improve classification accuracy (Alvarez et al., 2025). Additionally, integrating phonetic and morphological features will enhance model interpretability, while ethical considerations will guide meaningful collaboration with Indigenous communities for validation and tool development. These efforts aim to create more inclusive and effective language identification tools that actively support Indigenous language preservation.

7 Conclusion

This study highlights the severe shortcomings of commercial language identification systems for Indigenous languages, exemplified by Google LangID's failure to classify Cherokee—even incorrectly. While other Northern Iroquoian languages received misidentifications, Cherokee was uniquely ignored, raising concerns about how commercial models handle languages with distinct scripts. To address this gap, we developed a random forest classifier that effectively differentiates Cherokee, demonstrating that even with limited data, accurate classification is achievable. Our findings underscore the need for more inclusive NLP tools that support endangered languages. **We call upon the NLP community to move beyond discussion and take concrete steps, whether by expanding datasets or collaborating with Indigenous speakers, to ensure that these languages are not just studied, but actively supported.**

Limitations

While our study provides valuable insights into the deficiencies of commercial LangID models for Cherokee and Northern Iroquoian languages, it is constrained by the small dataset size and the absence of native speaker validation. Additionally, our classifier's effectiveness may not extend to other underrepresented Indigenous languages with different linguistic structures. Further research should explore larger datasets, multimodal approaches, and direct collaboration with Indigenous speakers to improve the accuracy and ethical implementation of language identification systems.

Ethics Statement

Our study prioritizes ethical data collection and representation of Indigenous languages. We sourced data only from publicly available and community-approved resources, ensuring that no proprietary or culturally sensitive materials were used without consent. Additionally, we acknowledge the historical and ongoing marginalization of Indigenous languages in NLP and aim to contribute to language preservation rather than commodification. Future work should actively involve Indigenous communities in data collection and validation to ensure their agency in technological advancements. In the spirit of transparent and ethical research, samples of our data and code has been made available at (https://github.com/Cherokee-Project/Classifier).

References

Jesus Alvarez, Daua Karajeanes, Ashley Prado, John Ruttan, Ivory Yang, Sean O'Brien, Vasu Sharma, and Kevin Zhu. 2025. Advancing uto-aztecan language technologies: A case study on the endangered comanche language. In *Proceedings of the Fifth Workshop on NLP for Indigenous Languages of the Americas (AmericasNLP)*, pages 27–37.

Niyati Bafna, Cristina España-Bonet, Josef Van Genabith, Benoît Sagot, and Rachel Bawden. 2023. Cross-lingual strategies for low-resource language modeling: A study on five indic dialects. In *18e Conférence en Recherche d'Information et Applications–16e Rencontres Jeunes Chercheurs en RI–30e Conférence sur le Traitement Automatique des Langues Naturelles–25e Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues*, pages 28–42. ATALA.

Kalika Bali, Monojit Choudhury, Sunaya Sitaram, and Vivek Seshadri. 2019. Ellora: Enabling low resource languages with technology. In *Proceedings of the 1st International Conference on Language Technologies for All*, pages 160–163.

Michael Barrie and Hiroto Uchihara. 2019. Iroquoian languages. In *The Routledge handbook of North American languages*, pages 424–451. Routledge.

Jennifer Birch. 2015. Current research on the historical development of northern iroquoian societies. *Journal of Archaeological Research*, 23:263–323.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608.

Paulo Cavalin, Pedro Domingues, Julio Nogima, and Claudio Pinhanez. 2023. Understanding native language identification for brazilian indigenous languages. In *Proceedings of the Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP)*, pages 12–18.

Wallace L Chafe. 2015. *A grammar of the Seneca language*, volume 149. Univ of California Press.

Ellen Cushman. 2012. *The Cherokee syllabary: Writing the people's perseverance*, volume 56. University of Oklahoma Press.

Ellen Cushman. 2019. Language perseverance and translation of cherokee documents. *College English*, 82(1):115–134.

Carrie Dyck and Ranjeet Kumar. 2012. A grammar-driven bilingual digital dictionary for cayuga (iroquoian). *Dictionaries: Journal of the Dictionary Society of North America*, 33(1):179–204.

Isaac Feldman and Rolando Coto-Solano. 2020. Neural machine translation models with back-translation for the extremely low-resource indigenous language bribri. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3965–3976.

Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. Random forests. *The elements of statistical learning: Data mining, inference, and prediction*, pages 587–604.

Michael L Hoover. 1992. The revival of the mohawk language in kahnawake. *Canadian Journal of Native Studies*, 12(2):269–287.

Amir Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schütze. 2023. Glotlid: Language identification for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218.

Marcin Kilarski. 2021. Sound systems in iroquoian languages. In *A History of the Study of the Indigenous Languages of North America*, pages 131–172. John Benjamins Publishing Company.

Duane Harold King. 1975. *A grammar and dictionary of the Cherokee language.* University of Georgia.

Agnes Kukulska-Hulme, Ram Ashish Giri, Saraswati Dawadi, Kamal Raj Devkota, and Mark Gaved. 2023. Languages and technologies in education at school and outside of school: Perspectives from young people in low-resource countries in africa and asia. *Frontiers in Communication*, 8:1081155.

Yanfei Lu, Patrick Littell, and Keren Rice. 2024. Empowering oneida language revitalization: Development of an oneida verb conjugator. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5757–5767.

Karin Michelson. 2021. A reference grammar of the onondaga language.

Antonios Anastasopoulos Milind Agarwal, Md. Mahfuz Ibn Alam. 2023. Limit: Language identification, misidentification, and translation using hierarchical models in 350+ languages. In *EMNLP 2023*.

Lizette Peter and Tracy E Hirata-Edds. 2006. Using assessment to inform instruction in cherokee language revitalisation. *International Journal of Bilingual Education and Bilingualism*, 9(5):643–658.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Zhaodi Qi, Yong Ma, and Mingliang Gu. 2019. A study on low-resource language identification. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1897–1902. IEEE.

Daniel K Richter. 2011. *The ordeal of the longhouse: the peoples of the Iroquois League in the era of European colonization*. UNC Press Books.

Helen C Rountree. 1987. The termination and dispersal of the nottoway indians of virginia. *The Virginia Magazine of History and Biography*, 95(2):193–214.

John K White. 1962. On the revival of printing in the cherokee language. *Current Anthropology*, 3(5):511–514.

Ivory Yang, Xiaobo Guo, Yuxin Wang, Hefan Zhang, Yaning Jia, William Dinauer, and Soroush Vosoughi. 2025a. Recontextualizing revitalization: A mixed media approach to reviving the nüshu language. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Ivory Yang, Weicheng Ma, Carlos Guerrero Alvarez, William Dinauer, and Soroush Vosoughi. 2025b. What is it? towards a generalizable native american language identification system. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 105–111.

Ivory Yang, Weicheng Ma, and Soroush Vosoughi. 2025c. Nüshurescue: Reviving the endangered nüshu language with ai. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7020–7034.

Ivory Yang, Weicheng Ma, Chunhui Zhang, and Soroush Vosoughi. 2025d. Is it Navajo? accurate language detection for endangered athabaskan languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 277–284.

Ivory Yang, Chunhui Zhang, Yuxin Wang, Zhongyu Ouyang, and Soroush Vosoughi. 2025e. Visibility as survival: Generalizing nlp for native alaskan language identification. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6965–6979.

Bryan Zhang. 2022. Improve MT for search with selected translation memory using search signals. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 123–131, Orlando, USA. Association for Machine Translation in the Americas.

Shiyue Zhang, Ben Frey, and Mohit Bansal. 2022. How can nlp help revitalize endangered languages? a case study and roadmap for the cherokee language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1529–1541.

Shiyue Zhang, Benjamin Frey, and Mohit Bansal. 2020. Chren: Cherokee-english machine translation for endangered language revitalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 577–595.

Building a Lightweight Classifier to Distinguish Closely Related Language Varieties with Limited Supervision: The Case of Catalan vs Valencian

Raúl García Cerdá **María Miró Maestre** **Miquel Canal**
University of Alicante
{raul.gc, maria.miro, mikel.canal}@ua.es

Abstract

Dialectal variation among closely related languages poses a major challenge in low-resource NLP, as their linguistic similarity increases confusability for automatic systems. We introduce the first supervised classifier to distinguish standard Catalan from its regional variety Valencian. Our lightweight approach finetunes a RoBERTa-base model on a manually curated corpus of 20 000 sentences—without any Valencian-specific tools—and achieves 98 % accuracy on unseen test data. In a human evaluation of 90 mixed-variety items per reviewer, acceptance rates reached 96.7 % for Valencian and 97.7 % for Catalan (97.2 % overall). We discuss limitations with out-of-distribution inputs and outline future work on confidence calibration and dialect-aware tokenization. Our findings demonstrate that high-impact dialect classification is feasible with minimal resources.

1 Introduction

Linguistic Background. Valencian is the variety of Catalan spoken in the Valencian Community and is officially recognized as one of its co–official languages. Linguistically, the Acadèmia Valenciana de la Llengua (AVL) "formally acknowledges Valencian as one variant of the common Catalan language" (European Language Equality (ELE), 2022; Acadèmia Valenciana de la Llengua, 2022). However, due to historical and political factors—such as the repression of Catalan during the Franco regime—Valencian has often occupied a minoritized position, surviving mainly in informal domains and facing strong pressure from Spanish (European Language Equality (ELE), 2022). This sociopolitical context is reflected in technology: **Google Translate** and most commercial voice assistants do not distinguish Valencian (they offer only "Catalan") (European Language Equality (ELE), 2022), and Microsoft Office requires a separate "Catalan (Valencian)" pack maintained by Softcatalà (Softcatalà, 2018). The Valencian variety of Catalan is commonly perceived as a regional dialect rather than a distinct linguistic entity, which has led to its underrepresentation in natural language processing (NLP) resources. Despite being an official language in the Valencian Community, Valencian lacks dedicated tools such as lemmatizers, spell checkers, or machine translation systems that treat it independently from standard Catalan. This scarcity of resources positions Valencian as a low-resource language variant in practical computational terms. Although Catalan has benefited from recent advances in language modeling and the availability of large-scale corpora, similar efforts for Valencian are virtually nonexistent.

In this paper, our aim is to contribute to the development of dialect-specific resources by presenting a lightweight binary text classifier capable of distinguishing between standard Catalan and Valencian. We train our model using manually curated data from official public sources and demonstrate that it is possible to obtain accurate and robust results even with limited supervision and minimal preprocessing tools. *To the best of our knowledge, this is the first work to formulate and evaluate the task of discriminating between standard Catalan and Valencian as a supervised classification problem.* Our work is framed within the broader context of dialectal NLP and highlights both the technical challenges and sociolinguistic implications of computationally differentiating closely related language varieties.

2 Related Work

Linguistic features. Catalan and Valencian are Romance languages derived from Vulgar Latin (Martines, 2024), sharing many features with Spanish and French but also showing systematic differences. One example is the present subjunctive: Catalan uses endings in *-i* (e.g., *canti*) (d'Estudis Catalans, 2022), whereas Valencian prefers *-e*

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 7–11, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-002

(e.g., *cante*) (Acadèmia Valenciana de la Llengua, 2006). Another contrast is in feminine possessive pronouns: Catalan *meva/teva/seva* vs. Valencian *meua/teua/seua* (Institut d'Estudis Catalans, 2022; Real Acadèmia de Cultura Valenciana, Secció de Llengua i Lliteratura Valencianes, 2025). Beyond morphology, numerous studies have documented lexical divergences between the two varieties (Wheeler, 2005; Marzà et al., 2006; Lledó, 2011), and ongoing online projects aim to compile them systematically (Idioma Valenciano, 2025; Acadèmia Valenciana de la Llengua, 2025).

Existing resources. Most NLP resources for Catalan do not explicitly handle Valencian. Spellcheckers such as Softcatalà provide a unified Catalan dictionary with a "Valencià" variant (Softcatalà, 2018), and LanguageTool or the SALT platform offer only basic configurations. Open-source MT systems (Apertium, Politraductor) adapt Valencian lexicon, while commercial engines (DeepL, Google Translate) collapse Valencian into Catalan (European Language Equality (ELE), 2022). Morphological analyzers like FreeLing or spaCy are trained for Catalan and must be reused for Valencian, which can miss regional features. State-of-the-art PLMs such as CALBERT and RoBERTa (Projecte AINA) are trained on broad Catalan data with no explicit Valencian component (VIVES, 2025), and available Valencian corpora (DOGV, À Punt Mèdia) remain relatively small for standalone training (European Language Equality (ELE), 2022; VIVES, 2025).

Dialect identification and lightweight models. Dialect identification has been studied extensively in other language pairs but not for Catalan/Valencian. The DSL shared tasks included Czech vs. Slovak and Brazilian vs. European Portuguese, achieving high accuracy on newswire (Zampieri et al., 2014, 2015). More recently, (Preda et al., 2024) revisited pt-BR vs. pt-PT with updated methods, and (Zampieri et al., 2020) provide a survey of techniques and pitfalls in similar-language discrimination. Lightweight fine-tuning has also proven effective in low-resource dialectal NLP: BERT on Arabic tweets (Mansour et al., 2020), AfriBERTa on African languages (Ogueji et al., 2021), and small multilingual models like mBERT or XLM-R that often outperform larger LLMs in limited-data regimes (Gurgurov et al., 2025).

3 Corpus

Because no labeled dataset exists to distinguish Catalan and Valencian, we compiled a new balanced corpus of 20 000 sentences (10 000 per variety). Sources included the Valencian government gazette (DOGV) and the À Punt Media portal for Valencian, and the Catalan government portal (gencat.cat) and the 3Cat/324 news site for Catalan.

We preserved all original tokens (including dates, headers, codes) to retain contextual cues, and only applied lowercasing. Sentence segmentation was carried out with regex rules plus manual review. Each sentence received a binary label: Valencian (1) or Catalan (0). The corpus was split into 80% training (16 000 sentences) and 20% test (4 000), ensuring class balance.

Data collection. We assembled 20 000 sentences (10 000 per class) from public institutional and media sources: the DOGV (`https://dogv.gva.es`) and À Punt Mèdia (`https://www.apuntmedia.es`) for Valencian, and the Catalan government's public portal gencat.cat (`https://web.gencat.cat`) (the Catalan equivalent of DOGV) and the 3Cat/324 news site (`https://www.3cat.cat/324`) for Catalan. All texts retained original metadata (dates, headers, codes) to leverage contextual cues.

Prior work has shown that case-sensitive models retain useful signals from capitalization and diacritics without loss in accuracy (e.g., BETO vs. lowercase BETO in Spanish; (Cóster and Martínez, 2021)), and that preserving punctuation and numerals maintains structural cues crucial for text classification (HaCohen-Kerner and Levin, 2020).

Preprocessing and labeling. Preprocessing follows Section 3: we only apply lowercasing. Sentence segmentation uses regex rules with manual review. Labels and the 80/20 split are as in Section 3.

4 Methodology

Model. We fine-tune RoBERTa-base (Liu et al., 2019), pre-trained on Catalan (Projecte AINA), for binary classification.

Training Setup.

- **Optimizer:** AdamW (weight decay 0.01).
- **Learning rate:** 2×10^{-5}, linear warmup (500 steps), total 3 epochs.
- **Batch size:** 16.
- **Scheduler:** Linear decay to zero.
- **Early stopping:** validation loss, patience = 1 epoch.

Input Representation. We tokenized with the standard RoBERTa tokenizer and truncated or padded sentences to 128 tokens. All other preprocessing followed the corpus description in Section 3.

Implementation. The experiments were run with HuggingFace Transformers v4.5.1 and PyTorch v1.10.1 on a single NVIDIA T4 GPU (16GB) provided via Google Colab, with 25GB host RAM available. We freeze the first 6 encoder layers for the first epoch to stabilize training, then unfreeze all layers.

Evaluation Protocol and Human Setup. Automatic metrics (accuracy, precision, recall, F_1) are computed on the held-out test set of 4 000 sentences (20% of the 20k corpus). In addition, we performed a human evaluation on a separate pool of 6 000 sentences (3 000 per variety). Following a statistically representative sampling procedure (Barros et al., 2021; Vázquez et al., 2010), we sampled up to 90 sentences per reviewer for each variety. We aimed for a balanced mix of error and correct cases (up to 45 each), though the exact proportions varied due to random sampling. To increase cross-variety exposure, some items came from the opposite class. A native Valencian speaker (bilingual in Spanish) annotated the Valencian-focused set, and a native Catalan speaker (also bilingual in Spanish) annotated the Catalan-focused set. We report per-class acceptance rates (96.7% for Valencian, 97.7% for Catalan) as the proportion of model predictions confirmed by humans.

5 Experiments and Results

Following the training setup described in Section 4, we fine-tuned RoBERTa-base for three epochs. Table 1 reports automatic test metrics and human acceptance rates.

Table 1: Automatic test metrics (n=4,000) and human acceptance (n=6,000).

	Acc (%)	Prec	Rec	F1
Automatic (overall)	98.0	0.978	0.976	0.977
Valencian (auto)	–	0.980	0.982	0.981
Catalan (auto)	–	0.982	0.980	0.981
Human acceptance				
Valencian		96.7%		
Catalan		97.7%		
Overall		97.2%		

The automatic confusion matrix (Table 2) remains unchanged, showing false positives and negatives below 2%.

	Predicted Catalan	Predicted Valencian
True Catalan	1,960	40
True Valencian	35	1,965

Table 2: Confusion Matrix (n=4,000).

6 Discussion

While our model achieves 98% automatic accuracy, human acceptance rates confirm high reliability across both varieties: it correctly labels 96.7% of Valencian sentences and 97.7% of Catalan ones, for an overall 97.2% acceptance. This consistency suggest robust performance, though further analysis is needed to ensure the model does not over-rely on contextual metadata and to better handle challenging or ambiguous cases.

7 Conclusions and Future Work

We have presented a lightweight classifier capable of distinguishing between standard Catalan and Valencian using minimal data and without dialect-specific tools. Trained in 20,000 sentences with contextual metadata, our model achieves automatic accuracy 98%. Future directions include:

- Incorporating human acceptance rates for confidence calibration.
- Extending training to informal varieties (e.g., social media dialects).
- Developing a dialect-aware tokenizer to better handle metadata and numerals.

All trained model checkpoints and associated code will be released upon acceptance. The resources will be accessible at `https://github.com/leurz/modelo-catalan-valenciano`.

Limitations

Our study is limited to formal, institutional sources; generalisation to informal or noisy domains (e.g., social media) remains untested. In addition, we did not run ablations to disentangle linguistic features from metadata cues, and comparisons to alternative classifiers (e.g., SVMs or multilingual BERT) remain for future work. These aspects should be addressed to fully understand the robustness and portability of our approach.

Acknowledgments

This research has been conducted within the project *The limits and future of data-driven approaches: a comparative study of deep learning, knowledge-based and rule-based models and methods in Natural Language Processing*, funded by the Generalitat Valenciana under the CIDEGENT programme (Exp. CIDEXG/2023/13).

All authors are supported through this grant and are members of the Research Group on Natural Language Processing (GPLSI), University of Alicante. We thank the anonymous reviewers for their constructive feedback.

References

Acadèmia Valenciana de la Llengua. 2006. Gramàtica normativa valenciana. https://www.avl.gva.es/va/gramatica-normativa-valenciana.

Acadèmia Valenciana de la Llengua. 2022. Statement on valencian as a variant of catalan. https://avl.gva.es.

Acadèmia Valenciana de la Llengua. 2025. L'avl i la universitat d'alacant col·laboren en l'atles lingüístic valencià. https://www.avl.gva.es/lacademia-valenciana-de-la-llengua-i-la-universitat-dalacant-collaboren-en-el-projecte-atles-linguistic-valencia/.

Cristina Barros, Manuel Vicente, and Elena Lloret. 2021. To what extent does content selection affect surface realization in the context of headline generation? *Computer Speech & Language*, 67:101179.

Institut d'Estudis Catalans. 2022. Gramàtica essencial de la llengua catalana. https://geiec.iec.cat/text/5.3.3.

Adam Cóster and Paula Martínez. 2021. Evaluating the impact of lowercasing on spanish bert (beto). In *IberLEF 2021*.

European Language Equality (ELE). 2022. Report on the catalan language in the digital age. Technical report, ELE Project, Deliverable D1.6.

Stefan Gurgurov, Anna Ivanov, and Pavel Petrov. 2025. Small models, big impact: Adaptation of small multilingual language models for low-resource languages. *arXiv*, 2502.10140.

Yaron HaCohen-Kerner and Boris Levin. 2020. On the role of punctuation in text classification. *Computational Linguistics*, 46(1):1–22.

Idioma Valenciano. 2025. Listado de palabras diferentes en valenciano y catalán. https://www.idiomavalenciano.com/listado-palabras-diferentes-valenciano-catalan.html.

Institut d'Estudis Catalans. 2022. Gramàtica bàsica i d'ús de la llengua catalana. https://gbu.iec.cat/text/14.4.

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.

Miquel Àngel Lledó. 2011. The independent standardization of valencian: From official use to underground. *Handbook of language and ethnic identity: The success-failure continuum in language and ethnic identity efforts*, 2:336–348.

Walid Mansour, Rafid Fakiyurd, and Ammar Farahat. 2020. Arabic dialect identification using bert fine-tuning. In *Proceedings of the 6th Workshop on Arabic Natural Language Processing (WANLP)*.

Josep Martines. 2024. History of the catalan lexicon. In *Oxford Research Encyclopedia of Linguistics*. Oxford University Press.

Anna Marzà, Frederic Chaume, Glòria Torralba, and Ana Alemany. 2006. The language we watch: an approach to the linguistic model of catalan in dubbing. In *Mercator Media Forum*, volume 9, pages 14–25. University of Wales Press.

Collins Ogueji, Rena Mika, and Temitope Adebowale. 2021. Small data? no problem! exploring the viability of pretrained multilingual models for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Diana Preda et al. 2024. Distinguishing between european and brazilian portuguese. In *Proceedings of PROPOR 2024*.

Real Acadèmia de Cultura Valenciana, Secció de Llengua i Lliteratura Valencianes. 2025. Diccionari general de la llengua valenciana. https://diccionari.llenguavalenciana.com/general/consulta/possessius.

Softcatalà. 2018. Catalan spell-checker dictionaries, including valencian variant. https://softcatala.org.

Yoselyn G. Vázquez, Francisco A. Orquín, Antonio M. Guijarro, and Sergio V. Pérez. 2010. Integración de recursos semánticos basados en wordnet. *Procesamiento del Lenguaje Natural*, 45:161–168.

Projecte VIVES. 2025. Plan for valencian language technologies. Data-gathering for speech and text corpora.

Max W Wheeler. 2005. *The phonology of Catalan*. OUP Oxford.

Marcos Zampieri, Preslav Nakov, Nikola Ljubešić, Jörg Tiedemann, and Shervin Malmasi. 2020. Natural language processing for similar languages, varieties and dialects: a survey. *Natural Language Engineering*, 26(6):695–717.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A report on the dsl shared task 2014. In *Proceedings of VarDial Workshop @ COLING*.

Marcos Zampieri et al. 2015. Overview of the dsl shared task 2015. In *Proceedings of LT4VarDial Workshop @ RANLP*.

A thresholding method for Improving translation Quality for Indic MT task

Sudhansu Bala Das **Leo Raphael Rodrigues**
Tapas Kumar Mishra **Bidyut Kumar Patra**
School of Languages, Literatures & Cultures;
Insight SFI Research Centre for Data Analytics, University of Galway, Ireland
National Institute of Technology (NIT) Rourkela, India
Indian Institute of Technology (BHU) Varanasi , India
baladas.sudhansu@gmail.com, leoraphaelro5@gmail.com, mishrat@nitrkl.ac.in, bidyut.cse@itbhu.ac.in

Abstract

Machine Translation (MT) automatically converts text from one language to another. End-to-end Neural Machine Translation (NMT) performs best when trained on large, clean parallel data. In the Indic setting, the SAMANANTAR corpus is the largest public resource, but its heterogeneous sources introduce noise such as misalignments, duplicates, boilerplate, and mistranslations, that may limit model quality. We present an exploratory, data-centric experiment: can a simple, validation-calibrated filter that removes non-parallel pairs *before* training improve Indic NMT? Our filter scores each pair with sentence-level Bilingual Evaluation Understudy (BLEU) and discards those below a single cutoff per direction $\tau = B_{\text{dev}}/4$ (with B_{dev} the corpus-level BLEU on the validation set). We do not assume this rule is optimal or universally applicable; instead, we assess its effect on two Indic language (IL) pairs—Hindi–English (HIN–ENG) and Odia–English (ODI–ENG)—and three training sizes (Full, Quarter, fixed 500k). Across BLEU, METEOR, and RIBES, the filtered data yield consistent gains, and IL→ENG directions outperform ENG→IL even when trained on the same data.

1 Introduction

Languages are vital for communication between individuals, groups, and nations (Das et al., 2025a) (Dalai et al., 2024). With increased global interdependence and interactions between cultures, machine translation (MT) is becoming more essential for promoting communication as well as fostering collaboration among individuals who speak various languages. Nonetheless, accomplishing precise and intact translations is still a crucial challenge when the dataset comes into play. A "dataset" is an extensive database of bilingual or multilingual texts utilized as training data for MT systems. It consists of pairs of interpretations from the source language to the target language. To examine these datasets, different MT models are used, which learn the textual structures and patterns in order to do the translation more accurately. However, the quality of the dataset used to train the MT models has a significant effect on the precision as well as the fluency of the output. When the dataset used for training is of poor quality, with inaccuracies, inconsistencies, or irrelevant data, then the MT system's performance will suffer, resulting in poor translations or even nonsensical outputs. Hence, the massive qualitative dataset is important for the MT system.

In a multilingual country such as India, the availability of massive, high-quality datasets for Indian languages (ILs) is important for establishing an effective MT system (Dalai et al., 2023), (Das et al., 2025b), (Kodati and Tene, 2025b). But it is observed that several ILs have constrained or inadequate datasets, which makes it challenging to establish robust MT models for ILs. However, researchers have come up with a massive parallel dataset collection for ILs, i.e., the Samanantar Dataset (Ramesh et al., 2022). Nevertheless, as this dataset has been collected from a variety of sources, it consists of an immense quantity of noise or wrong translations. The existence of noise in the dataset has an adverse effect on the efficiency and precision of the MT task. As a result, no MT models on ILs can generate flawless translations (precisely convey the meaning and actual translation from source to target language).

Many researchers (Bala Das et al., 2024),

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 12–20, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-003

(Kodati and Tene, 2025a), working with ILs have recently built various MT systems and models to experiment with various approaches to translate ILs to ENG and vice versa; however, they are still unable to achieve correct translation. The translation quality of models built on ILs can vary according to their dataset quality and quantity. Therefore, the individual relevance of the quality and quantity of the dataset during training remains an open question. In order to encourage the enhancement of MT for ILs, the performance evaluation of the translation quality of machine-generated output becomes increasingly important. Two ILs, such as Hindi (HIN) and Odia (ODI), from Samanantar datasets are taken into consideration for the experiment in this paper in order to eliminate and verify the impact of inaccurate and dissimilar translations in translation models. The motivation for selecting these languages is due to the fact that Odia has proportionally fewer parallel sentence pairings than Hindi (in the Samanantar Dataset), which is linked to a bigger dataset. The different representation sizes of these languages in the dataset led to their selection. The paper also proposes a novel technique to remove incorrect or dissimilar translations from the dataset and checks its effect on MT tasks. It also investigates how different dataset divisions (into multiple subsets) influence the translation quality. All the translations are evaluated using standard evaluation metrics. A baseline system using NMT models for 2 ILs is developed and examined in this paper. This paper is arranged as follows: Section 2 focuses on the dataset and languages used for the experiment. Section 3 discusses our methodology, which includes the algorithm proposed for the removal of dissimilar translations from the dataset. Results are illustrated in Section 4, and Section 5 gives the conclusion and future work.

2 Dataset and Languages Used

For the experiment, Samanantar dataset is used to compare and test the impact of dataset quality and quantity on MT tasks using Hindi (HIN) and Odia (ODI) languages. This Samanantar dataset contains over 40 million sentence pairs from English (ENG) to ILs. The Flores200 dataset (Costa-jussà et al., 2022) is utilized for testing purposes. A rich morphological and structural variation can be seen in ILs, which makes MT and NLP tasks challenging (Das et al., 2022), (Vikram, 2013).

3 Methodology

The quality of the dataset cannot be guaranteed when obtained from different sources. However, these datasets are crucial for an MT system to function effectively and give a correct translation. For the experiment, we have used both the original unfiltered (dataset taken directly from Samanatar) and after removing dissimilar translation (from the original data, we have removed dissimilar translation). The dataset is standardized and tokenized with the Indic NLP library (Kunchukuttan, 2020) for both Indian Languages. Previous research has shown that filtering techniques can positively and negatively affect training datasets (Steingrímsson et al., 2023), which motivates us to explore these datasets. Therefore, this paper proposes a novel technique that can be used to remove sentences with incorrect translations from the dataset.

3.1 A thresholding method for removing dissimilar Translation: To check the quality of dataset

While examining the dataset, it has been observed that it possesses lots of incorrect translations or dissimilar mistranslations (Bala Das et al., 2023). The term ‘‘dissimilar mistranslations or incorrect translations’’ in the context of a dataset refers to a translation that fails to accurately represent the intended significance of the original text. This can happen for various reasons, including translation errors, absence of contextual knowledge, variations in the grammar and structure of the source and target languages, or downright wrong translations. However, while training any MT models, it acquires knowledge by analyzing examples from these training datasets, and if this dataset has dissimilar inaccurate translations, it can cause improper learning. As a result, the model is found to underperform in MT Tasks (providing wrong translations while converting source to target languages). These dissimilar translations can impede model performance. A few examples of dissimilar translations that can be

observed in the Samanatar dataset are shown in Table 1 and Table 2.

It is important to remove *non-parallel* (incorrect) sentence pairs from the training data so that MT models learn faithful source–target mappings. In this work, we apply a fully automatic, no-human-in-the-loop filter before final training and study its impact by training on several corpus sizes (Full, Quarter, and a fixed 500k subset for comparability). We use the following notation. C_{TR} and C_V are the training and validation sets. Their source/target sides are $C_{TR|S}, C_{TR|T}$ and $C_{V|S}, C_{V|T}$. L_S and L_T denote the source and target language tags. A single training example is $(x, y) \in C_{TR}$, where $x \in L_S$ and $y \in L_T$. The baseline model's best checkpoint is P_B; its translation of x is $\hat{y} = P_B(x)$. We compute two BLEU scores: (i) *corpus* BLEU on the validation set, $B_{\text{dev}} = \text{BLEU}\big(C_{V|T}, P_B(C_{V|S})\big)$, which we use only to set a cutoff; and (ii) *sentence-level* BLEU (with smoothing) for each pair, $b(x, y) = \text{BLEU}\big(y, \hat{y}\big)$, which we use to keep or discard pairs.

We define a single cutoff per direction as

$$\tau \;=\; \frac{B_{\text{dev}}}{4}.$$

A training pair (x, y) is kept if $b(x, y) \geq \tau$ and discarded otherwise. Sentence-level BLEU is typically lower and more variable than corpus BLEU; the $B_{\text{dev}}/4$ calibration removes clear mismatches while preserving useful pairs. Empirically this often retains about 70–80% of pairs; this is descriptive, not a fixed percentile rule.

For EN→ODI, we translate the English side of the validation set into Odia to obtain B_{dev}, set $\tau = B_{\text{dev}}/4$, then score each EN–ODI training pair by sentence-level BLEU and remove pairs with $b(x, y) < \tau$. We apply the same procedure independently to ODI→EN and to the Hindi pair. After filtering, we retrain the final model from scratch on the retained pairs. We report results for **Unfiltered** (original data) and **Filtered** (after removing pairs with $b(x, y) < \tau$) at three sizes:

Case 1: Original (Full) dataset

Case 2: Quarter of the dataset

Case 3: 500k sentence pairs (chosen for comparability across languages)

Following are the steps conducted to remove dissimilar translations from the dataset:

1. Train the model on the original unfiltered dataset (without removing any dissimilar sentences from the original dataset).

2. Use the newly created model to translate the validation dataset and find its BLEU score using the existing validation dataset as a reference.

3. A threshold score (in this case, score/4) is created using this BLEU score *(here,* score *means the* corpus *BLEU on the validation set, i.e., B_{dev}, so the cutoff is $\tau = B_{dev}/4$; keep/discard decisions use* sentence-level *BLEU).*

4. The entire train set is translated using the existing model checkpoints.

5. The translated training set is compared with the original training set, and any sentences that receive a BLEU score lower than the computed threshold are eliminated from the original dataset.

Algorithm 1 Removal of dissimilar translation from the dataset

Input: Train dataset C_{TR}, Validation dataset C_V, Source Language L_S, $C_{V|T}$ denotes the part of C_V written in language T, Target Language L_T, Parameter: y-Number of Epochs

$T_F \leftarrow C_{TR} + C_V$
Train model on C_{TR} from L_S to L_T for y epochs with best checkpoints saved in P_B
$C'_{V|T} \leftarrow P_B(C_{V|S})$
$B \leftarrow \text{BLEU}(C_{V|T}, C'_{V|T})$
Threshold $\leftarrow B/4$ sentence pair (P_S, P_T) in $(C_{TR|S}, C_{TR|T})$
$P'_T \leftarrow P_B(P_S)$
$M \leftarrow \text{BLEU}(P_T, P'_T)$ $M <$ Threshold
Discard (P_S, P_T) from C_{TR} **Return** C_{TR}

Finally, to map symbols in the pseudocode to the prose above: B is the validation *corpus* BLEU B_{dev} (so "$B/4$" equals the cutoff $\tau = B_{\text{dev}}/4$); the loop variables (P_S, P_T) correspond to (x, y); P'_T is the hypothesis $\hat{y}$; and M is the sentence-level BLEU $b(x, y) =$

Table 1: Few Examples of Wrong Translation Observed in Odia Language Dataset

Source	Incorrect Translation from dataset	Actual Translation
think about the hospital administration staff, ambulance drivers, ward boys, sanitation workers who are serving others in these difficult conditions.	ଅନ୍- ସ୍ୱାସ୍ଥ୍ୟ କର୍ମଚାରୀ, ପରିମଳ କର୍ମଚାରୀ, କର୍ମଚାରୀ, କର୍ମଚାରୀ, କର୍ମଚାରୀ, ଆମ୍ବୁଲାନ୍ସ ସେବା କ୍ଷେତ୍ରରେ କାର୍ଯ୍ୟରତ ଅଛନ୍ତି। (swasthya karmachari, parimala karmachari, karmachari, karmachari, karmachari, ambulaans seba khetrare karjyarata achhanti.)	ଡାକ୍ତରଖାନା ପ୍ରଶାସନର କର୍ମଚାରୀ, ଆମ୍ବୁଲାନ୍ସ ଡ୍ରାଇଭର, ୱାର୍ଡ ବାଳକ, ପରିମଳ କର୍ମଚାରୀ ଯେଉଁମାନେ ଏହି କଠିନ ପରିସ୍ଥିତିରେ ଅନ୍ୟମାନଙ୍କୁ ସେବା କରୁଛନ୍ତି ସେମାନଙ୍କ ବିଷୟରେ ଚିନ୍ତା କରନ୍ତୁ	(daktarakhanaa prasasanara karmachari, ambulaans driver, ward balaka, parimala karmachari jeunmane ehi kathina paristhitire anyamananku seba karuchhanti semananka bisayare chinta karantu |)
ଏମାନେ ବହୁତ କଥା କହୁଛନ୍ତି।(emane bahuta katha kahuchhanti.)	these matter a lot.	They are talking a lot.

Table 2: Few Examples of Wrong Translation Observed in the Hindi Dataset

Source	Incorrect Translation from dataset	Actual Translation
on wednesday morning, sonu singh, devendra singh and abhishek singh came to the village.	बीते बुधवार को गांव के रहने वाले दबंग सोनू, अभिषेक और देवेन्द्र ने दलित परिवार पर ताबड़तोड़ फायरिंग कर दी थी।(beete budhavaar ko gaanv ke rahane vaale dabang sonoo, abhishek aur devendr ne dalit parivaar par taabadatod phaayaring kar dee thee.)	बुधवार की सुबह सोनू सिंह, देवेन्द्र सिंह और अभिषेक सिंह गांव आये।(budhavaar kee subah sonoo sinh, devendr sinh aur abhishek sinh gaanv aaye.)
one way is by being a good listener.	हो । "(ho)	एक तरीका अच्छा श्रोता बनना है।(ek tareeka achchha shrota banana hai.)
jagat mashay remained ever grateful.	मेरी बदनामी होगी।(meree badanaamee hogee.)	जगत महाशय सदैव आभारी रहे।(jagat mahaashay sadaiv aabhaaree rahe)

BLEU$(y, \hat{y})$. The assignment $T_F \leftarrow C_{TR} + C_V$ is a notational union; the model is trained on C_{TR} only.

Algorithm 1 describes the steps followed for our experiment. Tables 3 and 4 show the statistics of the dataset after filtering, as well as the removal of dissimilar sentences that were found in the dataset and the dissimilar translations that were noted in the dataset.

3.2 Model Training

The MT model utilized is based on the transformer (Vaswani et al., 2017) architecture, with six encoder-decoder pairs and 512 hidden layers and token embeddings. The models have been constructed by the Fairseq (Ott et al., 2019) library. The dataset is segmented using the Byte Pair Encoding (BPE) technique (Sennrich et al., 2015), which involves splitting the words into more manageable subwords. Experiments are conducted using unfiltered datasets and, after removing dissimilar translations, obtained from standardization and tokenization using the Indic NLP library (Kunchukuttan, 2020).

4 Results and Discussion

MT evaluation is the most important phase of any MT system. Three different evaluation metrics are utilized to verify the system's effectiveness. These metrics are well-known and effective in determining the quality of translated texts. METEOR (Banerjee and Lavie, 2005), RIBES (Tan et al., 2015), and BLEU (Papineni et al., 2002) are the evaluation metrics used in this work. The evaluation uses the Flores-200 dataset (Costa-jussà et al., 2022) for testing.

Table 3: Statistics of Dataset

Dataset Division	Category	Pairs	Training Dataset
Full dataset	**Unfiltered**	ODI	990439
		HIN	8431687
	After Removal of Dissimilar translation	ODI	938582
		HIN	5246667
Quarter dataset	**Unfiltered**	ODI	247609
		HIN	2115315
	After Removal of Dissimilar translation	ODI	232255
		HIN	1760096
500K Sentences dataset	**Unfiltered**	ODI	500000
		HIN	500000
	After Removal of Dissimilar translation	ODI	473747
		HIN	439207

Table 4: Observed incorrect translations in the Dataset

Language	Dataset Division	Wrong Translation Observed	Threshold
ODI	**Full**	50860	1.729
	Quarter	14357	0.299
	500K	25256	1.009
HIN	**Full**	3184023	8.166
	Quarter	354222	7.505
	500K	59281	6.132

Our model is run on a high-performance workstation equipped with an Intel Xeon W-1290 CPU, with 10 physical cores and 20 threads (3.20 GHz base frequency, up to 5.20 GHz boost), providing robust multi-threading and caching with 20 MiB of L3 cache. The system includes 62 GB of RAM and an NVIDIA Quadro RTX5000 GPU with 16GB of VRAM, supported by driver version 535.154.05. The system uses CUDA 11.5 for compilation and is compatible with CUDA 12.2 for runtime operations, optimizing model training performance. Table 4 displays the scores of the models trained with different dataset sizes without filtration and after the removal of dissimilar translations. RIBES and METEOR scores range from 0 to 1, whereas the BLEU score ranges from 0 to 100.

The results show that after removing dissimilar translations, the evaluation metrics outperform for Odia and Hindi languages. BLEU score is calculated only using the precision of the translation that is directly calculated over all n-grams (n ranges from 0 to 4), and recall is indirectly considered using the Brevity Penalty (BP) that penalizes shorter translations. METEOR score, on the other hand, makes use of precision and recall directly during calcula-

Table 5: Evaluation Metrics with Unfiltered Dataset and After removal of Dissimilar translation from the dataset

Dataset Division	Category	Pairs	BLEU	METEOR	RIBES
Full dataset	**Unfiltered**	ENG-ODI	6.38	0.17	0.66
		ODI-ENG	14.53	0.27	0.72
		ENG-HIN	32.37	0.47	0.81
		HIN-ENG	34.61	0.42	0.83
	Dissimilar translation	ENG-ODI	7.06	0.21	0.68
	Removal	ODI-ENG	16.22	0.33	0.73
		ENG-HIN	32.45	0.50	0.81
		HIN-ENG	35.32	0.53	0.84
500k Sentences dataset	**Unfiltered**	ENG-ODI	3.48	0.11	0.61
		ODI-ENG	9.15	0.20	0.65
		ENG-HIN	24.58	0.39	0.77
		HIN-ENG	26.75	0.40	0.79
	Dissimilar translation	ENG-ODI	4.09	0.16	0.63
	Removal	ODI-ENG	10.25	0.26	0.67
		ENG-HIN	25.44	0.43	0.77
		HIN-ENG	26.81	0.47	0.81
Quarter dataset	**Unfiltered**	ENG-ODI	1.06	0.06	0.46
		ODI-ENG	3.95	0.13	0.56
		ENG-HIN	30.03	0.44	0.79
		HIN-ENG	32.94	0.45	0.83
	Dissimilar translation	ENG-ODI	1.19	0.09	0.46
	Removal	ODI-ENG	4.67	0.17	0.58
		ENG-HIN	30.76	0.48	0.80
		HIN-ENG	32.96	0.51	0.83

tion and penalizes a translation based on the number of chunks (fluent "chunks" of translations) it has, i.e., the lower the number of chunks, the lesser the penalty. From the result, it is clear that in most cases, dissimilar translation removal performs better in evaluation metrics. The BLEU scores achieved are higher than those of the unfiltered dataset. The removal of grave dissimilar translations improves dataset quality tremendously. The trade-off between dataset quality and size is well paid off in the case of Odia, wherein the loss in data is at most 6% while the increase in BLEU score is considerable. Results also reveal that ILs to the English language performs better than English to the ILs.

4.1 ENG - ODI and ODI - ENG

In the case of the ODI dataset, models trained after removing dissimilar translation sentences from the dataset outperform the unfiltered dataset. The scores are found to be almost linearly increasing as the dataset size increases, and this implies room for improvement to be proportional to the growth of the ODI dataset. Using the ODI dataset, the lowest scores (in terms of evaluation metrics) occur in the quarter-sized dataset. The lowest BLEU score is found to be 1.06 for the ODI-ENG language for the quarter dataset. METEOR and RIBES scores for the quarter dataset are also less for the ENG-ODI language. For the ODI dataset, it becomes clear that using the full dataset division and removing Dissimilar translation consistently produces positive results (i.e., 16.22 BLEU score) across a range of dataset division categories. From all the cases, the highest results in terms of evaluation metrics are achieved with the original, unfiltered dataset with a BLEU score of 7.06 and 16.22 for ENG-ODI and ODI-ENG, respectively. Some outputs generated from the model are shown in Table 6.

Table 6: Sample Translation for ENG-ODI and ODI-ENG for different cases of dataset

Language Pairs	Source	Output generated	Reference
ENG-ODI (Full dataset)	ପରିଶେଷରେ ଅଗ୍ନିଶମ ଉଦ୍ଧାରକାରୀ ଦଳ ରାତି 11:35ଟା ସୁଦ୍ଧା ନିଆଁ ଲିଭାଇଥିଲେ।	Finally, the fire was doused by the rescue team at 11: 35pm.	The fire rescue team finally doused the fire by 11:35 pm.
ODI-ENG	ପରିଶେଷରେ ଅଗ୍ନିଶମ ଉଦ୍ଧାରକାରୀ ଦଳ ରାତି 11:35ଟା ସୁଦ୍ଧା ନିଆଁ ଲିଭାଇଥିଲେ।	Fire rescue crews eventually doused the fire by 11:35 pm.	The fire rescue team finally doused the fire by 11:35 p.m.

Table 7: Sample Translation for ENG-HIN and HIN-ENG generated from the Model

Language Pairs	Source	Output genertaed	Reference
ENG-HIN	he even suggests that such abilities in interpreting human behavior may be shared by animals such as domestic dogs.	वह यह भी सुझाव देता है कि मानव व यवहार की व याख या करने में ऐसी क षमताओं को घरेलू कुतर्तों जैसे जानवरों द वारा साझा किया जा सकता है।	वे यह भी सुझाव देते हैं कि मानव व्यवहार में व्याख्या करने की जो क्षमताएँ हैं उसे पालतू कुत्ते जैसे जानवर के ज़रिए भी शेयर किया जा सकता है.
HIN-ENG	वे यह भी सुझाव देते हैं कि मानव व्यवहार में व्याख्या करने की जो क्षमताएँ हैं उसे पालतू कुत्ते जैसे जानवर के ज़रिए भी शेयर किया जा सकता है.	due to only eighteen models available in a day, many countries failed to climb the model podium.	with only eighteen medals available a day, a number of countries have failed to make the medal podium.

4.2 ENG-HIN and HIN-ENG

It has been observed that from all three cases, the 500k sentences dataset with the removal of the dissimilar translation for HIN has the lowest BLEU score (24.54). In contrast, after removing the dissimilar translation from the full dataset, the BLEU score is 35.32. RIBES and METEOR scores after dissimilar translation removal from the full dataset for HIN-ENG are 0.53 and 0.84. Sample translations generated by the model are shown in Table 7.

5 Conclusion and Future Work

This paper examines the impact of the dataset in terms of size and quality for the MT task. A few dissimilar translations are noticed in the dataset of two languages, i.e., Hindi (HIN) and Odia, shown in the paper. First, a baseline NMT model is built for both languages and then the method to remove the dissimilar translation from the dataset is presented in this paper. For all our experiments, various assessment metrics, such as BLEU, RIBES, and METEOR, are used to check the overall quality of translation. The results have shown that removing dissimilar translations improves the quality of translation. It is also noticed that, even though the ILs-English and English-ILs systems are trained using the same corpus, ILs-English works more efficiently across all the evaluation metrics since ILs often differ in sentence form, word order, and morphology from English. Additionally, based on the analysis of th experiments, it is concluded that the size of the dataset is directly proportional to the quality of the translation. More language pairs with different dataset sizes for MT tasks need to be tested for future work. The impact of the removal of Dissimilar translation with different variations of the threshold will be studied and investigated on other datasets to establish the underlying reasons behind the observed results.

Limitations

This research mainly assesses and checks the effects of dataset size and mistranslation removal on the translation dataset, primarily focusing on two Indian languages. Despite its advantages, this method might not work for other languages, especially those with distinct linguistic structures or low-resource traits. Currently, our approach mainly uses particular

criteria to find and exclude dissimilar translations, which could need to be improved for wider use for other dataset. Future research might look into expanding this strategy to additional languages and assessing how well it works in various contexts and areas. To improve scalability, more studies should look into automated error-detection techniques.

6 Ethics Statement

This research mainly focuses on improving the quality of machine translation models for Indian Languages by checking and filtering translation data. The paper also proposes a novel technique to remove incorrect or disimmilar translations from the dataset and checks its effect on MT tasks. It also investigates how different dataset divisions (into multiple subsets) influence the translation quality. We have taken care to ensure that our training and evaluation processes do not contain any affecting or biased content, and all of the data used in our studies came from open sources.

References

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.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, and Juan Wang. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

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 Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–24.

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.

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. In *Proceedings of the 9th Workshop on Asian Translation*, pages 73–77, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

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

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.

Dheeraj Kodati and Ramakrishnudu Tene. 2025a. Advancing mental health detection in texts via multi-task learning with soft-parameter sharing transformers. *Neural Computing and Applications*, 37(5):3077–3110.

Dheeraj Kodati and Ramakrishnudu Tene. 2025b. 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(13):11729–11752.

Anoop Kunchukuttan. 2020. The indicnlp library. https://github.com/anoopkunchukuttan/indic_nlp_library.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 311–318.

Gokul Ramesh, Sumanth Doddapaneni, Anoop Bheemaraj, Mayank Jobanputra, AK R, Aishwarya Sharma, and Mitesh S. Khapra. 2022. Samanantar: The largest publicly available parallel dataset collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, pages 145–162.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Steinþor Steingrímsson, Höður Loftsson, and Andy Way. 2023. Filtering matters: Experiments in filtering training sets for machine translation. In *Proceedings of the 24th Nordic Conference on Computational Linguistics*, pages 588–600.

Liling Tan, Jon Dehdari, and Josef van Genabith. 2015. An awkward disparity between BLEU/RIBES scores and human judgements in machine translation. In *Proceedings of the 2nd Workshop on Asian Translation*, pages 74–81.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 1–11.

Shweta Vikram. 2013. Morphology: Indian languages and european languages. *International Journal of Scientific and Research Publications*, pages 1–5.

A Multi-Task Learning Approach to Dialectal Arabic Identification and Translation to Modern Standard Arabic

Abdullah Khered[1,2], **Youcef Benkhedda**[1] **and Riza Batista-Navarro**[1]
[1]The University of Manchester, UK
[2]King Abdulaziz University, Saudi Arabia
abdullah.khered@manchester.ac.uk
youcef.benkhedda@manchester.ac.uk
riza.batista@manchester.ac.uk

Abstract

Translating Dialectal Arabic (DA) into Modern Standard Arabic (MSA) is a complex task due to the linguistic diversity and informal nature of dialects, particularly in social media texts. To improve translation quality, we propose a Multi-Task Learning (MTL) framework that combines DA-MSA translation as the primary task and dialect identification as an auxiliary task. Additionally, we introduce LahjaTube, a new corpus containing DA transcripts and corresponding MSA and English translations, covering four major Arabic dialects: Egyptian (EGY), Gulf (GLF), Levantine (LEV), and Maghrebi (MGR), collected from YouTube. We evaluate AraT5 and AraBART on the Dial2MSA-Verified dataset under Single-Task Learning (STL) and MTL setups. Our results show that adopting the MTL framework and incorporating LahjaTube into the training data improve the translation performance, leading to a BLEU score improvement of 2.65 points over baseline models.

1 Introduction

Machine Translation (MT) is a Natural Language Processing (NLP) task that aims to translate between natural languages automatically. Over the last decade, Neural Machine Translation (NMT) has improved translation quality by leveraging deep learning to model complex linguistic patterns from large datasets. A widely used NMT architecture is the Sequence-to-Sequence (Seq2Seq) model, which consists of an encoder-decoder framework typically based on Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU)(Cho et al., 2014). The encoder processes the input sentence into a compressed representation, which the decoder then uses to generate the translated output(Sutskever et al., 2014). More recently, Transformer-based models have surpassed earlier Seq2Seq architectures by replacing recurrence with self-attention and parallel computation (Vaswani et al., 2017), resulting in faster translation, improved accuracy, and better handling of long-range dependencies. Furthermore, pre-trained Transformer-based models have demonstrated state-of-the-art performance across various NLP tasks beyond machine translation, solidifying the Transformer as the dominant architecture in modern NLP research (Qiu et al., 2020).

Despite these advancements, low-resource language translation remains a challenge, particularly for Dialectal Arabic (DA) to Modern Standard Arabic (MSA) translation. Arabic operates in a diglossic environment: MSA is the standardised form used in education, media, and formal communication, while DA is the informal variant shaped by regional cultures, local expressions, and daily communication (Salloum et al., 2014; Sadat et al., 2014). The challenge in DA-MSA translation lies in the variability across Arabic dialects. Each dialect has morphological and syntactic differences, often incorporating borrowed words from other languages and region-specific expressions (Mallek et al., 2017). Moreover, the rise of social media has further complicated these challenges, as Arabic speakers frequently mix dialects, use slang, abbreviations, emojis, and code-switching with other languages (Alruily, 2020).

To address these challenges, fine-tuning Arabic pre-trained Transformer models such as AraBART (Kamal Eddine et al., 2022) and AraT5 (Elmadany et al., 2023) on dialect-specific corpora has proven beneficial in overcoming data scarcity for DA-MSA translation (Khered et al., 2025). Moreover, Multi-Task Learning (MTL) has emerged as a promising approach for enhancing DA-MSA translation. Instead of training a model solely for translation, MTL enables joint training on multiple related

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 21–31, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-004

tasks, such as MSA-English translation (Baniata et al., 2018b), Part-Of-Speech (POS) tagging (Baniata et al., 2018a) and translation of multiple dialects (Moukafih et al., 2021). These auxiliary tasks provide additional linguistic signals that help improve model generalisation, contextual understanding, and robustness to informal variations. Our research builds upon normalising Arabic text in social media and improving the results on the Dial2MSA-Verified dataset (Khered et al., 2025) by integrating an MTL framework. We also introduce the LahjaTube dataset, a new corpus sourced from YouTube videos, to enrich model training. LahjaTube was developed to address the shortage of DA-MSA translation datasets, particularly those that include informal and real-world language from major Arabic dialects as commonly found on social media. Our objectives include:

- Develop an MTL framework for DA-MSA translation that incorporates dialect identification as an auxiliary task.
- Automatically collect and construct the LahjaTube dataset, covering four major Arabic dialects: Egyptian (EGY), Gulf (GLF), Levantine (LEV), and Maghrebi (MGR) with their corresponding MSA and English translations.
- Evaluate the performance of MTL models on both DA-MSA translation and dialect identification tasks using the Dial2MSA-Verified dataset. This includes training on different combinations of datasets, incorporating the newly introduced LahjaTube corpus.

The code for the MTL framework and supplementary material for this paper are available online at `https://github.com/khered20/MTL-Dial2MSA`.

2 Related Work

MTL is a machine learning technique that jointly trains multiple tasks, allowing knowledge sharing between related tasks (Zhang and Yang, 2017). MTL has been explored in various NLP tasks (Kumar et al., 2019; Chen et al., 2024) including those with limited data resources (Mamta et al., 2022; Guzman et al., 2024; Elgamal et al., 2024), leading to more generalised representations.

In Arabic, MTL has been applied to various linguistic tasks, including diacritic restoration, where auxiliary tasks such as word segmentation and POS tagging have been utilised to enhance accuracy (Alqahtani et al., 2020). Similarly, dialect identification has benefited from MTL approaches, with hierarchical attention mechanisms improving fine-grained classification at the city, state, and country levels (Abdul-Mageed et al., 2019). Moreover, MTL has been integrated with pre-trained language models such as MARBERT (Abdul-Mageed et al., 2021) for Arabic dialect identification at both the country and province levels, demonstrating that sharing task-specific attention layers improves generalisation across Arabic varieties (El Mekki et al., 2021). Additionally, Arabic Natural Language Understanding (ANLU) has been enhanced through MTL frameworks that facilitate parameter sharing across multiple tasks. This approach has led to a notable performance on some tasks within the ALUE benchmark, highlighting the importance of carefully considering task relationships and loss scaling (Alkhathlan and Alomar, 2024).

Recent studies in MTL with MT have revealed that incorporating auxiliary tasks can improve translation performance (Zaremoodi et al., 2018; Pham et al., 2023). In the context of DA-MSA translation, various MTL approaches have been proposed. Baniata et al. (2018b) explored a unified multitask NMT model where DA-MSA translation served as the main task and MSA-English translation as the auxiliary task. The architecture utilised a separate encoder for each task whilst sharing a single decoder. Another study by Baniata et al. (2018a) further improved MTL for Arabic dialect translation by integrating POS tagging as an auxiliary task. This model adopted a shared-private Bi-LSTM-CRF architecture, encoding DA sentences and segment-level POS tags. The results demonstrated that the POS tagging task improved the translation BLEU score. Similarly, Moukafih et al. (2021) adopted a seq2seq MTL framework, encoding and decoding pairs of different dialects and MSA within the PADIC-parallel dataset (Meftouh et al., 2018) using a shared GRU model. Their Many-to-One sitting improved the translation performance, surpassing statistical MT models in 88% of translation cases.

Our research focuses on Arabic social media normalisation, specifically translating DA into MSA within the Dial2MSA-Verified dataset (Khered et al., 2025). The Dial2MSA-Verified, an extension of Dial2MSA (Mubarak, 2018), is a multi-reference dataset covering tweets from four di-

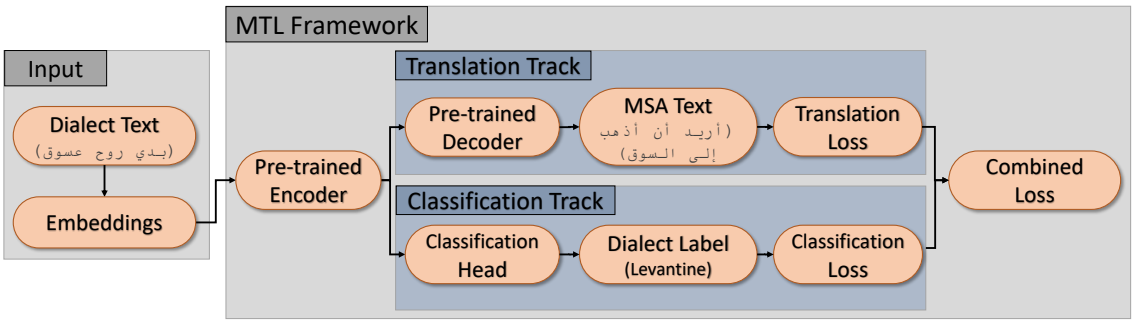


Figure 1: Architecture of the MTL framework processes a shared pre-trained encoder, followed by two parallel tasks: a translation track that generates MSA text using a pre-trained decoder and a classification track that predicts the dialect label

alects: EGY, MGR, GLF and LEV dialects with their multiple MSA translations. Khered et al. (2025) further explores joint and independent training strategies, demonstrating that joint training across dialects leads to superior translation performance. Additionally, transformer-based models, including AraT5 (Elmadany et al., 2023) and AraBART (Kamal Eddine et al., 2022), have been benchmarked, with AraT5 emerging as the best-performing model. In this context, we propose a novel MTL framework that leverages Arabic-specific pre-trained Transformer models (AraT5, AraBART) for DA-MSA translation and dialect identification. Furthermore, we introduce LahjaTube, a dataset containing four Arabic dialects, each with multiple transcripts, along with their corresponding MSA and English translations to enrich the training data.

3 MTL for DA-MSA Translation and Dialect Identification

In this section, we present our Multi-Task Learning (MTL) framework designed to simultaneously perform two tasks: DA-MSA translation and dialect identification, as illustrated in Figure 1. This framework is built upon state-of-the-art Transformer models and optimises both tasks through shared representations. The dataset used in this study consists of Arabic dialectal sentences paired with their corresponding MSA translations and dialect labels. The labels encompass the four dialects: EGY, GLF, LEV, MGR as well as MSA.

3.1 Architecture Overview

The architecture is based on a Transformer encoder-decoder model that is specifically pre-trained in Arabic Language, such as AraT5 and AraBART, enhanced with an additional classification head. The architecture consists of the following components:

- **Encoder** converts the input DA text into a high-dimensional vector representation, which is shared between the translation and classification tasks.
- **Decoder** generates the corresponding MSA translation from the encoder's representation.
- **Classification Head** is added to the encoder output to perform dialect classification.

3.2 Loss Functions

Our model is trained using an MTL approach that combines two objectives: dialect classification and Seq2Seq translation. To achieve this, we define two separate loss functions and combine them into a weighted objective function.

Classification Loss For dialect classification, we use a separate classification head, which applies a linear transformation followed by a softmax activation. The model predicts the probability distribution over C dialect classes. The classification loss is formulated using Cross-Entropy Loss:

$$\mathcal{L}_{\text{classification}} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \quad (1)$$

C is the total number of classes (four dialects and MSA). y_c is the true label, represented as a one-hot vector. $\hat{y}_c$ is the predicted probability for class c.

Translation Loss For the translation task, we also use the Cross-Entropy Loss, which measures the difference between the predicted probability distribution and the actual target sequence. The translation loss is defined as:

$$\mathcal{L}_{\text{translation}} = -\sum_{t=1}^{T}\sum_{v=1}^{V} y_{t,v}\log(\hat{y}_{t,v}) \quad (2)$$

T is the target sequence length and V is the vocabulary size. $y_{t,v}$ is a one-hot vector representing the true token at position t. $\hat{y}_{t,v}$ is the predicted probability for token v at position t.

Combined Loss Function To train the model jointly for both tasks, we define a weighted combination of the translation and classification losses:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{translation}} + (1-\alpha)\mathcal{L}_{\text{classification}} \quad (3)$$

where α is a hyperparameter in the range $(0, 1)$ that controls the relative importance of the two tasks. This combined loss ensures the model learns the classification and translation tasks simultaneously.

4 LahjaTube Dataset

In this section, we introduce the LahjaTube dataset, a collection of transcripts from YouTube videos covering DA from four Arabic-speaking regions: EGY, GLF, LEV, and MGR. These transcripts are accompanied by English translations which were translated into MSA. The LahjaTube dataset is available upon request for academic purposes.

4.1 Data Collection

The data collection concentrated on YouTube videos created by content creators who speak one of the four aforementioned dialects. We employed the YouTube Data API v3[1] to select videos from countries representative of these dialects. The filtering functions were used to include only videos under Creative Commons Attribution licenses and contain subtitles from both Arabic and English. Once an initial set of videos was identified, we explored the creators' other videos that met our specific criteria. We collected a total of 1,912 videos distributed across the four selected dialects. For the caption extraction process, we used the YouTube Video Subtitles Scraper from the Apify platform[2] to retrieve both the original Arabic transcripts and their corresponding English translations.

4.2 Data Processing and Cleaning

To ensure the quality of the extracted data, we undertook several cleaning and preprocessing steps. First, each sample was defined according to the timestamped segments provided by YouTube subtitles, so that each instance in our dataset corresponds to an English subtitle segment as determined by the video's original caption timing. If subtitles occurred with minimal time gaps and without sentence-final punctuation, we merged them into a single sample. In cases where a single subtitle segment contained multiple short sentences separated by in-line punctuation, we kept these grouped as a single data instance. We also removed any subtitle containing fewer than four words (in either the dialectal Arabic or English lines) to reduce potential noise and ensure sufficient linguistic content. Furthermore, the geographic location of a video's creator alone does not guarantee the actual dialect of the transcripts, as the creator could use different dialects or MSA, host guests from other regions, or produce videos while travelling. We addressed this by applying a dialect identification model to verify the dialect used in each line, ensuring our dataset includes only transcripts where the identified dialect corresponds with the creator's known dialect. The model used is MTL-AraBART, the high-performing dialect identification model produced in this study, trained on the same datasets used in Khered et al. (2025).

4.3 MSA Translation

To enable translation from DA to MSA, we generated MSA translations by translating the English subtitles into MSA using the few-shot GPT-4o[3] model via its API. For each dialect, we designed a specific prompt that included three few-shot examples, which were manually selected from our collected DA–English subtitle pairs. Native Arabic speakers provided accurate MSA translations for these selected dialectal samples, and these few examples were incorporated into the GPT-4o prompt. As illustrated in Figure 2, [Dialect] specifies the relevant dialect, while [DA] and [EN] refer to the original DA transcript and its English translation, respectively.

4.4 Corpus Statistics

The final corpus comprises a total of 31,938 transcripts from YouTube videos distributed across the four aforementioned dialects, along with their English and MSA translations. Subsequently, these transcripts are distributed as shown in Table 1, capturing a variety of dialect-specific expressions and

[1] https://developers.google.com/youtube/v3

[2] https://apify.com/

[3] https://platform.openai.com/docs/models/#gpt-4o

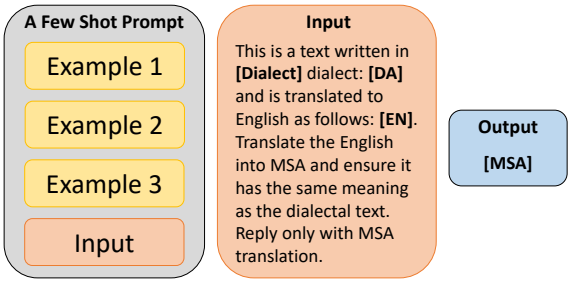


Figure 2: Few-shot prompting strategy used to convert English translations into MSA using the GPT-4o model

vocabulary. Table 1 provides detailed statistics including the size of each dialect corpus, along with the total and unique word counts for both the DA and the corresponding MSA translations.

Dialect	Size	Total Words	Unique Words	Total MSA Words	Unique MSA Words
EGY	10,279	110,387	21,417	112,470	21,613
GLF	7,762	106,669	13,269	112,277	16,192
LEV	7,695	98,138	15,996	102,010	16,851
MGR	6,202	95,148	16,329	95,829	17,439

Table 1: Statistics for LahjaTube corpus

Table 2 highlights several examples from the LahjaTube dataset. In some cases, such as the LEV example, the text might start or end suddenly because it could be part of a larger conversation. Despite this, the English and MSA translations accurately capture the original meaning of the DA conversation, ensuring that all versions convey the same text.

EGY	وبالتالي ممكن انك تشيل الطاقة وتعوض مكانها بالكلام ده
MSA	وبالتالي يمكنك أن تزيل الطاقة وتستبدلها بهذا الكلام.
EN	Therefore, you can remove the energy and replace it with this talk.
GLF	والله لاصدمه خله يكلمني والله لاصدمه طيب قل له خمسين ريال
MSA	والله لأفاجئه، دعه يكلمني، والله لأفاجئه. حسناً، قل له خمسين ريال
EN	I swear I will shock him. Let him talk to me. I swear I will shock him. Okay, tell him fifty riyals.
LEV	معك من القران بس ما بيعرف شو الحكم الفقهي بهال
MSA	معك من القرآن لكنه لا يعرف ما هو الحكم الفقهي في هذا
EN	You know from the Qur'an, but he does not know what the jurisprudential ruling is on this
MGR	المنافع ديالو ذاكشي علاش جربتو والله الحمد لقيت عليه نتيجه دابا
MSA	فوائده هي السبب الذي جعلني أجربه، والله الحمد وجدت نتائج الآن
EN	Its benefits are why I tried it, and thank God I found results now.

Table 2: DA transcripts with their MSA and English translations from LahjaTube dataset

4.5 Human Evaluation of MSA Translations

We conducted a human evaluation on a subset of 200 DA-MSA translation pairs from LahjaTube, with 50 samples per dialect. For each dialect, one annotator, a native speaker of the relevant dialect, evaluated only samples from their own dialect. The evaluation followed the multi-dimensional method proposed by Sadiq (2025), which assessed accuracy, fluency, style and tone, cultural suitability, and terminology on a 1-5 scale. As shown in Table 3, the MSA translations in LahjaTube showed high overall quality, with average ratings above 4.5 across most criteria and dialects.

Dialect	Acc	Flu	S&T	Cult	Term	Average
EGY	4.42	4.3	3.7	4.32	4.22	4.19
GLF	4.74	4.72	4.12	4.82	4.74	4.63
LEV	4.62	4.6	4.12	4.74	4.82	4.58
MGR	4.62	4.6	4.16	4.72	4.74	4.57

Table 3: Average human evaluation scores (Acc = Accuracy, Flu = Fluency, S&T = Style & Tone, Cult = Cultural Suitability, Term = Terminology) for DA-MSA translation on a LahjaTube subset (N=50 per dialect, 200 samples)

5 Experimental Design

We conduct experiments using the MTL structure, where DA-MSA translation forms the primary task, and dialect identification serves as an auxiliary task. This structure allows the model to leverage information about the dialect during the translation process, potentially improving translation accuracy.

5.1 Dataset

The Dial2MSA-Verified dataset (Khered et al., 2025) is a multi-reference evaluation dataset, fully verified and sourced from social media, specifically built for DA-MSA translation. Additionally, we integrate the newly introduced LahjaTube dataset, which was created from YouTube video transcripts based on the same four dialects. For all experiments reported in this work, we evaluated our models on the same fixed development and test sets from Dial2MSA-Verified. As summarised in Table 4, the test set contains 2,000 samples per dialect, with some dialect sentences paired with two or three MSA translation references. To assess the impact of the diversity in training data on model performance, we experimented with three different training subsets as presented in Table 4:

- **Subset 1**: The same training set used in Khered et al. (2025), which serves as a baseline. It includes Dial2MSA-Verified-train along with the following additional resources: PADIC (Meftouh et al., 2018), MADAR-train (Bouamor et al., 2018), Arabic STS (Al Sulaiman et al., 2022), and Emi-NADI (Khered et al., 2023) datasets. This is to compare the performance of our new MTL models against previous models.
- **Subset 2**: This training set combines the Dial2MSA-Verified-train and LahjaTube datasets. The goal of this subset is to evaluate the effectiveness of our newly introduced LahjaTube corpus for DA-MSA translation.
- **Subset 3**: The comprehensive training set, incorporating LahjaTube with all training data from Subset 1. This setup aims to produce the most effective translation models by leveraging the largest dataset available.

	Dataset	EGY	GLF	LEV	MGR
Subset 1	Dial2MSA-V-train	9,099	6,575	4,101	3,312
	PADIC	0	0	12,824	25,648
	MADAR-train	13,800	15,400	18,600	29,200
	Arabic STS	2,758	2,758	0	0
	Emi-NADI	0	2,712	0	0
	Total-train-1	**25,657**	**27,445**	**35,525**	**58,160**
Subset 2	Dial2MSA-V-train	9,099	6,575	4,101	3,312
	LahjaTube	10,279	7,762	7,695	6,202
	Total-train-2	**19,378**	**14,337**	**11,796**	**9,514**
Subset 3	Training Subset 1	25,657	27,445	35,525	58,160
	LahjaTube	10,279	7,762	7,695	6,202
	Total-train-3	**35,936**	**35,207**	**43,220**	**64,362**
	Dial2MSA-V-dev	200	200	200	200
	Dial2MSA-V-test	2000 3-R	2000 3-R	2000 2-R	2000 2-R

Table 4: Dataset setup showing the sizes of the three training subsets. In the test set, R indicates the number of reference MSA translations per DA sentence

5.2 Model Configurations and Training Setup

In this study, we used the second version of **AraT5**[4] model (Elmadany et al., 2023), which is based on the T5 architecture (Raffel et al., 2020). AraT5 has a 12-layer encoder and decoder with 768 hidden units per layer. We also used **AraBART** (Kamal Eddine et al., 2022) model, based on the BART architecture (Lewis et al., 2020), which features a 6-layer encoder and a 6-layer decoder, each with 768 hidden units. Both models are pre-trained on large-scale Arabic corpora and further modified by adding a classification head to the encoder's output for our multi-task setup. The additional classification head enables dialect classification, and its loss is calculated separately, as detailed in Section 3. We generated additional pairs from the MSA targets by replicating them as both source and target sentences. These newly created pairs were assigned to the MSA class. Each model is trained under two different settings:

- **Single-Task Learning (STL)**: The model is trained exclusively for DA-MSA translation, serving as the baseline.
- **Multi-Task Learning (MTL)**: The model is trained jointly for DA-MSA translation and dialect identification.

5.3 Evaluation Metrics

We evaluate model performance using both translation and dialect classification metrics. For translation, we use the Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) and chrF++ (Popović, 2017) scores, both implemented in SacreBLEU (Post, 2018). For dialect classification, we report accuracy, which measures the percentage of correctly predicted dialects, Macro-F1, which computes the F1-score for each class and averages them equally, and Weighted-F1, which adjusts for class imbalance by weighting each class's F1-score based on the number of true instances.

5.4 Hyperparameter Optimisation

Each experiment is conducted under the same hyperparameter settings to ensure fair comparisons. The configurations include a batch size of 16, a learning rate of 5e-5, a maximum sequence length of 128, and training for up to 20 epochs, with early stopping applied if the best BLEU score on the validation set does not improve for three consecutive epochs. BLEU was chosen as the primary metric for early stopping since this study focuses on translation quality. All experiments are run on two Nvidia V100 GPUs. For the MTL setup, the combined loss function, introduced in Section 3, is optimised using weighting values of 0.3, 0.5, and 0.8 to examine the effect of different weighting schemes.

6 Results and Discussion

In this section, we analyse the results of our experiments for DA-MSA translation, comparing the performance of the baseline STL models (STL-AraT5

[4] https://huggingface.co/UBC-NLP/AraT5v2-base-1024

and STL-AraBART) from Khered et al. (2025) with our proposed MTL models. For the dialect identification task, we use as a baseline the results reported by Khered et al. (2025) for an ensemble of multiple fine-tuned MARBERT models (Abdul-Mageed et al., 2021). The MARBERT ensemble was trained and optimised using the hyperparameter described by Khered et al. (2022). We compare these results against our proposed MTL models. While we experimented with different values of α, all results reported in this section are based on the combined loss with $\alpha = 0.5$, which achieved stable performance on both tasks on the development set.

6.1 DA-MSA Translation

Table 5 measures the translation performance using BLEU and chrF++ scores across the three proposed training subsets. For Training Subset 1, the MTL models generally outperform STL models from Khered et al. (2025) in terms of both BLEU and chrF++ scores, particularly for the GLF, LEV and MGR dialects. The overall average BLEU score for MTL-AraT5 reaches 42.23, compared to 41.12 for STL-AraT5, while chrF++ increases from 62.05 to 62.84, highlighting the benefits of incorporating dialect classification as an auxiliary task.

	Model		EGY	GLF	LEV	MGR	Avg
Training Subset 1	STL-AraT5	BLEU	**30.94**	53.96	45.37	34.24	41.12
		chrF++	**52.94**	70.86	65.40	58.99	62.05
	STL-AraBART	BLEU	29.87	51.38	43.07	32.95	39.32
		chrF++	52.26	69.49	64.13	58.12	61.00
	MTL-AraT5	BLEU	29.71	**55.31**	**48.39**	**35.53**	**42.23**
		chrF++	52.21	**72.01**	**67.19**	**59.95**	**62.84**
	MTL-AraBART	BLEU	29.52	53.79	45.98	33.12	40.60
		chrF++	52.31	70.96	66.08	58.48	61.96
Training Subset 2	STL-AraT5	BLEU	29.96	54.21	46.99	34.50	41.42
		chrF++	52.34	71.43	66.52	59.44	62.71
	STL-AraBART	BLEU	26.25	50.44	43.18	32.93	38.20
		chrF++	49.11	69.15	63.94	58.11	60.08
	MTL-AraT5	BLEU	**30.70**	**55.94**	**48.42**	**35.77**	**42.71**
		chrF++	**52.90**	**72.35**	**67.47**	**60.36**	**63.27**
	MTL-AraBART	BLEU	28.41	51.15	43.56	32.88	39.00
		chrF++	50.98	69.28	64.24	57.80	60.58
Training Subset 3	STL-AraT5	BLEU	31.00	54.27	47.68	35.16	42.03
		chrF++	53.19	71.54	66.86	59.78	62.84
	STL-AraBART	BLEU	29.96	53.37	46.64	32.70	40.67
		chrF++	52.34	70.70	66.19	57.57	61.70
	MTL-AraT5	BLEU	**31.73**	**56.51**	**50.31**	**36.55**	**43.77**
		chrF++	**53.81**	**72.71**	**68.54**	**60.77**	**63.96**
	MTL-AraBART	BLEU	29.70	54.41	46.81	34.18	41.27
		chrF++	52.49	70.77	66.19	58.66	62.03

Table 5: The translation performance of STL vs. MTL models evaluated on the Dial2MSA-Verified test dataset, where the results of STL models on Training Subset 1 are from Khered et al. (2025)

In Training Subset 2, which includes only samples from the Dial2MSA-Verified training set and the new LahjaTube dataset, MTL models continue to outperform STL models, with MTL-AraT5 achieving the highest BLEU score of 42.71 and a chrF++ score of 63.27. Surprisingly, this model outperforms models trained on the larger Training Subset 1, highlighting the effectiveness of the LahjaTube dataset in improving DA-MSA translation.

The highest performance is observed in Training Subset 3, where MTL-AraT5, fine-tuned on all training datasets, achieves the best overall results, with a BLEU score of 43.77 and a chrF++ score of 63.96. This demonstrates that combining diverse datasets further improves translation quality.

6.2 Dialect Identification

Table 6 presents the classification performance of the ensemble MARBERT baseline, as reported in Khered et al. (2025), alongside the results of our proposed MTL models (MTL-AraT5 and MTL-AraBART) on the Dial2MSA-Verified test dataset. While MTL-AraBART consistently achieves the highest overall performance, a drop in the Macro-Average F1-score is observed in all MTL models compared to MARBERT. This drop is likely due to the training setup: MARBERT was trained exclusively on the four dialect classes, whereas the MTL models were trained on both the four dialects and MSA. Despite this, MTL-AraBART achieves the best results on other metrics, with an accuracy of 98.85% and a Weighted-F1 score of 99.10%, all obtained with Training Subset 2, which includes only the Dial2MSA-Verified-train and LahjaTube datasets. These results highlight that dialect identification also benefits from the MTL framework.

	Model	Acc	M-F1	W-F1
Training Subset 1	MARBERT	96.950	**96.942**	96.942
	MTL-AraT5	95.750	77.867	97.334
	MTL-AraBART	**97.275**	78.447	**98.059**
Training Subset 2	MTL-AraT5	98.213	78.943	98.678
	MTL-AraBART	**98.850**	**79.284**	**99.104**
Training Subset 3	MTL-AraT5	98.450	79.125	98.906
	MTL-AraBART	**98.688**	**79.273**	**99.091**

Table 6: Dialect identification performance of the ensemble MARBERT baseline and our proposed MTL models (MTL-AraT5 and MTL-AraBART), evaluated on the Dial2MSA-Verified test dataset using Accuracy (Acc), Macro-Average F1 (M-F1), and Weighted-Average F1 (W-F1) scores

6.3 Model Impact on Translation Quality

The results highlight the advantages of the MTL approach for DA-MSA translation, demonstrating consistent improvements over STL models. Translation performance improved when the weighting

parameter prioritised DA-MSA translation while still incorporating dialect identification (e.g., α = 0.5 or 0.8). In contrast, setting α to 0.3 resulted in a decline in performance, likely due to the classification task receiving greater importance, reducing the model's focus on translation. Among the architectures, MTL-AraT5 emerges as the most effective for DA-MSA translation, likely due to AraT5's pretraining on a more extensive and diverse Arabic dataset. Additionally, the results highlight the significance of the training dataset size and diversity, as larger and more varied training datasets enhance translation performance.

6.4 Error Analysis

To evaluate the impact of MTL on DA-MSA translation, we conducted a comparative analysis between the STL-AraT5 model and its multitask-enhanced version, MTL-AraT5. Both models usually produce similar translations, often differing by only one or two words. In many cases, these words had multiple valid translations, making it difficult to determine a single correct output. MTL-AraT5 consistently provides more contextually appropriate translations, likely due to the additional integration of dialect classification, which enhances the model's ability to differentiate and preserve dialect-specific meanings. However, misclassification occasionally affected translation performance. For example, when the model misclassified the input as MSA, it assumed no translation was needed, leading to the reproduction of the original dialectal sentence instead of converting it to MSA.

Despite MTL-AraT5 demonstrating improvements in handling idiomatic expressions and dialect-specific phrases, STL-AraT5 performed better in some instances, particularly in more straightforward lexical mappings. BLEU score comparisons reinforce these findings, indicating that while both models achieve comparable overall performance, MTL-AraT5 excels in dialect-sensitive contexts, whereas STL-AraT5 sometimes provides more direct and literal translations.

7 Conclusion and Future Work

This paper proposed an MTL framework for DA-MSA translation, integrating dialect identification as an auxiliary task. To support this research, we introduced LahjaTube, a new dataset of YouTube video transcripts covering four major Arabic dialects with their corresponding MSA and English translations. Our experiments with AraT5 and AraBART showed that MTL improves translation performance, particularly when LahjaTube is included in the training. MTL-AraT5 achieves the best overall translation performance, outperforming both STL models and MTL-AraBART, with a BLEU score of 43.77 and a chrF++ score of 63.96 when trained on the most comprehensive dataset (Training Subset 3). Meanwhile, MTL-AraBART consistently achieved the highest performance in dialect classification, reaching 98.85% accuracy and a weighted-F1 score of 99.10% in Training Subset 2. These results indicate that both tasks, DA-MSA translation and dialect identification, benefit from the MTL approach, as incorporating dialect identification helps improve translation quality while translation modelling enhances dialect classification. Despite these improvements, challenges remain in handling transliterated words, informal expressions, and code-switching. Additionally, optimising the balance between translation and classification tasks is an area for further research.

Building on our findings, future research can explore several directions to enhance DA-MSA translation. Expanding training data with additional dialectal resources and data augmentation techniques, can improve generalisation. Additionally, utilising large language models (LLMs) with decoder-only Transformer architecture, such as LLaMA, Gemma, and Jais, could improve DA-MSA translation by taking advantage of their strong language understanding and transfer learning abilities.

Limitations

Despite the promising results of our MTL framework, several limitations remain. Although LahjaTube introduces a new source of dialectal data, its coverage may be uneven, potentially underrepresenting certain countries within each dialectal region. While GPT-4o was used to generate MSA translations from English, most translations have not undergone manual verification, and only a small subset was reviewed through human evaluation; thus, some errors or inconsistencies may remain in the automatic MSA translations, which could reduce overall quality. Furthermore, although integrating dialect identification as an auxiliary task improves translation performance, misclassifying DA sentences as MSA can lead to incorrect outputs, with the model simply reproducing the input instead of providing a proper translation.

Ethical Considerations

The LahjaTube dataset consists of transcriptions from publicly available YouTube videos. To ensure ethical and legal compliance, we exclusively collected content licensed under Creative Commons, which permits reuse, including speech transcription for research purposes. Furthermore, we verified that the dataset does not include personal, sensitive, or harmful content. Moreover, the MSA translations were generated automatically from the English transcripts using the GPT-4o model. No manual correction was performed on the entire dataset; however, to assess the translation quality and support the reliability of LahjaTube, we conducted a human evaluation of a small subset of 200 DA-MSA translation pairs.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. 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 7088–7105, Online. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim A. Elmadany, Arun Rajendran, and Lyle H. Ungar. 2019. Dianet: BERT and hierarchical attention multi-task learning of fine-grained dialect. *CoRR*, abs/1910.14243.

Mansour Al Sulaiman, Abdullah M. Moussa, Sherif Abdou, Hebah Elgibreen, Mohammed Faisal, and Mohsen Rashwan. 2022. Semantic textual similarity for modern standard and dialectal arabic using transfer learning. *PLOS ONE*, 17(8):1–14.

Ali Alkhathlan and Khalid Alomar. 2024. Armt-tnn: Enhancing natural language understanding performance through hard parameter multitask learning in arabic. *Int. J. Know.-Based Intell. Eng. Syst.*, 28(3):483495.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2020. A multitask learning approach for diacritic restoration. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8238–8247, Online. Association for Computational Linguistics.

Meshrif Alruily. 2020. Issues of dialectal saudi twitter corpus. *The International Arab Journal of Information Technology*, 17:367–374.

Laith Baniata, Seyoung Park, and Seong-Bae Park. 2018a. A multitask-based neural machine translation model with part-of-speech tags integration for arabic dialects. *Applied Sciences*, 8:2502.

Laith Baniata, Seyoung Park, and Seong-Bae Park. 2018b. A neural machine translation model for arabic dialects that utilizes multitask learning (mtl). *Computational Intelligence and Neuroscience*, 2018.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The madar arabic dialect corpus and lexicon. In *International Conference on Language Resources and Evaluation*.

Shijie Chen, Yu Zhang, and Qiang Yang. 2024. Multitask learning in natural language processing: An overview. *ACM Comput. Surv.*, 56(12).

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Abdellah El Mekki, Abdelkader El Mahdaouy, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. BERT-based multi-task model for country and province level MSA and dialectal Arabic identification. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 271–275, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Salman Elgamal, Ossama Obeid, Mhd Kabbani, Go Inoue, and Nizar Habash. 2024. Arabic diacritics in the wild: Exploiting opportunities for improved diacritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14815–14829, Bangkok, Thailand. Association for Computational Linguistics.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Octopus: A multitask model and toolkit for Arabic natural language generation. In *Proceedings of ArabicNLP 2023*, pages 232–243, Singapore (Hybrid). Association for Computational Linguistics.

Erick Mendez Guzman, Viktor Schlegel, and Riza Batista-Navarro. 2024. Towards explainable multi-label text classification: A multi-task rationalisation framework for identifying indicators of forced labour. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 98–112, Miami, Florida, USA. Association for Computational Linguistics.

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. AraBART: a pretrained Arabic sequence-to-sequence model for abstractive summarization. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 31–42, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Abdullah Khered, Ingy Abdelhalim, Nadine Abdelhalim, Ahmed Soliman, and Riza Theresa Batista-Navarro. 2023. Unimanc at nadi 2023 shared task: A comparison of various t5-based models for translating arabic dialectical text to modern standard arabic. In *Proceedings of ArabicNLP 2023*, pages 658–664.

Abdullah Khered, Ingy Yasser Hassan Abdou Abdelhalim, and Riza Batista-Navarro. 2022. Building an ensemble of transformer models for Arabic dialect classification and sentiment analysis. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 479–484, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Abdullah Khered, Youcef Benkhedda, and Riza Batista-Navarro. 2025. Dial2MSA-verified: A multi-dialect Arabic social media dataset for neural machine translation to Modern Standard Arabic. In *Proceedings of the 4th Workshop on Arabic Corpus Linguistics (WACL-4)*, pages 50–62, Abu Dhabi, UAE. Association for Computational Linguistics.

Abhishek Kumar, Asif Ekbal, Daisuke Kawahra, and Sadao Kurohashi. 2019. Emotion helps sentiment: A multi-task model for sentiment and emotion analysis. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fatma Mallek, Billal Belainine, and Fatiha Sadat. 2017. Arabic social media analysis and translation. *Procedia Computer Science*, 117:298–303. Arabic Computational Linguistics.

Mamta, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Exploring multi-lingual, multi-task, and adversarial learning for low-resource sentiment analysis. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 21(5).

Karima. Meftouh, Salima Harrat, and Kamel Smaïli. 2018. PADIC: extension and new experiments. In *7th International Conference on Advanced Technologies ICAT*, Antalya, Turkey.

Youness Moukafih, Nada Sbihi, Mounir Ghogho, and Kamel Smaili. 2021. Improving machine translation ofnbsp;arabic dialects through multi-task learning. In *AIxIA 2021 Advances in Artificial Intelligence: 20th International Conference of the Italian Association for Artificial Intelligence, Virtual Event, December 13, 2021, Revised Selected Papers*, page 580590, Berlin, Heidelberg. Springer-Verlag.

Hamdy Mubarak. 2018. Dial2msa: A tweets corpus for converting dialectal arabic to modern standard arabic. In *OSACT 3: The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools*, pages 49–53.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Hai Pham, Young Jin Kim, Subhabrata Mukherjee, David P. Woodruff, Barnab,s Pczos, and Hany Hassan Awadalla. 2023. Task-based moe for multitask multilingual machine translation. *CoRR*, abs/2308.15772.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63:1872–1897.

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.

Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. 2014. Automatic identification of arabic dialects in social media. SoMeRA '14, page 3540, New York, NY, USA. Association for Computing Machinery.

Saudi Sadiq. 2025. Evaluating english-arabic translation: Human translators vs. google translate and chatgpt. *Journal of Languages and Translation*, 12(1):67–95.

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. Sentence level dialect identification for machine translation system selection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 772–778, Baltimore, Maryland. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 656–661, Melbourne, Australia. Association for Computational Linguistics.

Yu Zhang and Qiang Yang. 2017. An overview of multi-task learning. *National Science Review*, 5(1):30–43.

Low-Resource Machine Translation for Moroccan Arabic

Alexei Rosca **Abderrahmane Issam** **Gerasimos Spanakis**
Department of Advanced Computing Sciences
Maastricht University
{alexei.rosca, abderrahmane.issam, jerry.spanakis}@maastrichtuniversity.nl

Abstract

Neural Machine Translation (NMT) has achieved significant progress especially for languages with large amounts of data (referred to as high resource languages). However, most of the world languages lack sufficient data and are thus considered as low resource or endangered. Previous research explored various techniques for improving NMT performance on low resource languages, with no guarantees that they will perform similarly on other languages. In this work, we explore various low resource NMT techniques for improving performance on Moroccan Arabic (Darija), a dialect of Arabic that is considered a low resource language. We experiment with three techniques that are prominent in low resource Natural Language Processing (NLP), namely: back-translation, paraphrasing and transfer learning. Our results indicate that transfer learning, especially in combination with back-translation is effective at improving translation performance on Moroccan Arabic, achieving a BLEU score of 26.79 on Darija→English and 9.98 on English→Darija. [1]

1 Introduction

Neural Machine translation (NMT) has achieved impressive results for high resource languages, supported by extensive linguistic data and resources that facilitate model training and optimization (Johnson et al., 2017; Vaswani et al., 2017). However, due to limited data availability and inherent linguistic complexity, significant performance gaps remain for low-resource languages (Lakew et al., 2020). This work addresses this critical gap by developing MT solutions for Moroccan Arabic (commonly referred to as Darija), a dialect characterized by unique linguistic features and a lack of training data. By focusing on Moroccan Arabic, this work seeks to advance natural language processing (i.e. NLP) methods for underrepresented languages and contribute to a more inclusive landscape in machine translation.

The main objective of this paper is build a Darija-English translation model using state-of-the-art low resource NMT techniques. We experiment with three techniques that have been shown to be effective in low resource scenarios (Haddow et al., 2022), namely: back-translation, paraphrasing and transfer learning. More specificaly, we apply back-translation by training a model to translate from the target to the source language and using it to generate translations. We then translate monolingual target sentences to the source language then use them for training. For paraphrasing, we use a paraphrasing model to generate synthetic copies of the input sentence and use them to augment the training set. In combination with paraphrasing and back-translation, we experiment with transfer learning from publicly available pretrained models. We fine-tune a multilingual model that supports Moroccan Arabic, and given that Arabic is the mother language of Moroccan Arabic, we also fine-tune a bilingual model that is trained for Arabic-English translation. Furthermore, we compose an encoder-decoder translation model from the checkpoints of a BERT model pretrained on Moroccan Arabic and an Arabic to English translation model and we fine-tune it for Darija→English translation.

Our results demonstrate the effectiveness of transfer learning on training an NMT model for Moroccan Arabic, especially when combined with back-translation, achieving a BLEU score of 26.79 on Darija→English and 9.98 on English→Darija. However, when evaluating on datasets from a different domain, we find the improvements are unstable, which questions the generalization of these techniques to other domains. Furthermore, we observe a significant disparity between translation directions: translating into English from Darija achieves more than twice the performance of translating

[1] https://github.com/RoscaAlex00/lowresource_mt

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 32–38, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-005

from English→Darija. This calls for further research into democratizing NLP for low resource languages such as Moroccan Arabic.

2 Related Works

Moroccan Arabic NLP: NLP resources for Moroccan Arabic are characterized by scarcity. Previous work introduced resources for different NLP tasks (Samih and Maier, 2016; Issam and Mrini, 2021; Boujou et al., 2021; Moussa and Mourhir, 2023). For machine translation, Tachicart et al. (2014) introduce the Moroccan Dialect Electronic Dictionary (MDED): a bilingual dictionary for Moroccan and Modern Standard Arabic (MSA). Mrini and Bond (2017) introduce Moroccan Darija WordNet (MDW): an extension of the Open Multilingual WordNet (Bond and Foster, 2013) to Moroccan Arabic. Unfortunately, these works are limited to word level translations. Outchakoucht and Es-Samaali (2021) introduce Darija Open Dataset (DODa), a collaborative dataset of word and sentence level translations between Darija and English. In this work, we leverage this dataset for training and evaluating machine translation models.

Low Resource NMT: Low resource NMT is an active area of research with real-world impact. Various techniques were introduced to deal with data and resource scarcity and were shown to be effective (Haddow et al., 2022). The most straightforward technique is data augmentation (Feng et al., 2021), where rule based or neural based techniques can be used to generate more data that can be used for training. In this work, we study two successful data augmentation techniques, namely, back-translation (Sennrich et al., 2016, 2017; Hoang et al., 2018; Edunov et al., 2018) and paraphrasing (Callison-Burch et al., 2006; Wang et al., 2016; Mallinson et al., 2017). Back-translation translation target sentences to the source language using an available model, while paraphrasing creates synthetic copies of the source sentences. Furthermore, previous work has shown the effectiveness of transfer learning especially in low resource scenarios (Zoph et al., 2016; Howard and Ruder, 2018; Devlin et al., 2019), where a model is pretrained on large amounts of data and fine-tuned for a target task or domain. We similarly leverage pretrained models in combination with data augmentation and evaluate their performance when fine-tuned on Moroccan Arabic.

3 Methodology

3.1 Back-Translation

Back-translation is a frequently used technique for data augmentation in machine translation (Sennrich et al., 2016, 2017; Hoang et al., 2018; Edunov et al., 2018). It helps to overcome the lack of parallel corpora, particularly for languages with limited resources such as Moroccan Arabic. This method takes advantage of the greater availability of the target language (i.e. English) to generate new synthetic sentence pairs. The process begins with the training of a target-to-source model (i.e. English-to-Moroccan Arabic), which is used to translate the target language text into the source language text. This newly generated dataset is then included in the training process of the source-to-target translation model.

3.2 Paraphrasing

Similar to back-translation, paraphrasing aims at augmenting the training data and diversifying the linguistic structure and vocabulary of the input sentences (Callison-Burch et al., 2006; Wang et al., 2016), while preserving their original meaning. Paraphrasing can be achieved either using rule based techniques such as synonym replacement or using neural models. Previous work shows that neural based models generate better paraphrases (Mallinson et al., 2017). In this work, we generate paraphrases of the source sentences using BART-paraphrase [2] which is a BART model (Lewis et al., 2020) fine-tuned for paraphrasing. We use this model to paraphrase the English sentences either on source or the target side depending on the translation direction (i.e. English→Darija or Darija→English). A single paraphrase for each example is added to the dataset.

3.3 Transfer Learning

Transfer learning leverages knowledge learned during pretraining to improve performance on closely related tasks. It has been particularly effective in low resource scenarios, since the pretrained model is often trained on larger amount of data than is available in the target task. We similarly apply transfer learning by fine-tuning models that are trained on massive amounts of data on translating Moroccan Arabic.

[2] https://huggingface.co/eugenesiow/bart-paraphrase

4 Experiments

4.1 Datasets

For training and evaluation, we use the Darija Open Dataset (i.e. DODa) (Outchakoucht and Es-Samaali, 2021, 2024). It is of the largest dataset for Darija-English translation with more than 45000 translation pairs. We split this dataset randomly into a training, test and validation set in a ratio of 80%, 15% and 5% respectively.

To assess the generalization of our experiments, we evaluate the models on two different datasets that contain Darija-English translation pairs. Specifically, we use translations from the New Testament that were collected for Moroccan Arabic (Sajjad et al., 2020) which we refer to as BIBLE, and MADAR (Bouamor et al., 2018), which is a dataset that contains translations between English and 26 Arabic dialects including Moroccan Arabic. These two test sets contain 500 and 5500 examples respectively.

For both backtranslation and paraphrasing, we generate one copy of the training set and include it in the training.

4.2 Models

No Language Left Behind (NLLB) (Team et al., 2022): NLLB is a massively multilingual model that supports more than 200 languages including Moroccan Arabic. We fine-tune this model on Darija→English and English→Darija. We use the small distilled version of NLLB [3] for both directions.

OPUS-MT (Tiedemann and Thottingal, 2020): Is an open source initiative that has released a collection of resources and models. Although there is no OPUS-MT translation model that supports Moroccan Arabic, there are models that support Arabic. Since Arabic is the mother language of Darija, we experiment with fine-tuning OPUS-MT English to Arabic (i.e. OPUS-MT-En-Ar [4]) and Arabic to English (i.e. OPUS-MT-Ar-En [5]) models on translating between English and Darija.

Encoder-decoder checkpointing (Rothe et al., 2020): We experiment with composing a translation encoder-decoder from different pretrained encoder and decoder checkpoints . Specifically, we use a BERT (Devlin et al., 2019) model that is pretrained for Moroccan Arabic (i.e. DarijaBERT [6] (Gaanoun et al., 2023)) to initialize the encoder, and the decoder part of OPUS-MT Arabic to English to initialize the decoder. Although the two models are different, previous research (Rothe et al., 2020) has shown that this can be effective for fine-tuning.

For training and evaluation details, please see Appendix A.

5 Results and Discussion

5.1 Main Results

Table 1 and 2 show the BLEU score results of our experiments on Darija→English and English→Darija respectively on DODa, BIBLE and MADAR test sets (we include chrF score results in Appendix B). We first notice a significant disparity between the performance on Darija→English versus English→Darija, especially of NLLB, which is significantly better at translating to English than to Darija. Furthermore, the results show that fine-tuning consistently improves performance over the base model especially on the in-domain DODa test set. However, when looking at BIBLE and MADAR test sets, we notice that fine-tuning negatively affects the performance of NLLB on Darija→English translation.

Back-translation leads to better results than paraphrasing especially when translating Darija→English. Paraphrasing is even worse than fine-tuning on this direction. In the English-to-Darija direction, paraphrasing consistently outperforms fine-tuning. We attribute this disparity to the direction of paraphrasing. In the case of Darija→English translation, paraphrasing is applied to the target sentences, which can negatively impact model outputs. In contrast, paraphrasing the source sentences, as in English→Darija, tends to be more robust and beneficial.

BERT-OPUS achieves the best results on DODa Darija→English translation, with a slight improvement over NLLB. This is still significant given that BERT-OPUS is smaller than NLLB (i.e. 200M parameters in BERT-OPUS compared to 600M parameters in NLLB), and is trained on significantly less data. This shows the advantage of training language specific models, where DarijaBERT is pretrained on Darija sentences mined from the web.

Although NLLB supports Darija, the results of translating English→Darija are very low, even lower than 1 BLEU point on DODa and BIBLE test

[3] https://huggingface.co/facebook/nllb-200-distilled-600M
[4] https://huggingface.co/Helsinki-NLP/opus-mt-en-ar
[5] https://huggingface.co/Helsinki-NLP/opus-mt-ar-en
[6] https://huggingface.co/SI2M-Lab/DarijaBERT

Model	Dataset	Base	FT	Para	BT
NLLB	DODa	8.66	26.50	20.57	**26.65**
	BIBLE	**20.43**	13.53	11.84	13.48
	MADAR	**29.31**	27.44	28.27	28.19
OPUS-MT	DODa	2.03	14.39	10.52	**15.89**
	BIBLE	4.05	4.30	4.42	**4.80**
	MADAR	7.03	13.81	**16.56**	15.32
BERT-OPUS	DODa	0.00	26.78	20.54	**26.79**
	BIBLE	0.05	1.94	1.46	**2.32**
	MADAR	0.01	15.53	15.87	**17.33**

Table 1: We provide the BLEU score results on Darija→English. *Base* shows the results of the pretrained model, *FT* the results after fine-tuning, *Para* the results of fine-tuning on paraphrased dataset, and *BT* shows the results of fine-tuning on the dataset with back-translated data.

sets. Fine-tuning is effective especially on DODa test set, increasing BLEU score by more than 7 BLEU points. Fine-tuning OPUS-MT is not as effective as fine-tuning NLLB (2.58 vs 8.10 after fine-tuning respectively). This illustrates the effectiveness of multilingual pretraining, while OPUS-MT-En-Ar struggles to generalize to Moroccan Arabic given its divergence from MSA.

Model	Dataset	Base	FT	Para	BT
NLLB	DODa	0.82	8.10	9.68	**9.98**
	BIBLE	0.04	0.34	0.82	**0.94**
	MADAR	4.42	5.89	4.67	**6.63**
OPUS-MT	DODa	0.25	2.58	5.02	**5.11**
	BIBLE	**0.35**	0.30	0.21	0.29
	MADAR	1.30	2.05	2.16	**3.20**

Table 2: We provide the BLEU score results on English→Darija. *Base* shows the results of the pretrained model, *FT* the results after fine-tuning, *Para* the results of fine-tuning on paraphrase dataset, and *BT* shows the results of fine-tuning on the dataset with back-translated data.

5.2 Discussion

Disparity between Darija→English and English→Darija performance: There is a significant performance disparity between Darija→English and English→Darija, where the best BLEU score performance on Darija→English is more than 16 BLEU points higher than the performance on English→Darija. We expalin this independently for the two models as follows: In the case of NLLB, we attribute this discrepancy to the amount of English data compared to Darija data. NLLB was trained on significantly more English than Darija data or even MSA data, which makes translating into English easier, this can be seen in the difference in performance of the Base model on the two directions. In the case of OPUS-MT, we explain the performance by the linguistic difference between MSA and Darija, where the decoder of OPUS-MT-EN-AR is trained on generating MSA and struggles to translate into Darija. This means that even after fine-tuning the model will struggle to learn to generate in a new vocabulary and linguistic structure.

Out of distribution generalization: We find that fine-tuning on DODa dataset lacks generalization on the BIBLE dataset, while the performance on MADAR in general improves except for NLLB on Darija→English (Table 1). We attribute this to the fact that MADAR sentences are closely similar to DODa sentences, while BIBLE data is significancy different in domain, and uses a high number of rare MSA words that are not used in Darija due to its vernacular nature, which we suggest explains the significant decrease in performance of NLLB on BIBLE after fine-tuning on DODa (Table 1).

6 Conclusion

In this work, we apply three low resource techniques to train machine translation models for English and Moroccan Arabic. Namely, we experiment with paraphrasing, back-translation and transfer learning. Our results show that combining back-translation with transfer learning achieves the best results, especially when fine-tuning a massively multilingual model such as NLLB, or an encoder that is pretrained on the source language such as DarijaBERT. Furthermore, our results raise concerns about the generalization of these techniques to out-of-domain datasets such as the BIBLE, where fine-tuning can even degrade the performance. Across all the techniques and models, we see a significant disparity between the performance on translating to English versus translating from it to Darija. Overall, our work contributes to the understanding of low-resource MT strategies in real-world scenarios and highlights the need for more equitable approaches to multilingual NLP. Future work should focus on improving robustness to domain shift, developing techniques that work well in both translation directions and investing in better resources and benchmarks for dialectal and underrepresented languages like Moroccan Darija.

Broader Impact

This work contributes to the broader goal of making language technologies more inclusive by advancing machine translation for low-resource languages and dialects such as Moroccan Darija. MT plays an important role in enabling access to information, public services, education and communication, especially in linguistically diverse regions where speakers may have limited proficiency in high-resource languages such as English or even MSA. For many speakers of Moroccan Darija, MT systems can support everyday tasks such as understanding online content, communicating across language barriers and participating in digital platforms that otherwise would be inaccessible. By evaluating practical low-resource MT techniques and revealing key challenges such as out-of-domain generalization and translation direction asymmetry, our work encourages the development of more robust and equitable NLP systems.

At the same time, it is essential to recognize limitations of current models and ensure that MT systems are used responsibly, especially in high-stakes domains like healthcare, law, and public policy where human oversight is critical. Finally, evaluating translation quality solely through automated metrics remains a limitation, therefore future work should include human evaluations by native Darija speakers to better assess usefulness, fluency and cultural relevance.

References

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and 1 others. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

ElMehdi Boujou, Hamza Chataoui, Abdellah el Mekki, Saad Benjelloun, Ikram Chairi, and Ismail Berrada. 2021. An open access nlp dataset for arabic dialects : Data collection, labeling, and model construction.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*, pages 17–24. 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.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Kamel Gaanoun, Abdou Mohamed Naira, Anass Allak, and Imade Benelallam. 2023. Darijabert: a step forward in nlp for the written moroccan dialect.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. Association for Computational Linguistics. 2nd Workshop on Neural Machine Translation and Generation, WNMT 2018 ; Conference date: 15-07-2018 Through 20-07-2018.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Abderrahmane Issam and Khalil Mrini. 2021. Goud.ma: a news article dataset for summarization in moroccan darija. In *3rd Workshop on African Natural Language Processing*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, and 1 others. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Surafel M Lakew, Stephan Gouws, Yulia Tsvetkov, Vukosi Marivate, and Stefan Weber. 2020. Low-resource neural machine translation: A benchmark for five african languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2602–2611.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jonathan Mallinson, Rico Sennrich, and Maria Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893. Association for Computational Linguistics (ACL). The 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 ; Conference date: 03-04-2017 Through 07-04-2017.

Hanane Nour Moussa and Asmaa Mourhir. 2023. Darnercorp: An annotated named entity recognition dataset in the moroccan dialect. *Data in Brief*, 48:109234.

Khalil Mrini and Francis Bond. 2017. Building the moroccan darija wordnet (mdw) using bilingual resources.

Aissam Outchakoucht and Hamza Es-Samaali. 2021. Moroccan dialect -darija- open dataset. *Preprint*, arXiv:2103.09687.

Aissam Outchakoucht and Hamza Es-Samaali. 2024. The evolution of darija open dataset: Introducing version 2. *Preprint*, arXiv:2405.13016.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Hassan Sajjad, Ahmed Abdelali, Nadir Durrani, and Fahim Dalvi. 2020. Arabench: Benchmarking dialectal arabic-english machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5094–5107.

Younes Samih and Wolfgang Maier. 2016. An Arabic-Moroccan Darija code-switched corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4170–4175, Portorož, Slovenia. European Language Resources Association (ELRA).

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ridouane Tachicart, Karim Bouzoubaa, and Hamid Jaafar. 2014. Building a moroccan dialect electronic dictionary (mded).

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Pidong Wang, Preslav Nakov, and Hwee Tou Ng. 2016. Source language adaptation approaches for resource-poor machine translation. *Computational Linguistics*, 42(2):277–306.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Training and Evaluation

In all experiments, we keep the same data split to ensure the robustness of the results. We experimentally tune the hyperparameters of each model. In Table 3, we list the final hyperparameters we used for fine-tuning the models in both directions (i.e. Darija→English and English→Darija). We use HuggingFace transformers [7] library for training

[7] https://huggingface.co/docs/transformers

and evaluation. For more details, we release our code publicly [8].

Hyperparameter	NLLB	OPUS-MT	BERT-OPUS
Learning Rate	1e-5	1e-6	8e-5
Batch Size	4	16	16
Weight Decay	0.01	0.01	0.0
Number of Epochs	3	5	7
Warmup Steps	0	0	500

Table 3: Hyperparameters for fine-tuning each model.

B chrF Results

In this section, we provide the results of using chrF metric to compute translation performance in Table 4 and 5.

Model	Dataset	Base	FT	Para	BT
NLLB	DODa	31.35	43.24	43.63	**44.68**
	BIBLE	**42.54**	36.69	34.32	37.08
	MADAR	**47.43**	45.66	45.67	46.70
OPUS-MT	DODa	20.03	33.31	34.02	**35.32**
	BIBLE	**25.94**	24.17	23.01	25.39
	MADAR	25.46	31.58	32.89	**32.94**
BERT-OPUS	DODa	5.06	44.24	42.78	**44.80**
	BIBLE	12.38	19.52	17.01	**20.41**
	MADAR	6.89	33.29	32.95	**35.27**

Table 4: We provide the chrF score results on Darija→English. *Base* shows the results of the pre-trained model, *FT* the results after fine-tuning, *Para* the results of fine-tuning on paraphrase dataset, and *BT* shows the results of fine-tuning on the dataset with back-translated data.

Model	Dataset	Base	FT	Para	BT
NLLB	DODa	14.98	32.49	35.05	**35.59**
	BIBLE	13.24	19.57	24.41	**25.92**
	MADAR	26.27	33.50	32.40	**35.35**
OPUS-MT	DODa	13.27	22.55	26.48	**26.74**
	BIBLE	**20.84**	19.56	17.32	20.77
	MADAR	21.54	23.17	24.34	**26.16**

Table 5: We provide the chrF score results on English→Darija. *Base* shows the results of the pre-trained model, *FT* the results after fine-tuning, *Para* the results of fine-tuning on paraphrase dataset, and *BT* shows the results of fine-tuning on the dataset with back-translated data.

[8] https://github.com/RoscaAlex00/lowresource_mt

Efficient Architectures For Low-Resource Machine Translation

Edoardo Signoroni
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno, Czechia
e.signoroni@mail.muni.cz

Pavel Rychlý
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno, Czechia
pary@fi.muni.cz

Ruggero Signoroni
Department of Information Engineering
University of Brescia
Via Branze 38, Brescia, Italy
r.signoroni001@studenti.unibs.it

Abstract

Low-resource Neural Machine Translation is highly sensitive to hyperparameters and needs careful tuning to achieve the best results with small amounts of training data. We focus on exploring the impact of changes in the Transformer architecture on downstream translation quality, and propose a metric to score the computational efficiency of such changes. By experimenting on English-Akkadian, German-Lower Sorbian, English-Italian, and English-Manipuri, we confirm previous finding in low-resource machine translation optimization, and show that smaller and more parameter-efficient models can achieve the same translation quality of larger and unwieldy ones at a fraction of the computational cost. Optimized models have around 95% less parameters, while dropping only up to 14.8% ChrF. We compile a list of optimal ranges for each hyperparameter.

1 Introduction

Neural machine translation (NMT) has done massive progress in high-resource conditions, due to the performance of models based on encoder-decoder architectures, such as the Transformer (Vaswani et al., 2017). Often, this progress did not trickle down to low or extremely low-resource languages, due to the huge requirements in terms of available training data and computational resources (Ranathunga et al., 2023). Default settings and assumptions for high-resource scenarios, such as the correlation of model size and performance, are not true in a low-resource one. While some attempts are being done to adapt and prompt large language models (LLMs) for low-resource machine translation (Guo et al., 2024; Lu et al., 2024; Merx et al., 2024; Aycock et al., 2025; Joshi et al., 2025; Khade et al., 2025), handling them is not always feasible nor convenient. Suitable hardware may not be available to deploy or train sufficiently large models. Moreover, even if capable hardware is available, using LLMs may still be a suboptimal choice, since it is much harder to freely fit the model and its architecture to the scarce data. Moreover, as Petrov et al. (2023) shows, LLMs' own pre-trained tokenizers are biased against low-resource languages due to their low share of training data. Considering that even state-of-the-art LLMs underperform standard NMT (Robinson et al., 2023), employing them when smaller options are available is inefficient.

Training a Transformer in these settings remains a challenging task, and one that requires careful hyperparameter tuning (Popel and Bojar, 2018). However, if done correctly, it can lead to well-performing and competitive models (van Biljon et al., 2020; Araabi and Monz, 2020). Most of the work regarding low-resource machine translation focuses on several techniques, such as fine-tuning, or transfer learning (Ranathunga et al., 2023). Research on scaling and optimizing machine translation has mainly been done in a high-resource setting (Ghorbani et al., 2022), or on other aspects of training (Sennrich and Zhang, 2019; Araabi and Monz, 2020; Signoroni and Rychlý, 2024).

Following the finding that not only size, but also shape of the Transformer influences downstream performance (Tay et al., 2022), our work aims to broaden the understanding of the scaling of machine translation in low-resource settings by experimenting with four key components in the architecture of the Transformer model: encoder layers, decoder layers, embedding size, and feedforward dimension. We conduct experiments on one simulated low-resource pair, and three true low-resource

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 39–64, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-006

pairs, to explore the impact of each hyperparameter on the downstream translation task. We propose a novel **P**arameter **I**ncrease **E**fficiency **Score** (PIES) to measure the efficiency of changing the configuration of the model, and to find the most parameter-efficient combinations for each dataset. We compile a list of empirically-found optimal ranges for each hyperparameter to inform future exploration and training of low-resource machine translation models.

We confirm that in low-resource conditions the Transformer is highly susceptible to hyperparameter variation. We also find that smaller models can perform as well as much bigger models, at just a tiny fraction of the computational cost.[1]

2 Related Work

Our work intersects previous studies on Transformer and Machine Translation scaling laws and optimization on both high and low-resource languages.

2.1 Scaling Laws and Optimization

Works tackled the challenge of finding empirical scaling laws that govern neural language model scaling, considering model, computational, or dataset size.

Tay et al. (2022) conduct extensive experiments involving over 200 Transformer configurations considering both upstream and several downstream tasks (though, crucially, not machine translation). They find that model shape, and not only size (Kaplan et al., 2020), strongly influences downstream performance. They also find that scaling laws change substantially when considering metrics on actual downstream fine-tuning. Notably, they show that scaling strategies differ at different compute regions, and thus finding strategies at small scale might not necessarily transfer or generalize to higher compute regions.

Some work has also been conducted for machine translation.

Ghorbani et al. (2022) explore scaling laws for machine translation on a high-resource English-German dataset. Their results indicate that the scaling behavior is largely determined by the total capacity of the model, and its allocation between the encoder and the decoder. Moreover, they suggest that scaling behavior of encoder-decoder NMT models is predictable, but the scaling laws might vary depending on the particular architecture or task.

Gordon et al. (2021) study the predictability of MT system performance as parameters/data increase. They train many Transformers of various sizes randomly selected subsets of data for Russian-English, German-English, and Chinese-English. Crucially, they find that extending their previous experiments to datasets smaller than 50MB, using 0.05% - 0.0125% of the data, the data scaling power law breaks down, indicating the impossibility of extrapolating extremely low-resource results to medium and high-resource data regimes.

Some research (Hsu et al., 2020; Kasai et al., 2021; Berard et al., 2021) has also departed from the convention of using balanced encoder and decoders, resulting in "deep encoder, shallow decoder" models that can speed up inference while maintaining a similar translation performance.

2.2 Optimization for Low-Resource Settings

Some studies have also been done on optimizing NMT for low-resource scenarios.

Sennrich and Zhang (2019) find that best practices differ between high-resource and low-resource MT and that the latter is highly sensitive to hyperparameters by training RNNs with different techniques and hyperparameters on a simulated English-German, and a true Korean-English low-resource dataset.

Araabi and Monz (2020) trains Transformers for a diverse set of true and simulated low-resource pairs to find that a proper combination of Transformer configurations results in substantial improvements over a Transformer system with default settings. For example, they observe that a shallower Transformer combined with a smaller feed-forward layer dimension and two attention heads is more effective.

van Biljon et al. (2020) experiment with different Transformer configurations on the translation of three low-resource languages, showing that medium (6 total layers) and shallow (2 total layers) perform better than the canonical configuration of 6 encoder and 6 decoder layers.

[1]Full results and code is available at https://github.com/edoardosignoroni/eff_archs_lowre

3 Methodology

This section describes the dataset we tested on (Section 3.1)[2]. It then reports the training framework and the hyperparameters we used (Section 3.2). Next, it explains our proposed efficiency metric (Section 3.3). And finally, it outlines our experimental setup (Section 3.4).

3.1 Datasets

Our experiments are carried out on publicly available low-resource datasets, and one simulated low-resource dataset retrieved from OPUS (Tiedemann, 2009). The datasets involve both high-resource languages (English, German, Italian), and a selection of under-resourced languages (Akkadian, Lower Sorbian, Manipuri). The datasets have between 21k and 50k sentence pairs, thus can be considered as extremely low-resource (Ranathunga et al., 2023). Their content is from different domains, mainly news and Wikipedia text, except for Akkadian, which is mostly assorted fragments of cuneiform texts. The low-resource datasets have their own validation and test splits, while for the simulated English-Italian dataset we use the *dev* and *devtest* splits from the Flores-200 benchmark corpus (Goyal et al., 2022). The datasets are summarized in Table 1.[3]

3.2 Hyperparameters and Training

After tokenizing the data using BPE (Sennrich et al., 2016), as implemented in SentencePiece (Kudo and Richardson, 2018), we learn separated vocabularies for source and target with a size of 4k items, without a frequency threshold.

We train Transformers (Vaswani et al., 2017) with Fairseq (Ott et al., 2019) until BLEU score on validation does not increase for 20 consecutive epochs or until 50000 updates. As our baseline, we chose a *small* model that performed sufficiently well in previous experiments for all pairs. Its architecture and training hyperparameters are given in Table 2. We share embeddings between the encoder and the decoder. Each model is trained on a single Nvidia A40 or A100 GPU.

During the experiments, we focus on tuning the architecture of the model by changing the number of encoder and decoder layers, the size of the embeddings, and the feed forward dimension. We leave all other hyperparameters unchanged. We leave the number of heads at 2, following Araabi and Monz (2020).

We will refer to the models with the following naming scheme: *enc_dec_embs_ffw_heads*. E.g. our baseline model may be referred as *4_4_256_1024_2*.

3.3 Efficiency Score

To evaluate the efficiency of the models, we introduce a **P**arameter **I**ncrease **E**fficiency **S**core, or **PIES**, computed as follows:

$$PIES = \frac{size/score}{10^6}$$

where score means a machine translation metric such as COMET, ChrF, or BLEU, and size means the total number of parameters of the model. Thus, PIES is computed as the ratio between the size of the model in number of parameters and the machine translation score it achieved, divided by 1 million. This is an easily interpretable and straightforward metric that gives the millions of parameters needed for each score point. A lower value denotes a more efficient system.

We compute the total number of parameters for each model as follows:

$$params = (2 \times E \times V) + (4 \times E^2 + 2 \times E \times F + 9 \times E + F) \times enc + (8 \times E^2 + 2 \times E \times F + 15 \times E + F) \times dec$$

where E is the size of the embeddings, V the number of items in the vocabulary, F is the feed-forward dimension, and *enc/dec* is the number of layers in the encoder/decoder, respectively.

To obtain the score for each model after training, we generate test set translations for each model and obtain sentence-level BLEU (Papineni et al., 2002), ChrF (Popović, 2015), ChrF++ (Popović, 2017), and COMET (Rei et al., 2020) scores as implemented in Hugging Face `evaluate` library.[4] We employ bootstrap evaluation on 200 batches of 400 test sentences to obtain the final scores.

Mathur et al. (2020) argue for the retirement of BLEU in favour of ChrF++. Sai B et al. (2023)

[2]Appendix A provides more information about the languages involved.

[3]We use a simple Python script to split the tokenized data at the newline character and the whitespace and then return the length of the resulting lists to obtain the number of lines and tokens for each pair.

[4]Scores for metrics other than ChrF are available in the Appendices (Tables 17-23) and in the GitHub repository.

Languages	Abbreviation	Dataset	N. of Pairs	Src Tokens	Tgt Tokens
English-Akkadian	eng-akk	EvaCun 2023	45,269	1,177,138	630,535
German-Lower Sorbian	deu-dsb	WMT22 Low-res shared Task	40,194	1,064,087	1,032,701
English-Italian	eng-ita	WikiMatrix Random Selection	50,000	1,571,843	1,723,391
English-Manipuri	eng-mni	WMT23 Indic Shared Task	21,287	748,407	715,548

Table 1: **Summary of the datasets in our experiments.** The columns report the languages in the dataset, its original source, and the size of the training split in number of tokens and sentence pairs.

Parameters	
vocabulary size	4,000
encoder layers	4
decoder layers	4
enc/dec embedding dim	256
enc/dec feed forward dim	1,024
enc/dec attention heads	2
optimizer	adam
adam betas	0.9, 0.98
learning rate	1e-4
warmup updates	5,000
dropout	0.1
label smoothing	0.1
max tokens	16,000

Table 2: **Hyperparameters for our baseline model.** For the other models in our experiments, we change only the number of layers, the size of the embeddings, and the feed forward dimension.

finds that ChrF++ performs the best among overlap metrics for a selection of Indic languages. The results of recent WMT Metrics shared tasks (Freitag et al., 2022) demonstrate that learned neural metrics are the most optimal. Among these, COMET is the current state-of-the-art, and is widely employed in machine translation studies. However, pretrained neural metrics are unreliable for unseen languages, especially under-resourced ones. Works such as the ones by Sai B et al. (2023) and Wang et al. (2024) show that fine-tuned COMET models perform better for specific sets of low-resource languages, than baseline models. For these reasons we chose ChrF as the metric of reference in both our observations and PIES.

By computing Pearson's r between PIES and ChrF score on the aggregate results of our experiments, we obtain r=-0.543, indicating a negative correlation between PIES and translation quality: a lower PIES corresponds to a higher ChrF.

3.4 Experiments

Our aim is to investigate efficient architectures for low-resource machine translation models by tuning hyperparameters such as encoder and decoder layers, embeddings and feed forward dimension. We fix all other training hyperparameters to values found to be optimal or close to optimal in previous and preliminary experiments on the same data (Signoroni and Rychlý, 2024).

3.4.1 Experiment 1: Change One, Fix All

Hyperparameters	
encoder layers	2, **4**, 6, 8, 12, 16, 24, 32
decoder layers	2, **4**, 6, 8, 12, 16, 24, 32
embedding dimension	**256**, 512, 1024, 2048, 4096
feed forward dimension	256, 512, **1024**, 2048, 4096

Table 3: **Values for each hyperparameter tried in Experiment 1.** Baseline values are in bold.

Size	Hyperparameters	N. of Parameters
small	4_4_256_1024_2	8.5M < **9.4M** < 10M
base	6_6_512_2048_2	40M < **48M** < 53M
large	6_6_1024_4096_2	166M < **184M** < 203M

Table 4: **Baseline hyperparameters and sizes (in bold) for the models in Experiment 2.** We consider all possible architectures in a range of $\pm 10\%$ parameters from these baseline models.

Our first experiment focuses on changing only one hyperparameter at a time in the architecture of the model without controlling the total amount of parameters. We start from our baseline values of 4 encoder and decoder layers, embedding size of 256, and feedforward dimension of 1024, and change them one step at a time according to Table 3.

3.4.2 Experiment 2: Parameters Budget

In Experiment 2, we fix the number of parameters to $\pm 10\%$ of transformer small, base, and large and test all possible combinations of hyperparameters

that fall into the ranges given in Table 4. For each dataset, we test each possible configuration that falls within these ranges: 13 for *small* (counting the baseline 4_4_256_1024_2 model), 58 for *base*, and 60 for *large*, that is 131 combinations for dataset, for a total of 524 models. By allowing all possible combinations of hyperparmeters, we overcome one limitation of the previous setup, that is the chance of missing possible optimal configurations due to changing only one hyperparameter at a time.

4 Results

4.1 Experiment 1: Change One, Fix All

As we discuss the results of our experiments, recall that a lower PIES denotes a more efficient model.

In Experiment 1, we start from the baseline 4_4_256_1024_2 model and increase or decrease only one hyperparameter at a time, leaving all other unchanged. Table 5 summarizes the results of Experiment 1.

As expected, increasing the embedding size leads to the biggest increase in model size, since it scales quadratically with the amount of parameters. Conversely, all the other hyperparameters we considered scale linearly with the number of parameters, with feedforward dimension being the least impactful per unit. Increasing the number of decoder layers results in a slightly steeper rate of increase in parameters than adding more encoder layers.

In this experimental setup, we allow the model size to grow freely. We observe that for all datasets increasing embedding size to 2048 or 1024 leads to the best ChrF scores, but also to disproportionally big models, reaching 75M or 251M parameters. For all four datasets, it is sufficient to scale back both embedding size and feedforward dimension to 256 to obtain the most efficient configuration. These optimized models have between 91.6% and 97.5% parameters less than the best architectures according to ChrF, while losing 1%-13.7% of the translation performance. We argue this is a favourable trade-off, especially in a low-resource setting where it may be needed to train several models in sequence for techniques such as back-translation.

As a matter of comparison, we prompted two language models, mt5-small (300M parameters) (Xue et al., 2021) and Mistral Small (24B parameters)[5], for translation in a zero-shot scenario. mt5-small fails at translating between all pairs, even the high-resource English-Italian[6], reaching only 3.1 ChrF for *deu-dsb*. Consequently, using the model in this way is very inefficient, as reflected by hugely inflated PIES scores. The much bigger Mistral Small fares much better for English-Italian, achieving 59 ChrF, thus making it the best model overall in our experiment for this pair. Its efficiency is, however, debatable, as indicated by PIES scores in the hundreds and thousands. With respect of the low-resource pairs, Mistral Small fails for Akkadian and Manipuri, and performs poorly for Lower Sorbian.

We then trained a new mt5-small model for each pair by finetuning the original mt5-small on each dataset for 5 epochs. As expected, this increased ChrF and decreased PIES across all pairs. The best performance is for English-Italian, which reaches a ChrF of 44 points, comparable to other models trained from scratch. PIES is still quite high, however, at 6.8. Finetuning significantly helps mt5-small to make sense of unseen low-resource languages, especially with Lower Sorbian and Manipuri, but both performance and efficiency are well below models trained from scratch. This suggests that when only a small amount of data is available, training from scratch may still be the best choice, especially if the under-resourced languages in question are not in the training data of the language model to finetune or adapt.

Table 7 gives ChrF and PIES for two language models prompted for translation across the language pairs in our experiment.[7]

4.2 Experiment 2: Parameters Budget

In Experiment 2, we limit the number of parameters in three ranges, corresponding to the sizes of Transformer *small*, *base*, and *large* (Table 4). The higher number of combinations per dataset (131) allows for observations regarding some *average* trends in our results. To extrapolate optimal ranges for the hyperparameters and their interactions, we proceed in three steps: 1. we filter out all combinations with a ChrF<35; 2. among these, we keep only those with PIES<1; 3. we select ranges where the remaining combinations are optimal across the majority of the datasets. To generalize these ranges,

[5]https://mistral.ai/news/mistral-small-3-1

[6]English and Italian are known to mt5. English has the biggest share of the training data (5.67%), and Italian has the sixth biggest (2.63%) of the total amount.

[7]Refer to Appendix B for information about the prompting and finetuning of these models.

	eng-akk	deu-dsb	eng_wiki-ita_wiki	eng-mni
Best Model (ChrF)	**4_4_2048_1024_2**	**4_4_2048_1024_2**	**4_4_2048_1024_2**	**4_4_1024_1024_2**
ChrF	41.792	48.881	45.612	48.505
PIES	6.017	5.145	5.513	1.555
Num. Parameters	251M	251M	251M	75M
Best Model (PIES)	**4_4_256_256_2**	**4_4_256_256_2**	**4_4_256_256_2**	**4_4_256_256_2**
ChrF	39.681	44.527	45.156	41.844
PIES	0.158	0.141	0.139	0.150
Num. Parameters	6.3M	6.3M	6.3M	6.3M
Δ ChrF ↑	**-2.111**	**-4.354**	**-0.456**	**-6.661**
% of best	**-5.052%**	**-8.907%**	**-1.000%**	**-13.732%**
Δ PIES ↓	**-5.859**	**-5.004**	**-5.374**	**-1.405**
% of best	**-97.374%**	**-89.308%**	**-97.479%**	**-90.354%**
Δ Params ↓	**-245M**	**-245M**	**-245M**	**-69M**
% of best	**-97.507%**	**-97.490%**	**-97.507%**	**-91.600%**

Table 5: **Best models from Experiment 1 according to ChrF and PIES.** Below the model name (in the form *enc_dec_embs_ffw_heads*), we report ChrF, PIES, and size of the model. In the bottom part of the table, we report the differences in scores ($\Delta ChrF \uparrow$, $\Delta PIES \downarrow$) and size ($\Delta Params \downarrow$) between the best model for translation quality (highest ChrF) and the most efficient one (lowest PIES), both in absolute terms and as a percentage (% of best).

	eng-akk	deu-dsb	eng_wiki-ita_wiki	eng-mni
Best Model (ChrF)	**6_8_1024_2048_2**	**12_2_1024_4096_2**	**12_2_1024_4096_2**	**2_16_1024_256_2**
ChrF	43.393	51.569	47.890	49.882
PIES	3.673	3.741	4.029	3.217
Num. Parameters	176M	193M	193M	160M
Size range	large	large	large	large
Best Model (PIES)	**4_6_256_512_2**	**6_2_256_1024_2**	**6_2_256_1024_2**	**4_8_256_256_2**
ChrF	38.811	44.001	45.347	44.483
PIES	0.229	0.202	0.196	0.200
Num. Parameters	8.9M	8.9M	8.9M	8.9M
Size range	small	small	small	small
Δ ChrF ↑	**-4.582**	**-7.568**	**-2.543**	**-5.339**
% of best	**-10.559%**	**-14.790%**	**-5.310%**	**-10.823%**
Δ PIES ↓	**-3.444**	**-3.539**	**-3.833**	**-3.017**
% of best	**-93.375%**	**-94.600%**	**-95.135%**	**-93.783%**
Δ Params ↓	**-167M**	**-184M**	**-184M**	**-151M**
% of best	**-94.943%**	**-95.389%**	**-95.389%**	**-94.438%**

Table 6: **Best models from Experiment 2 according to ChrF and PIES.** Below the model name (in the form *enc_dec_embs_ffw_heads*), we report ChrF, PIES, and size of the model. In the bottom part of the table, we report the differences in scores ($\Delta ChrF \uparrow$, $\Delta PIES \downarrow$) and size ($\Delta Params \downarrow$) between the best model for translation quality (highest ChrF) and the most efficient one (lowest PIES), both in absolute terms and as a percentage (% of best).

we filter all combinations according to the values we found, starting with the most impactful ones. Table 8 summarizes these findings, and Table 9 reports the possible optimal configurations we found.

At a glance, it can be observed that pruning models that are either too deep or too imbalanced in terms of encoder and decoder layers leads, on average, to better ChrF and PIES, and greatly reduces the size of the models. Limiting embedding size may reduce quality, however this can be circumvented by selecting balanced architectures according to other criteria. We can also observe that setting the feedforward dimension in the suggested range, when taken in relation to the number of encoder and decoder layers, slightly increases both ChrF and PIES. Further experiments may ad-

Model	Metric	eng-akk	deu-dsb	eng_wiki-ita_wiki	eng-mni
mt5-small	**ChrF**	0.070	3.101	1.584	0.004
(300M)	**PIES**	*4271.370*	*96.729*	*189.406*	*82101.976*
mt5-small-finetuned	**ChrF**	7.136	32.681	44.033	25.687
(300M)	**PIES**	*42.043*	*9.180*	*6.813*	*11.679*
mistral-small	**ChrF**	0.358	18.332	59.086	4.966
(24B)	**PIES**	*67039.379*	*1309.188*	*406.190*	*4832.926*

Table 7: **ChrF and PIES for zero-shot mt5-small, finetuned mt5-small, and zero-shot Mistral-Small.**

Filter	Low	High	Remaining Configurations	Avg ChrF	Avg PIES	Avg Size (in M)
Initial	-	-	524	33.93	5.06	102M
layer_sum	6	18	244	42.137	2.087	84.2
embs	256	512	148	40.659	0.898	31.8
enc_layers	2	12	136	41.491	0.871	31.7
dec_layers	2	12	132	41.653	0.806	31.3
layer_diff	-6	8	116	41.999	0.771	30.8
ffw	256	2048	92	42.163	0.675	27.3
enc/dec	0.25	3	92	42.163	0.675	27.3
embs/ffw	0.125	2	92	42.163	0.675	27.3
embs/enc	21.333	256	92	42.163	0.675	27.3
embs/dec	32	128	92	42.163	0.675	27.3
embs/num_layers	16	51.2	92	42.163	0.675	27.3
ffw/dec	21.333	1024	92	42.163	0.675	27.3
ffw/enc	32	1024	88	42.3	0.695	28.1

Table 8: **Optimal ranges of the hyperparameters and their interactions.** The first two columns give minimum and maximum values for each one. The other report the remaining configurations after filtering the possible combinations, and their average ChrF, PIES, and size in millions of parameters.

dress the impact of this particular relation. In summary, optimal architectures should have a limited number of layers (6 to 18), which must be not too unbalanced on either encoder or decoder side (-6 to 8 encoder - decoder difference; 0.125 to 3 encoder/decoder ratio). Embedding size should be kept on the smaller side (256 to 512), while feed-forward dimension can be bigger (up to 2048).

If we apply these guidelines to the full results, without prior filtering for ChrF and PIES, we are left with the 22 combinations in Table 9, out of the initial 131 per dataset. Apart from four, all have an average ChrF > 40, and six have an average PIES > 1. All of these are of *base* size. There is a noticeable gap between PIES for this size bracket and *small*, which features the best efficiency scores.

Table 6 reports the best models and scores in Experiment 2. All the best model according to ChrF are in the *large* range, whereas the most efficient ones according to PIES are in the *small* bracket. For two datasets, *deu-dsb* and *eng_wiki-ita_wiki*, the best ChrF model is the same (12_2_1024_4096_2). The best ChrF model for *eng-mni* is quite peculiar:

Model	Avg ChrF	Avg PIES	Size (in M)
6_2_256_1024_2	42.757	0.209	8.9
4_6_256_512_2	42.062	0.212	8.9
4_4_256_1024_2	42.836	0.221	9.4
8_4_256_512_2	41.845	0.226	9.4
6_8_256_256_2	42.470	0.229	9.7
6_6_256_512_2	42.859	0.233	10.0
4_8_256_256_2	39.176	0.242	8.9
2_8_256_512_2	40.237	0.247	9.4
2_6_256_1024_2	40.762	0.254	9.9
8_6_256_256_2	36.428	0.316	9.2
8_6_512_1024_2	45.171	0.885	39.8
6_4_512_2048_2	44.738	0.894	39.8
4_6_512_2048_2	45.423	0.925	41.9
6_8_512_1024_2	44.926	0.937	41.9
12_4_512_1024_2	43.269	0.978	41.9
2_8_512_2048_2	44.249	0.998	44.0
8_4_512_2048_2	45.516	1.015	46.1
6_6_512_2048_2	44.920	1.077	48.2
12_6_512_1024_2	43.353	1.122	48.3
6_12_512_512_2	40.087	1.258	45.1
6_12_512_256_2	38.743	1.316	40.4
8_8_512_1024_2	38.762	1.484	46.2

Table 9: **Optimal model configurations** and their size, with ChrF and PIES averaged over all four datasets.

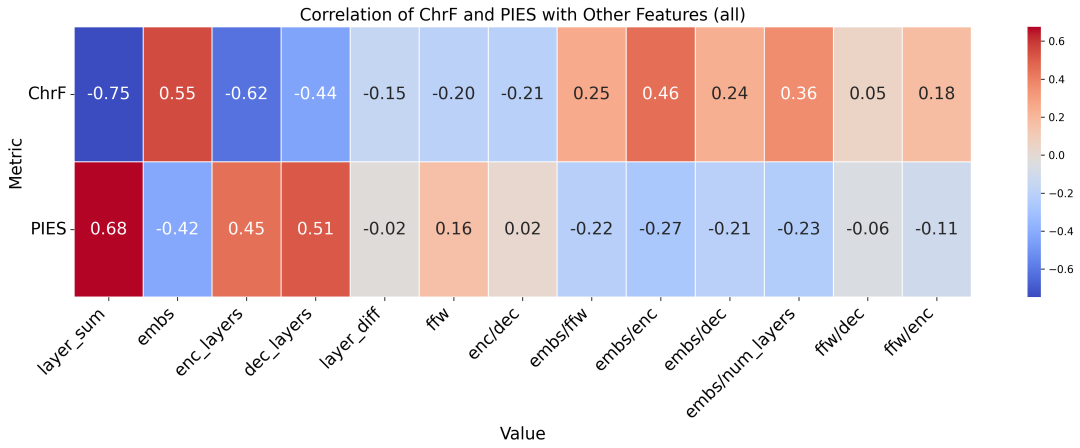


Figure 1: **Correlation matrix of ChrF and PIES** with other hyperparameters and their interactions. A lower PIES is better.

it has just 2 encoder layers, 16 decoder layers, an embedding size of 1024, and a narrow feedforward of just 256. Again we see decrements in ChrF between 5.3% and 14.8%, against a 95% reduction in number of parameters. While bigger models may in principle achieve a slightly higher ChrF, this comes at the cost of efficiency. We argue that in a low-resource scenario, when both data and hardware are scarce, the increased computational cost needed to find and train the optimal model in this size range is not well spent. Smaller models can achieve a comparable, or almost comparable translation performance, at just a fraction of the cost. This is also true for exploratory runs, or intermediate stages of development, such as systems for backtranslation.

From Figure 1, reporting the correlation between ChrF and PIES and all the features in our study, we can point out some interesting observations. The number of layers is the most impactful hyperparameter for both ChrF and PIES. Deeper architectures lose both in terms of quality and efficiency. It follows that the number of encoder and decoder layers impacts the metrics, with changes in the encoder slightly more important for ChrF, and changes in the decoder a bit more impactful on PIES. Embedding dimension is relevant for both metrics. Feedforward appears to have a lesser impact. According to these findings, **balancing embeddings dimension, the number of layers** in the encoder and the decoder, **and their interactions is key** to an efficient model with good translation quality.

5 Conclusions

In this paper, we explored scaling and optimizing the Transformer architecture for low-resource machine translation by experimenting with several hundred configurations over four language pairs.

We confirm previous findings that the Transformer, and low-resource NMT in general, is highly sensitive to hyperparameters in low-resource conditions, and that standard settings are not optimal. We observe some trends and interactions between the number of encoder and decoder layers, embedding size, feedforward dimension, and the quality of the translation.

We propose PIES as a novel metric to measure the efficiency of changing a model's architecture, and use it to show that increasing model size is not always the optimal choice, since smaller and balanced models can reach a comparable performance for a fraction of the computational cost. We also outline some empirical findings and guidelines regarding the optimal hyperparameter ranges that result in more efficient low-resource machine translation models.

Limitations

The main limitations of our experiments are the following. First, the dataset selection, while trying to be diverse both in terms of typology (Germanic, Slavic, Romance, Tibeto-Burman, Semitic) and writing system (Latin, Bengali, Cuneiform), is only a tiny fraction of the world's 7000+ languages. If we include, also historical ones, such is the case with Akkadian, the number grows even more. We acknowledge that this fact may hinder

generalization, and to avoid even more grid search and computation, we attempted to gain as much information as possible from these datasets. We leave to future work to test our intuitions on a wider range of languages.

Second, we could not perform a systematic qualitative analysis on the outputs of the models, and had to rely on automated metrics to score the translations. This comes with another set of problems altogether, that is out of the scope of this paper to discuss. This is also relevant for PIES, which in its present iteration is closely correlated with the translation metric. In the future, we plan to extend it to account for multiple metrics, and to consider also train and inference times, and environmental concerns. For now, it is only as good as the translation metric chosen to compute it.

Lastly, we are aware that testing all possible combinations, across all hyperparameters, is a monumental task that evades the scope of just one paper. We focused on four specific architecture hyperparameters and their interactions. Other possible optimal configurations, that may need other changes in training hyperparameters (e. g. learning rate, dropout, etc.) to work best are left to future work. The same can be said for all LLMs, for which architecture cannot be modified as freely, with one led to employ different approaches such as fine-tuning and prompting techniques. These fall outside the scope of this paper and are left to future work.

Ethical Considerations

We did not collect any new data for these experiments, as we used publicly available dataset or parts thereof. The systems we trained are not intended to be deployed or used in any actual translation scenario, in such a case, they will incur in biases, errors, and issues common to this kind of NLP models, and as such they should be used with care. We are also aware of the environmental cost of training language models and tried our best to avoid grid search all the while getting a meaningful picture of the topic at hand.

Following (Lacoste et al., 2019), we estimate that our experiments lasted 4200 GPU hours on a private infrastructure with a carbon efficiency of 0.59 kgCO_2eq/kWh for a total emissions of 708 kgCO_2eq.

Acknowledgments

We would like to thank the reviewers for their useful comments. This work has been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic within the LINDAT-CLARIAH-CZ project LM2023062.

References

Ali Araabi and Christof Monz. 2020. Optimizing transformer for low-resource neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3429–3435, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Seth Aycock, David Stap, Di Wu, Christof Monz, and Khalil Sima'an. 2025. Can LLMs really learn to translate a low-resource language from one grammar book? In *The Thirteenth International Conference on Learning Representations*.

Alexandre Berard, Dain Lee, Stephane Clinchant, Kweonwoo Jung, and Vassilina Nikoulina. 2021. Efficient inference for multilingual neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8563–8583, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Elan van Biljon, Arnu Pretorius, and Julia Kreutzer. 2020. On optimal transformer depth for low-resource language translation.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. 2022. Scaling laws for neural machine translation. In *International Conference on Learning Representations*.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. 2021. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán,

and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Ping Guo, Yubing Ren, Yue Hu, Yunpeng Li, Jiarui Zhang, Xingsheng Zhang, and Heyan Huang. 2024. Teaching large language models to translate on low-resource languages with textbook prompting. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15685–15697, Torino, Italia. ELRA and ICCL.

Gai Gutherz, Shai Gordin, Luis Sáenz, Omer Levy, and Jonathan Berant. 2023. Translating Akkadian to English with neural machine translation. *PNAS Nexus*, 2(5):pgad096.

Barry Haddow and Faheem Kirefu. 2020. Pmindia – a collection of parallel corpora of languages of india.

Yi-Te Hsu, Sarthak Garg, Yi-Hsiu Liao, and Ilya Chatsviorkin. 2020. Efficient inference for neural machine translation. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 48–53, Online. Association for Computational Linguistics.

Rudali Huidrom, Yves Lepage, and Khogendra Khomdram. 2021. EM corpus: a comparable corpus for a less-resourced language pair Manipuri-English. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, pages 60–67, Online (Virtual Mode). INCOMA Ltd.

Raviraj Joshi, Kanishk Singla, Anusha Kamath, Raunak Kalani, Rakesh Paul, Utkarsh Vaidya, Sanjay Singh Chauhan, Niranjan Wartikar, and Eileen Long. 2025. Adapting multilingual LLMs to low-resource languages using continued pre-training and synthetic corpus: A case study for Hindi LLMs. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 50–57, Abu Dhabi. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In *International Conference on Learning Representations*.

Omkar Khade, Shruti Jagdale, Abhishek Phaltankar, Gauri Takalikar, and Raviraj Joshi. 2025. Challenges in adapting multilingual LLMs to low-resource languages using LoRA PEFT tuning. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025)*, pages 217–222, Abu Dhabi, UAE. International Committee on Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Lenin Laitonjam and Sanasam Ranbir Singh. 2021. Manipuri-English machine translation using comparable corpus. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*, pages 78–88, Virtual. Association for Machine Translation in the Americas.

Hongyuan Lu, Haoran Yang, Haoyang Huang, Dongdong Zhang, Wai Lam, and Furu Wei. 2024. Chain-of-dictionary prompting elicits translation in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 958–976, Miami, Florida, USA. Association for Computational Linguistics.

Marion Weller-di Marco and Alexander Fraser. 2022. Findings of the WMT 2022 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 801–805, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Raphaël Merx, Aso Mahmudi, Katrina Langford, Leo Alberto de Araujo, and Ekaterina Vylomova. 2024. Low-resource machine translation through retrieval-augmented LLM prompting: A study on the Mambai language. In *Proceedings of the 2nd Workshop on Resources and Technologies for Indigenous, Endangered and Lesser-resourced Languages in Eurasia (EURALI) @ LREC-COLING 2024*, pages 1–11, Torino, Italia. ELRA and ICCL.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Santanu Pal, Partha Pakray, Sahinur Rahman Laskar, Lenin Laitonjam, Vanlalmuansangi Khenglawt,

Sunita Warjri, Pankaj Kundan Dadure, and Sandeep Kumar Dash. 2023. Findings of the WMT 2023 shared task on low-resource Indic language translation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 682–694, Singapore. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

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.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2023. Neural machine translation for low-resource languages: A survey. *ACM Comput. Surv.*, 55(11).

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.

Nathaniel R. Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. Chatgpt mt: Competitive for high- (but not low-) resource languages.

Ananya Sai B, Tanay Dixit, Vignesh Nagarajan, Anoop Kunchukuttan, Pratyush Kumar, Mitesh M. Khapra, and Raj Dabre. 2023. IndicMT eval: A dataset to meta-evaluate machine translation metrics for Indian languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14210–14228, Toronto, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Edoardo Signoroni and Pavel Rychlý. 2024. Better low-resource machine translation with smaller vocabularies. In *Text, Speech, and Dialogue*, pages 184–195, Cham. Springer Nature Switzerland.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2022. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations*.

Jörg Tiedemann. 2009. *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*, volume V, pages 237–248.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Jiayi Wang, David Ifeoluwa Adelani, Sweta Agrawal, Marek Masiak, Ricardo Rei, Eleftheria Briakou, Marine Carpuat, Xuanli He, Sofia Bourhim, Andiswa Bukula, Muhidin Mohamed, Temitayo Olatoye, Tosin Adewumi, Hamam Mokayede, Christine Mwase, Wangui Kimotho, Foutse Yuehgoh, Anuoluwapo Aremu, Jessica Ojo, Shamsuddeen Hassan Muhammad, Salomey Osei, Abdul-Hakeem Omotayo, Chiamaka Chukwuneke, Perez Ogayo, Oumaima Hourrane, Salma El Anigri, Lolwethu Ndolela, Thabiso Mangwana, Shafie Abdi Mohamed a nd Ayinde Hassan, Oluwabusayo Olufunke Awoyomi, Lama Alkhaled, Sana Al-Azzawi, Naome A. Etori, Millicent Ochieng, Clemencia Siro, Samuel Njoroge, Eric Muchiri, Wangari Kimotho, Lyse Naomi Wamba Momo, Daud Abolade, Simbiat Ajao, Iyanuoluwa Shode, Ricky Macharm, Ruqayya Nasir Iro, Saheed S. Abdullahi, Stephen E. Moore, Bernard Opoku, Zainab Akinjobi, Abeeb Afolabi, Nnaemeka Obiefuna, Onyekachi Raphael Ogbu, Sam Brian, Verrah Akinyi Otiende, Chinedu Emmanuel Mbonu, Sakayo Toadoum Sari, Yao Lu, and Pontus Stenetorp. 2024. Afrimte and africomet: Enhancing comet to embrace under-resourced african languages.

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 Languages

Lower Sorbian (*"Dolnoserbšćina"*) is a West Slavic language predominantly spoken in eastern Germany by approximately 7,000 native speakers. Most of these speakers are from older generations, making the language critically endangered. Written in Latin script with additional diacritics, Lower Sorbian features six grammatical cases and a dual number system for nouns, pronouns, adjectives, and verbs. It does not employ articles. The dataset for our experiments was compiled by the Witaj Sprachzentrum [8] (Witaj Language Centre) (Weller-di Marco and Fraser, 2022).

Manipuri (*"Meiteilon"*) is a Tibeto-Burman language recognized as one of the official languages in the Indian state of Manipur and at the national level. It is spoken by approximately 1.8 million native speakers, primarily the Meitei people, both in Manipur and neighboring regions. UNESCO classifies Manipuri as "vulnerable." The language exhibits extensive suffixation with limited prefixation and follows an SVO word order. Other linguistic characteristics include agglutinative verb morphology, tone, a lack of grammatical person, number, and gender distinctions, and a focus on aspect rather than tense (Pal et al., 2023). Manipuri is written using several scripts, including the Meitei and Bengali scripts, with the latter being used for all the Manipuri data in our experiments. The Latin script is also employed. The dataset is a modified version (Pal et al., 2023) based on previous work by Haddow and Kirefu (2020), Laitonjam and Ranbir Singh (2021), and Huidrom et al. (2021). Each segment of the data set contains mainly news and other informational texts.

Akkadian, an extinct East Semitic language, was spoken in ancient Mesopotamia from the third millennium BCE until the 1st century CE. It utilized the cuneiform script, a logophonetic writing system in which symbols could serve as logograms, determinatives, or phonograms/syllabograms, each with a distinct interpretation. Akkadian is a fusional language with grammatical case and employs a root-based consonantal system. The dataset used in our study is derived from portions of the ORACC corpus [9] and mainly comprises Neo-Assyrian royal inscriptions and administrative correspondence. The stylistic variation between genres poses challenges for NLP (Gutherz et al., 2023). Additionally, because of the medium of preservation (clay tablets), the data is often incomplete, with truncated sentences.

[8] https://www.witaj-sprachzentrum.de/

[9] https://oracc.museum.upenn.edu/index.html

B Large Language Models' Prompting and Finetuning

mt5-small was prompted and finetuned using the HuggingFace Transfomers library, while Mistral Small was prompted with Ollama. For mt5-small, we used the prompt 'Translate SRC to TGT: SRC_EXAMPLE'. For fine-tuning, we restructured the data in a similar way, by giving the prompt, the source line, as input, and the target line as label. For Mistral Small, we opted for a somewhat more complex 'Translate the following text from SRC to TGT. Write only the translation. SRC: SRC_EXAMPLE TGT: '

C Tables and Charts

Below we report the results of our experiments. Tables 10 and 11 summarize the ChrF and PIES scores for each pair and size bracket of Experiment 2. Figures and plot the trend of ChrF when modifying each hyperparameter for each pair. Figures 4 and 5 show the counts of optimal configurations (ChrF>35 and PIES<1) for each hyperparameter across all datasets. Figure 6 report the average ChrF and PIES for the configurations in the optimal range for each language pair and size bracket. Finally, Tables 12 to 18 contain the BLEU, ChrF, and COMET scores for all combinations in Experiment 1 and 2.

tgt	size_tag	min	max	median	mean
akk	base	4.901	42.568	19.969	23.426
akk	large	6.088	43.394	30.677	26.093
akk	small	14.408	39.211	35.026	32.287
dsb	base	3.017	48.324	41.664	32.098
dsb	large	2.898	51.569	47.095	36.073
dsb	small	37.982	44.329	43.008	42.741
ita_wiki	base	2.928	46.282	43.531	33.336
ita_wiki	large	1.599	47.890	44.698	34.910
ita_wiki	small	42.504	45.347	44.919	44.708
mni	base	7.917	48.299	43.241	38.805
mni	large	7.733	49.883	44.606	40.643
mni	small	35.996	45.960	43.443	42.614
all	base	2.928	48.324	38.619	31.915
all	large	1.599	51.569	41.837	34.430
all	small	14.408	45.960	42.987	40.588

Table 10: Minimum, Maximum, Average, and Median ChrF values by language and size bracket in Experiment 2

tgt	size_tag	min	max	median	mean
akk	base	0.935	9.100	2.196	2.907
akk	large	3.673	29.323	5.951	10.493
akk	small	0.229	0.636	0.284	0.315
dsb	base	0.848	14.655	1.076	2.614
dsb	large	3.131	59.464	3.950	9.467
dsb	small	0.202	0.255	0.219	0.221
ita_wiki	base	0.861	16.206	1.067	2.378
ita_wiki	large	3.346	101.225	4.204	9.545
ita_wiki	small	0.196	0.228	0.210	0.211
mni	base	0.836	5.048	1.031	1.395
mni	large	3.218	22.284	3.953	5.422
mni	small	0.200	0.269	0.217	0.223
all	base	0.836	16.206	1.165	2.322
all	large	3.131	101.225	4.397	8.732
all	small	0.196	0.636	0.222	0.243

Table 11: Minimum, Maximum, Average, and Median PIES values by language and size bracket in Experiment 2

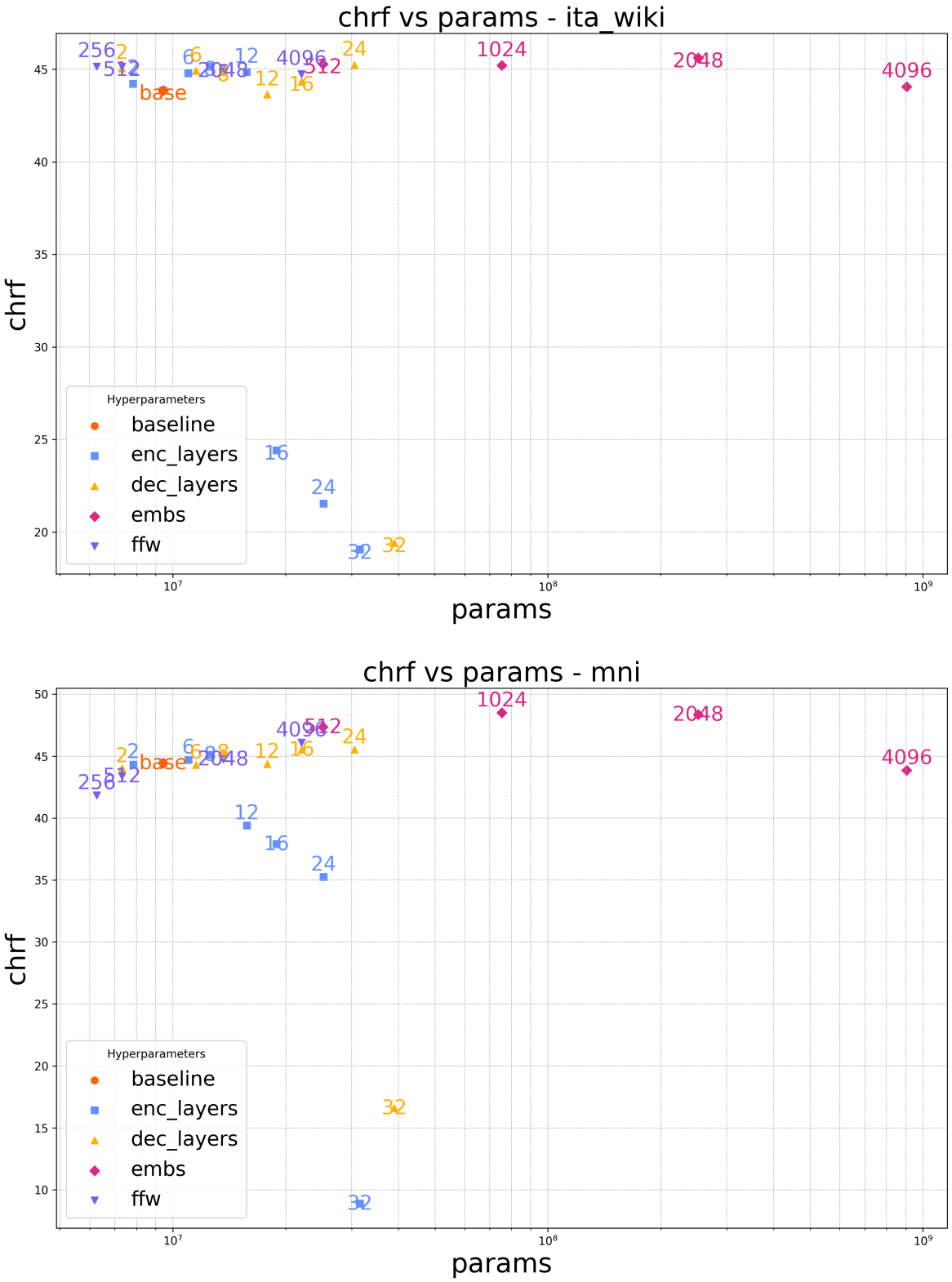


Figure 2: Charts plotting ChrF for each hyperparameter against the increase in number of parameters.

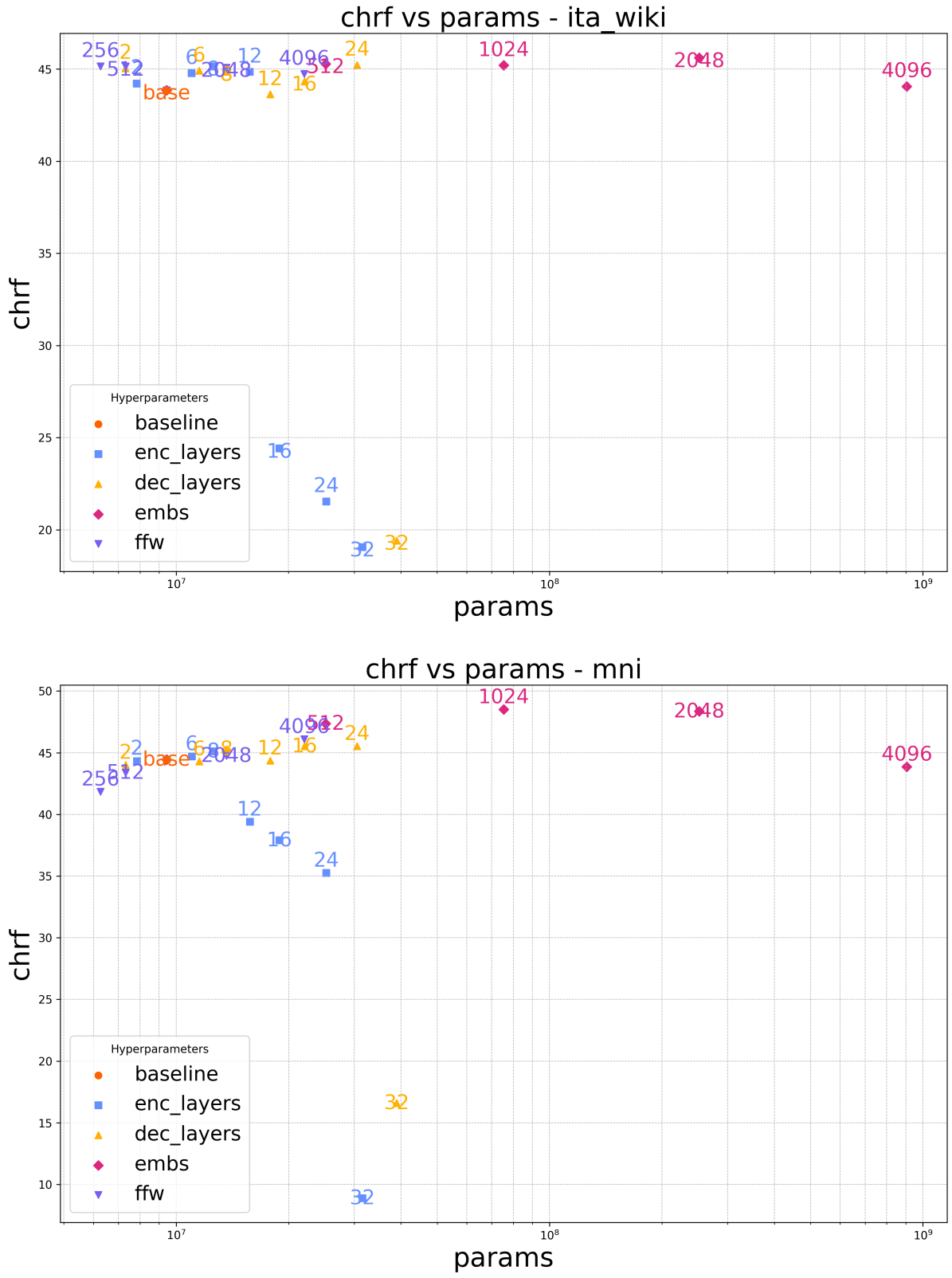


Figure 3: Charts plotting ChrF for each hyperparameter against the increase in number of parameters. (Continued)

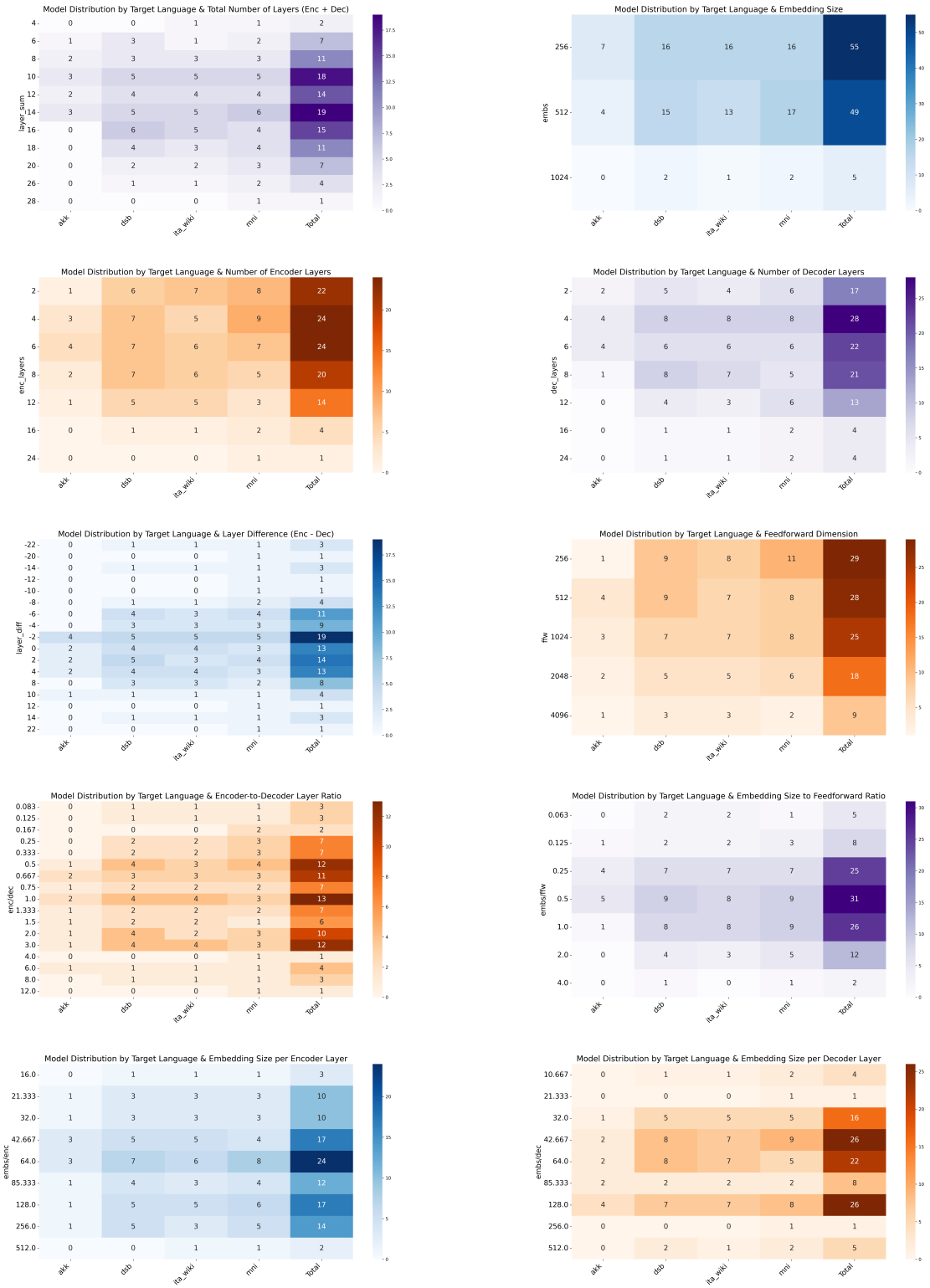


Figure 4: Heatmaps of the counts of configurations that achieve ChrF>35 with a PIES<1.

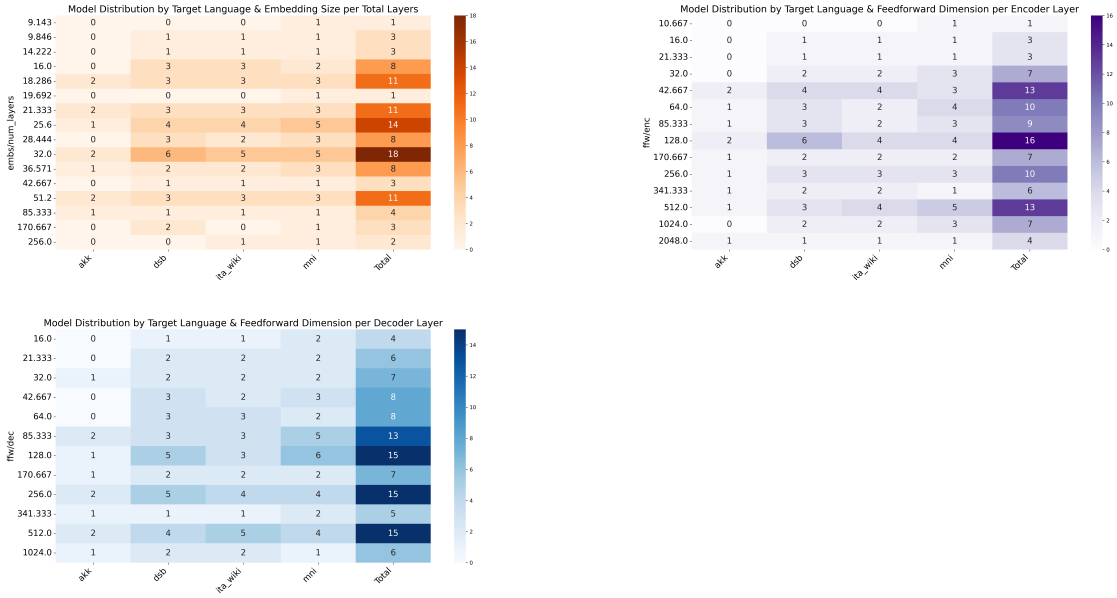


Figure 5: Heatmaps of the counts of configurations that achieve ChrF>35 with a PIES<1. (Continued)

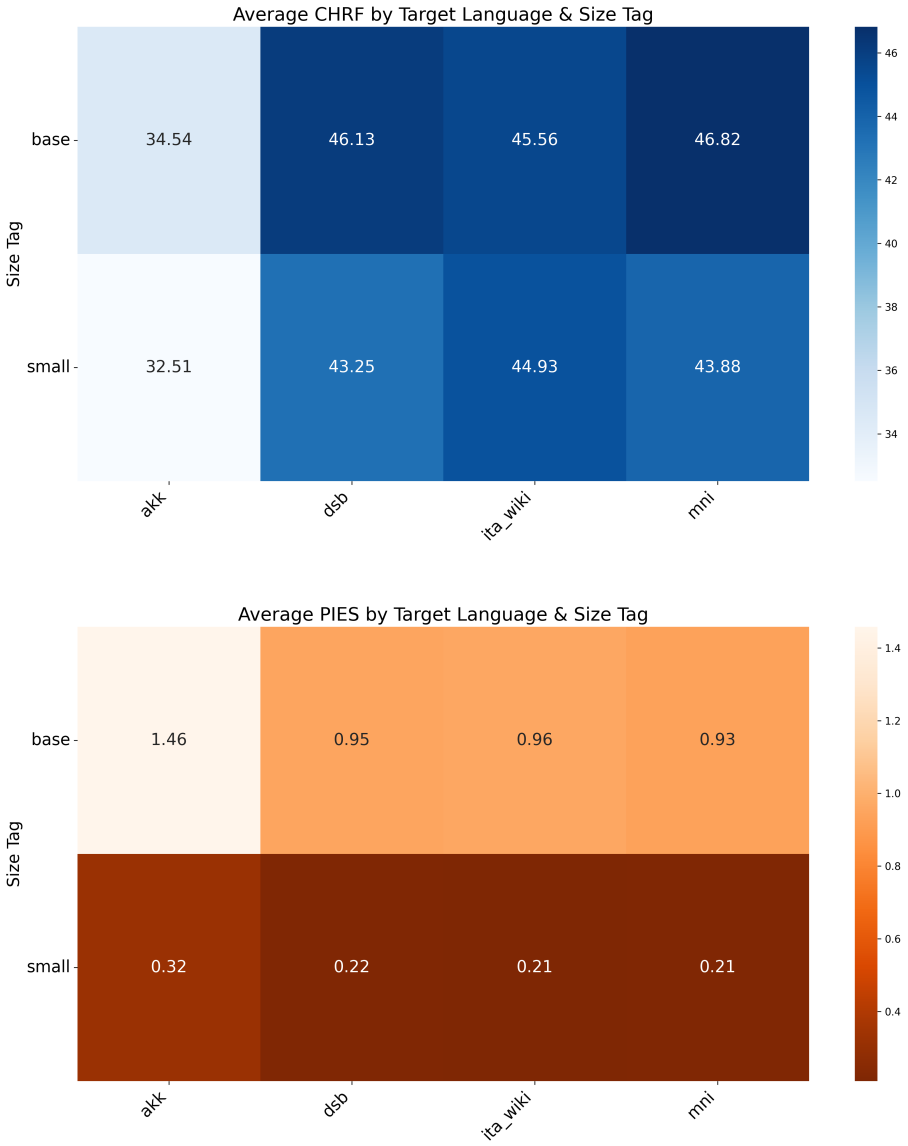


Figure 6: Average ChrF and PIES for the configurations in the optimal ranges, per size bracket. A lower PIES is better.

Table 12: Results for Experiment 1

src	tgt	model	bleu	chrf	comet	src	tgt	model	bleu	chrf	comet
eng	akk	2_4_256_1024_2	29.91	38.95	0.93	eng_wiki	ita_wiki	4_4_256_1024_2	12.59	43.86	0.59
eng	akk	4_2_256_1024_2	29.25	37.32	0.93	eng_wiki	ita_wiki	4_4_256_2048_2	13.38	45.08	0.61
eng	akk	4_4_256_256_2	30.23	39.68	0.93	eng_wiki	ita_wiki	4_4_256_4096_2	13.59	44.74	0.6
eng	akk	4_4_256_512_2	30.03	38.67	0.93	eng_wiki	ita_wiki	4_4_512_1024_2	13.53	45.28	0.6
eng	akk	4_4_256_1024_2	30.05	38.83	0.93	eng_wiki	ita_wiki	4_4_1024_1024_2	13.48	45.2	0.61
eng	akk	4_4_256_2048_2	29.82	38.43	0.93	eng_wiki	ita_wiki	4_4_2048_1024_2	14.04	45.61	0.62
eng	akk	4_4_256_4096_2	30.61	40.86	0.93	eng_wiki	ita_wiki	4_4_4096_1024_2	12.93	44.06	0.59
eng	akk	4_4_512_1024_2	30.33	40.49	0.93	eng_wiki	ita_wiki	4_6_256_1024_2	13.31	44.91	0.6
eng	akk	4_4_1024_1024_2	29.99	41.18	0.93	eng_wiki	ita_wiki	4_8_256_1024_2	13.57	44.86	0.6
eng	akk	4_4_2048_1024_2	30.17	41.79	0.93	eng_wiki	ita_wiki	4_12_256_1024_2	12.38	43.64	0.57
eng	akk	4_4_4096_1024_2	28.41	38.81	0.93	eng_wiki	ita_wiki	4_16_256_1024_2	13.07	44.33	0.58
eng	akk	4_6_256_1024_2	30.27	39.59	0.93	eng_wiki	ita_wiki	4_24_256_1024_2	13.57	45.23	0.61
eng	akk	4_8_256_1024_2	30.07	39.68	0.93	eng_wiki	ita_wiki	4_32_256_1024_2	2.18	19.42	0.26
eng	akk	4_12_256_1024_2	28.41	33.85	0.93	eng_wiki	ita_wiki	6_4_256_1024_2	13.28	44.79	0.61
eng	akk	4_16_256_1024_2	23.26	16.96	0.9	eng_wiki	ita_wiki	8_4_256_1024_2	13.74	45.15	0.61
eng	akk	4_24_256_1024_2	15.94	7.76	0.77	eng_wiki	ita_wiki	12_4_256_1024_2	13.55	44.86	0.61
eng	akk	4_32_256_1024_2	15.61	7.54	0.8	eng_wiki	ita_wiki	16_4_256_1024_2	3.68	24.42	0.36
eng	akk	6_4_256_1024_2	29.45	38.41	0.93	eng_wiki	ita_wiki	24_4_256_1024_2	3.26	21.55	0.35
eng	akk	8_4_256_1024_2	23.95	23.18	0.92	eng_wiki	ita_wiki	32_4_256_1024_2	2.81	19.07	0.34
eng	akk	12_4_256_1024_2	27.73	33.21	0.93	eng	mni	2_4_256_1024_2	16.15	44.33	0.69
eng	akk	16_4_256_1024_2	27.81	32.55	0.93	eng	mni	4_2_256_1024_2	16.93	44.01	0.69
eng	akk	24_4_256_1024_2	27.53	30.62	0.93	eng	mni	4_4_256_256_2	13.93	41.84	0.68
eng	akk	32_4_256_1024_2	21.49	16.32	0.89	eng	mni	4_4_256_512_2	16.06	43.39	0.69
deu	dsb	2_4_256_1024_2	28.06	43.7	0.63	eng	mni	4_4_256_1024_2	18.03	44.45	0.7
deu	dsb	4_2_256_1024_2	28.45	43.92	0.62	eng	mni	4_4_256_2048_2	18.2	44.75	0.7
deu	dsb	4_4_256_256_2	28.41	44.53	0.63	eng	mni	4_4_256_4096_2	20.45	46.11	0.7
deu	dsb	4_4_256_512_2	27.93	43.58	0.63	eng	mni	4_4_512_1024_2	20.2	47.35	0.71
deu	dsb	4_4_256_1024_2	28.09	43.01	0.62	eng	mni	4_4_1024_1024_2	21.88	48.5	0.71
deu	dsb	4_4_256_2048_2	29.28	45.19	0.64	eng	mni	4_4_2048_1024_2	21.45	48.36	0.7
deu	dsb	4_4_256_4096_2	28.53	43.95	0.63	eng	mni	4_4_4096_1024_2	18.7	43.87	0.68
deu	dsb	4_4_512_1024_2	29.34	46.72	0.64	eng	mni	4_6_256_1024_2	17.09	44.29	0.69
deu	dsb	4_4_1024_1024_2	29.96	48.29	0.65	eng	mni	4_8_256_1024_2	19.53	45.3	0.7
deu	dsb	4_4_2048_1024_2	30.48	48.88	0.65	eng	mni	4_12_256_1024_2	18.1	44.36	0.69
deu	dsb	4_4_4096_1024_2	29.8	48.28	0.64	eng	mni	4_16_256_1024_2	19.56	45.53	0.7
deu	dsb	4_6_256_1024_2	27.21	42.85	0.62	eng	mni	4_24_256_1024_2	18.84	45.54	0.69
deu	dsb	4_8_256_1024_2	27.65	42.74	0.62	eng	mni	4_32_256_1024_2	4.0	16.61	0.42
deu	dsb	4_12_256_1024_2	28.25	44.73	0.63	eng	mni	6_4_256_1024_2	18.39	44.71	0.7
deu	dsb	4_16_256_1024_2	27.34	44.02	0.63	eng	mni	8_4_256_1024_2	18.74	45.11	0.7
deu	dsb	4_24_256_1024_2	27.08	44.6	0.63	eng	mni	12_4_256_1024_2	15.59	39.42	0.68
deu	dsb	4_32_256_1024_2	11.53	25.88	0.51	eng	mni	16_4_256_1024_2	14.96	37.9	0.68
deu	dsb	6_4_256_1024_2	27.77	43.28	0.62	eng	mni	24_4_256_1024_2	13.63	35.27	0.66
deu	dsb	8_4_256_1024_2	27.82	43.06	0.62	eng	mni	32_4_256_1024_2	1.56	8.89	0.42
deu	dsb	12_4_256_1024_2	27.14	41.99	0.62						
deu	dsb	16_4_256_1024_2	16.14	22.42	0.52						
deu	dsb	24_4_256_1024_2	15.52	20.33	0.51						
deu	dsb	32_4_256_1024_2	13.81	15.04	0.48						
eng_wiki	ita_wiki	2_4_256_1024_2	12.77	44.22	0.59						
eng_wiki	ita_wiki	4_2_256_1024_2	13.72	45.05	0.62						
eng_wiki	ita_wiki	4_4_256_256_2	13.46	45.16	0.61						
eng_wiki	ita_wiki	4_4_256_512_2	13.65	45.14	0.61						

Table 13: Results for Experiment 2 (Part 1/6)

src	tgt	model	bleu	chrf	comet	src	tgt	model	bleu	chrf	comet
eng	akk	2_2_1024_1024_2	29.32	37.35	0.93	eng	akk	6_16_1024_512_2	29.16	38.74	0.93
eng	akk	2_2_2048_4096_2	29.53	39.33	0.93	eng	akk	6_24_256_2048_2	17.19	8.29	0.78
eng	akk	2_4_512_4096_2	31.08	42.57	0.93	eng	akk	6_24_512_4096_2	22.44	6.93	0.77
eng	akk	2_4_2048_256_2	29.88	40.84	0.93	eng	akk	6_32_256_1024_2	17.55	7.46	0.78
eng	akk	2_6_256_1024_2	26.61	28.13	0.93	eng	akk	8_4_256_512_2	29.67	38.25	0.93
eng	akk	2_8_256_512_2	25.76	26.4	0.92	eng	akk	8_4_512_2048_2	30.68	42.29	0.93
eng	akk	2_8_512_2048_2	30.31	41.23	0.94	eng	akk	8_4_1024_4096_2	30.18	42.55	0.93
eng	akk	2_8_1024_4096_2	29.94	41.77	0.93	eng	akk	8_6_256_256_2	23.57	14.41	0.88
eng	akk	2_12_512_1024_2	24.67	26.49	0.92	eng	akk	8_6_512_1024_2	30.06	40.84	0.93
eng	akk	2_12_1024_2048_2	30.48	42.58	0.93	eng	akk	8_8_256_4096_2	22.16	13.1	0.88
eng	akk	2_16_512_256_2	20.85	4.9	0.75	eng	akk	8_8_512_1024_2	21.25	15.6	0.88
eng	akk	2_16_1024_256_2	30.24	42.42	0.93	eng	akk	8_8_1024_2048_2	22.59	14.15	0.89
eng	akk	2_16_1024_512_2	30.1	41.81	0.93	eng	akk	8_12_512_256_2	20.22	14.54	0.89
eng	akk	2_16_1024_1024_2	30.81	43.07	0.93	eng	akk	8_12_512_512_2	20.67	15.01	0.9
eng	akk	2_24_256_2048_2	15.62	8.01	0.78	eng	akk	8_12_1024_512_2	21.77	14.94	0.88
eng	akk	2_24_512_4096_2	16.75	7.89	0.84	eng	akk	8_12_1024_1024_2	22.37	16.06	0.91
eng	akk	4_2_1024_256_2	30.08	40.15	0.93	eng	akk	8_16_1024_256_2	21.07	15.57	0.91
eng	akk	4_2_1024_512_2	29.27	37.93	0.93	eng	akk	8_32_256_1024_2	17.49	7.52	0.78
eng	akk	4_2_2048_512_2	28.78	37.81	0.93	eng	akk	8_32_512_2048_2	24.66	6.09	0.74
eng	akk	4_2_2048_1024_2	29.4	39.11	0.93	eng	akk	12_2_256_512_2	28.6	35.03	0.93
eng	akk	4_4_256_1024_2	30.18	38.92	0.93	eng	akk	12_2_1024_4096_2	28.09	37.46	0.93
eng	akk	4_6_256_512_2	29.87	38.81	0.93	eng	akk	12_4_256_256_2	26.68	28.97	0.93
eng	akk	4_6_512_2048_2	31.0	42.54	0.93	eng	akk	12_4_256_4096_2	28.29	34.0	0.93
eng	akk	4_6_1024_4096_2	29.65	42.02	0.93	eng	akk	12_4_512_1024_2	29.17	36.83	0.93
eng	akk	4_8_256_256_2	25.09	24.74	0.93	eng	akk	12_4_1024_2048_2	22.01	15.21	0.89
eng	akk	4_8_1024_4096_2	30.74	42.5	0.93	eng	akk	12_6_256_4096_2	27.56	30.33	0.93
eng	akk	4_12_256_4096_2	23.5	18.49	0.91	eng	akk	12_6_512_1024_2	29.52	37.17	0.93
eng	akk	4_12_512_512_2	22.87	13.21	0.89	eng	akk	12_6_1024_2048_2	29.04	38.79	0.93
eng	akk	4_12_1024_1024_2	30.45	42.45	0.93	eng	akk	12_8_512_512_2	28.55	35.78	0.93
eng	akk	4_12_1024_2048_2	30.45	42.74	0.93	eng	akk	12_8_1024_1024_2	20.41	16.14	0.9
eng	akk	4_16_512_256_2	30.3	39.35	0.93	eng	akk	12_12_512_256_2	22.66	18.15	0.91
eng	akk	4_16_1024_256_2	30.01	42.16	0.93	eng	akk	12_12_1024_256_2	21.42	18.64	0.91
eng	akk	4_16_1024_512_2	28.1	39.74	0.93	eng	akk	12_12_1024_512_2	22.83	16.31	0.9
eng	akk	4_24_256_2048_2	14.81	8.03	0.77	eng	akk	12_16_256_2048_2	17.54	8.26	0.78
eng	akk	4_24_512_4096_2	20.75	7.12	0.78	eng	akk	12_16_512_4096_2	19.53	10.25	0.84
eng	akk	6_2_256_1024_2	29.45	38.24	0.93	eng	akk	12_32_256_1024_2	17.82	9.23	0.8
eng	akk	6_2_512_4096_2	30.55	41.64	0.93	eng	akk	12_32_512_2048_2	18.07	9.72	0.82
eng	akk	6_2_2048_256_2	28.64	36.5	0.93	eng	akk	16_2_256_256_2	27.06	30.64	0.93
eng	akk	6_4_512_2048_2	29.93	39.88	0.94	eng	akk	16_2_256_4096_2	26.15	26.97	0.92
eng	akk	6_6_256_512_2	29.98	39.21	0.93	eng	akk	16_2_512_1024_2	27.54	34.01	0.93
eng	akk	6_6_512_2048_2	29.85	40.58	0.93	eng	akk	16_2_1024_2048_2	28.46	37.33	0.93
eng	akk	6_6_1024_4096_2	27.61	39.26	0.93	eng	akk	16_4_512_512_2	26.23	31.16	0.93
eng	akk	6_8_256_256_2	29.7	37.99	0.93	eng	akk	16_4_1024_2048_2	28.41	35.7	0.93
eng	akk	6_8_512_1024_2	30.04	40.92	0.93	eng	akk	16_6_512_512_2	28.35	35.79	0.93
eng	akk	6_8_1024_2048_2	30.97	43.39	0.93	eng	akk	16_6_1024_1024_2	26.32	28.94	0.91
eng	akk	6_12_256_4096_2	31.09	41.63	0.94	eng	akk	16_8_512_256_2	21.81	18.5	0.9
eng	akk	6_12_512_256_2	20.87	15.17	0.89	eng	akk	16_8_1024_512_2	20.68	17.16	0.89
eng	akk	6_12_512_512_2	22.29	21.44	0.92	eng	akk	16_8_1024_1024_2	28.35	36.26	0.92
eng	akk	6_12_1024_1024_2	22.22	14.15	0.9	eng	akk	16_12_256_2048_2	24.71	23.05	0.92
eng	akk	6_16_1024_256_2	30.27	41.0	0.94	eng	akk	16_12_512_4096_2	22.74	19.9	0.92

Table 14: Results for Experiment 2 (Part 2/6)

src	tgt	model	bleu	chrf	comet	src	tgt	model	bleu	chrf	comet
eng	akk	16_12_1024_256_2	20.4	15.52	0.89	deu	dsb	4_4_256_1024_2	28.09	43.01	0.62
eng	akk	16_16_256_2048_2	15.97	8.39	0.81	deu	dsb	4_6_256_512_2	26.91	42.86	0.62
eng	akk	16_16_512_4096_2	16.2	11.54	0.81	deu	dsb	4_6_512_2048_2	28.82	45.93	0.63
eng	akk	16_24_256_1024_2	17.39	8.28	0.8	deu	dsb	4_6_1024_4096_2	31.57	49.65	0.65
eng	akk	16_32_256_1024_2	24.86	6.13	0.74	deu	dsb	4_8_256_256_2	28.06	43.14	0.62
eng	akk	16_32_512_2048_2	16.7	7.76	0.82	deu	dsb	4_8_1024_4096_2	32.1	50.21	0.66
eng	akk	24_2_512_256_2	27.45	32.36	0.93	deu	dsb	4_12_256_4096_2	27.49	42.94	0.62
eng	akk	24_2_512_512_2	27.56	33.44	0.93	deu	dsb	4_12_512_512_2	28.78	45.54	0.63
eng	akk	24_2_1024_1024_2	27.51	33.86	0.93	deu	dsb	4_12_1024_1024_2	30.11	47.66	0.65
eng	akk	24_4_256_2048_2	21.96	12.81	0.9	deu	dsb	4_12_1024_2048_2	31.2	48.69	0.65
eng	akk	24_4_512_256_2	25.96	31.73	0.93	deu	dsb	4_16_512_256_2	28.8	46.63	0.64
eng	akk	24_4_1024_512_2	27.35	33.82	0.92	deu	dsb	4_16_1024_256_2	29.46	47.91	0.65
eng	akk	24_6_256_2048_2	21.88	11.6	0.87	deu	dsb	4_16_1024_512_2	30.57	48.87	0.65
eng	akk	24_6_512_4096_2	19.84	8.84	0.85	deu	dsb	4_24_256_2048_2	12.22	24.03	0.5
eng	akk	24_6_1024_256_2	27.85	35.54	0.92	deu	dsb	4_24_512_4096_2	15.34	29.54	0.53
eng	akk	24_6_1024_512_2	23.13	22.75	0.9	deu	dsb	6_2_256_1024_2	27.86	44.0	0.63
eng	akk	24_8_256_2048_2	21.87	8.91	0.82	deu	dsb	6_2_512_4096_2	29.87	47.84	0.64
eng	akk	24_8_512_4096_2	20.25	9.48	0.86	deu	dsb	6_2_2048_256_2	28.13	46.89	0.64
eng	akk	24_8_1024_256_2	26.93	32.41	0.92	deu	dsb	6_4_512_2048_2	28.34	45.13	0.64
eng	akk	24_24_256_1024_2	17.66	8.16	0.82	deu	dsb	6_6_256_512_2	27.4	42.87	0.62
eng	akk	24_24_512_2048_2	24.81	6.16	0.74	deu	dsb	6_6_512_2048_2	30.05	46.83	0.64
eng	akk	24_32_256_512_2	19.73	8.9	0.8	deu	dsb	6_6_1024_4096_2	32.08	50.72	0.66
eng	akk	32_2_256_2048_2	21.87	8.76	0.83	deu	dsb	6_8_256_256_2	27.48	42.48	0.62
eng	akk	32_2_512_4096_2	22.38	7.18	0.86	deu	dsb	6_8_512_1024_2	28.41	45.33	0.63
eng	akk	32_2_1024_256_2	20.99	15.41	0.9	deu	dsb	6_8_1024_2048_2	33.03	50.91	0.66
eng	akk	32_12_256_1024_2	23.7	17.46	0.91	deu	dsb	6_12_256_4096_2	28.67	44.24	0.63
eng	akk	32_16_256_1024_2	14.03	7.99	0.77	deu	dsb	6_12_512_256_2	29.46	46.16	0.64
eng	akk	32_16_512_2048_2	18.57	8.35	0.83	deu	dsb	6_12_512_512_2	28.63	45.56	0.64
eng	akk	32_32_256_512_2	19.11	8.39	0.82	deu	dsb	6_12_1024_1024_2	31.58	49.56	0.65
eng	akk	32_32_256_4096_2	24.52	6.24	0.74	deu	dsb	6_16_1024_256_2	30.51	48.94	0.65
eng	akk	32_32_512_1024_2	16.92	8.25	0.82	deu	dsb	6_16_1024_512_2	31.32	50.22	0.66
deu	dsb	2_2_2048_4096_2	28.96	46.8	0.65	deu	dsb	6_24_256_2048_2	10.04	13.03	0.46
deu	dsb	2_4_512_4096_2	29.43	46.96	0.64	deu	dsb	6_24_512_4096_2	8.71	15.32	0.45
deu	dsb	2_4_2048_256_2	29.03	47.44	0.64	deu	dsb	6_32_256_1024_2	10.96	24.02	0.5
deu	dsb	2_6_256_1024_2	28.35	44.33	0.63	deu	dsb	8_4_256_512_2	26.89	42.42	0.62
deu	dsb	2_8_256_512_2	28.84	43.92	0.63	deu	dsb	8_4_512_2048_2	29.05	46.24	0.64
deu	dsb	2_8_512_2048_2	27.02	44.16	0.63	deu	dsb	8_4_1024_4096_2	32.82	51.13	0.67
deu	dsb	2_8_1024_4096_2	29.98	47.89	0.65	deu	dsb	8_6_256_256_2	28.06	43.52	0.62
deu	dsb	2_12_512_1024_2	28.46	45.47	0.63	deu	dsb	8_6_512_1024_2	30.38	46.91	0.64
deu	dsb	2_12_1024_2048_2	29.99	47.56	0.64	deu	dsb	8_8_256_4096_2	27.89	43.46	0.62
deu	dsb	2_16_512_256_2	29.11	46.67	0.64	deu	dsb	8_8_512_1024_2	31.31	47.63	0.65
deu	dsb	2_16_1024_256_2	29.41	47.41	0.64	deu	dsb	8_8_1024_2048_2	31.24	49.36	0.65
deu	dsb	2_16_1024_512_2	30.36	48.81	0.65	deu	dsb	8_12_512_256_2	28.07	45.09	0.63
deu	dsb	2_16_1024_1024_2	30.68	49.03	0.65	deu	dsb	8_12_512_512_2	28.69	45.59	0.64
deu	dsb	2_24_256_2048_2	27.0	44.05	0.63	deu	dsb	8_12_1024_512_2	31.46	49.17	0.66
deu	dsb	2_24_512_4096_2	29.72	47.3	0.65	deu	dsb	8_12_1024_1024_2	31.66	50.01	0.66
deu	dsb	4_2_1024_256_2	29.47	46.86	0.64	deu	dsb	8_16_1024_256_2	31.34	50.58	0.66
deu	dsb	4_2_1024_512_2	29.71	48.32	0.65	deu	dsb	8_32_256_1024_2	11.97	26.74	0.51
deu	dsb	4_2_2048_512_2	27.07	45.25	0.63	deu	dsb	8_32_512_2048_2	13.85	30.58	0.53
deu	dsb	4_2_2048_1024_2	28.91	47.68	0.64	deu	dsb	12_2_256_512_2	27.53	42.04	0.61

Table 15: Results for Experiment 2 (Part 3/6)

src	tgt	model	bleu	chrf	comet	src	tgt	model	bleu	chrf	comet
deu	dsb	12_2_1024_4096_2	33.18	51.57	0.67	deu	dsb	24_24_512_2048_2	9.03	3.64	0.4
deu	dsb	12_4_256_256_2	28.59	43.07	0.62	deu	dsb	24_32_256_512_2	5.29	3.32	0.38
deu	dsb	12_4_256_4096_2	26.51	41.66	0.62	deu	dsb	32_2_256_2048_2	11.11	6.02	0.44
deu	dsb	12_4_512_1024_2	29.51	46.58	0.64	deu	dsb	32_2_512_4096_2	5.8	5.57	0.42
deu	dsb	12_4_1024_2048_2	30.87	49.04	0.65	deu	dsb	32_2_1024_256_2	15.96	22.55	0.52
deu	dsb	12_6_256_4096_2	26.13	40.19	0.6	deu	dsb	32_12_256_1024_2	12.44	10.4	0.46
deu	dsb	12_6_512_1024_2	30.35	47.13	0.65	deu	dsb	32_16_256_1024_2	12.44	10.01	0.46
deu	dsb	12_6_1024_2048_2	30.34	47.96	0.65	deu	dsb	32_16_512_2048_2	10.36	9.99	0.44
deu	dsb	12_8_512_512_2	28.85	44.68	0.63	deu	dsb	32_32_256_512_2	10.26	3.02	0.39
deu	dsb	12_8_1024_1024_2	30.41	48.14	0.65	deu	dsb	32_32_256_4096_2	5.24	3.13	0.41
deu	dsb	12_12_512_256_2	28.8	45.28	0.64	deu	dsb	32_32_512_1024_2	10.79	2.9	0.4
deu	dsb	12_12_1024_256_2	31.39	48.98	0.65	eng_wiki	ita_wiki	2_2_1024_1024_2	12.56	44.02	0.59
deu	dsb	12_12_1024_512_2	32.07	49.79	0.66	eng_wiki	ita_wiki	2_2_2048_4096_2	13.56	44.71	0.61
deu	dsb	12_16_256_2048_2	26.18	40.04	0.61	eng_wiki	ita_wiki	2_4_512_4096_2	13.86	44.95	0.61
deu	dsb	12_16_512_4096_2	26.25	41.22	0.61	eng_wiki	ita_wiki	2_4_2048_256_2	12.37	43.23	0.57
deu	dsb	12_32_256_1024_2	10.84	19.88	0.49	eng_wiki	ita_wiki	2_6_256_1024_2	13.43	44.79	0.61
deu	dsb	12_32_512_2048_2	7.47	3.75	0.4	eng_wiki	ita_wiki	2_8_256_512_2	13.1	44.67	0.6
deu	dsb	16_2_256_256_2	24.96	37.98	0.6	eng_wiki	ita_wiki	2_8_512_2048_2	12.93	44.14	0.59
deu	dsb	16_2_256_4096_2	15.16	17.72	0.49	eng_wiki	ita_wiki	2_8_1024_4096_2	13.74	44.96	0.61
deu	dsb	16_2_512_1024_2	17.76	24.74	0.54	eng_wiki	ita_wiki	2_12_512_1024_2	12.54	43.97	0.59
deu	dsb	16_2_1024_2048_2	18.64	28.2	0.55	eng_wiki	ita_wiki	2_12_1024_2048_2	13.1	44.71	0.61
deu	dsb	16_4_512_512_2	17.65	25.5	0.53	eng_wiki	ita_wiki	2_16_512_256_2	13.04	45.0	0.59
deu	dsb	16_4_1024_2048_2	17.98	27.12	0.55	eng_wiki	ita_wiki	2_16_1024_256_2	13.42	45.15	0.61
deu	dsb	16_6_512_512_2	18.54	26.36	0.54	eng_wiki	ita_wiki	2_16_1024_512_2	13.19	44.68	0.61
deu	dsb	16_6_1024_1024_2	17.22	27.33	0.54	eng_wiki	ita_wiki	2_16_1024_1024_2	13.42	44.85	0.61
deu	dsb	16_8_512_256_2	18.76	26.24	0.54	eng_wiki	ita_wiki	2_24_256_2048_2	12.31	43.76	0.58
deu	dsb	16_8_1024_512_2	18.9	28.61	0.55	eng_wiki	ita_wiki	2_24_512_4096_2	3.16	24.5	0.28
deu	dsb	16_8_1024_1024_2	18.49	28.1	0.55	eng_wiki	ita_wiki	4_2_1024_256_2	12.73	44.67	0.59
deu	dsb	16_12_256_2048_2	16.13	21.51	0.51	eng_wiki	ita_wiki	4_2_1024_512_2	12.76	44.19	0.59
deu	dsb	16_12_512_4096_2	17.47	25.2	0.53	eng_wiki	ita_wiki	4_2_2048_512_2	13.04	44.83	0.59
deu	dsb	16_12_1024_256_2	18.15	29.32	0.55	eng_wiki	ita_wiki	4_2_2048_1024_2	13.91	45.42	0.62
deu	dsb	16_16_256_2048_2	16.62	20.96	0.51	eng_wiki	ita_wiki	4_4_256_1024_2	13.4	44.92	0.61
deu	dsb	16_16_512_4096_2	17.78	24.4	0.53	eng_wiki	ita_wiki	4_6_256_512_2	13.68	45.07	0.61
deu	dsb	16_24_256_1024_2	8.87	3.16	0.4	eng_wiki	ita_wiki	4_6_512_2048_2	14.15	45.72	0.62
deu	dsb	16_32_256_1024_2	6.44	4.5	0.39	eng_wiki	ita_wiki	4_6_1024_4096_2	14.7	46.49	0.65
deu	dsb	16_32_512_2048_2	10.55	4.15	0.41	eng_wiki	ita_wiki	4_8_256_256_2	12.97	44.34	0.59
deu	dsb	24_2_512_256_2	16.25	24.86	0.52	eng_wiki	ita_wiki	4_8_1024_4096_2	14.53	45.99	0.63
deu	dsb	24_2_512_512_2	16.83	24.89	0.52	eng_wiki	ita_wiki	4_12_256_4096_2	11.88	42.9	0.56
deu	dsb	24_2_1024_1024_2	18.11	25.05	0.54	eng_wiki	ita_wiki	4_12_512_512_2	13.4	45.14	0.6
deu	dsb	24_4_256_2048_2	12.05	10.72	0.46	eng_wiki	ita_wiki	4_12_1024_1024_2	13.84	45.25	0.62
deu	dsb	24_4_512_256_2	17.07	24.17	0.53	eng_wiki	ita_wiki	4_12_1024_2048_2	13.91	45.54	0.62
deu	dsb	24_4_1024_512_2	17.84	26.84	0.54	eng_wiki	ita_wiki	4_16_512_256_2	13.58	45.21	0.62
deu	dsb	24_6_256_2048_2	13.07	12.1	0.47	eng_wiki	ita_wiki	4_16_1024_256_2	13.92	45.8	0.62
deu	dsb	24_6_512_4096_2	11.96	15.7	0.47	eng_wiki	ita_wiki	4_16_1024_512_2	14.01	45.91	0.62
deu	dsb	24_6_1024_256_2	18.5	24.87	0.54	eng_wiki	ita_wiki	4_24_256_2048_2	3.22	26.13	0.28
deu	dsb	24_6_1024_512_2	18.55	26.37	0.55	eng_wiki	ita_wiki	4_24_512_4096_2	2.35	22.59	0.26
deu	dsb	24_8_256_2048_2	12.73	12.56	0.46	eng_wiki	ita_wiki	6_2_256_1024_2	13.81	45.35	0.61
deu	dsb	24_8_512_4096_2	13.9	16.93	0.5	eng_wiki	ita_wiki	6_2_512_4096_2	14.02	45.48	0.62
deu	dsb	24_8_1024_256_2	17.26	26.91	0.53	eng_wiki	ita_wiki	6_2_2048_256_2	14.07	45.9	0.62
deu	dsb	24_24_256_1024_2	7.41	4.06	0.41	eng_wiki	ita_wiki	6_4_512_2048_2	14.59	46.28	0.63

Table 16: Results for Experiment 2 (Part 4/6)

src	tgt	model	bleu	chrf	comet	src	tgt	model	bleu	chrf	comet
eng_wiki	ita_wiki	6_6_256_512_2	13.51	45.14	0.61	eng_wiki	ita_wiki	16_2_512_1024_2	3.76	25.12	0.35
eng_wiki	ita_wiki	6_6_512_2048_2	14.27	45.57	0.61	eng_wiki	ita_wiki	16_2_1024_2048_2	4.01	26.05	0.35
eng_wiki	ita_wiki	6_6_1024_4096_2	15.27	46.69	0.65	eng_wiki	ita_wiki	16_4_512_512_2	4.03	26.47	0.37
eng_wiki	ita_wiki	6_8_256_256_2	13.61	45.16	0.61	eng_wiki	ita_wiki	16_4_1024_2048_2	3.89	26.76	0.38
eng_wiki	ita_wiki	6_8_512_1024_2	13.73	45.33	0.61	eng_wiki	ita_wiki	16_6_512_512_2	4.05	25.89	0.38
eng_wiki	ita_wiki	6_8_1024_2048_2	14.4	46.14	0.65	eng_wiki	ita_wiki	16_6_1024_1024_2	4.02	27.6	0.38
eng_wiki	ita_wiki	6_12_256_4096_2	12.2	43.39	0.56	eng_wiki	ita_wiki	16_8_512_256_2	3.96	27.52	0.38
eng_wiki	ita_wiki	6_12_512_256_2	13.54	45.62	0.61	eng_wiki	ita_wiki	16_8_1024_512_2	4.22	28.34	0.38
eng_wiki	ita_wiki	6_12_512_512_2	13.39	45.05	0.61	eng_wiki	ita_wiki	16_8_1024_1024_2	4.0	28.29	0.38
eng_wiki	ita_wiki	6_12_1024_1024_2	14.61	46.4	0.64	eng_wiki	ita_wiki	16_12_256_2048_2	3.48	23.67	0.37
eng_wiki	ita_wiki	6_16_1024_256_2	14.4	46.27	0.63	eng_wiki	ita_wiki	16_12_512_4096_2	3.63	25.0	0.38
eng_wiki	ita_wiki	6_16_1024_512_2	14.41	46.48	0.64	eng_wiki	ita_wiki	16_12_1024_256_2	4.16	28.04	0.39
eng_wiki	ita_wiki	6_24_256_2048_2	3.03	25.75	0.28	eng_wiki	ita_wiki	16_16_256_2048_2	3.47	23.38	0.37
eng_wiki	ita_wiki	6_24_512_4096_2	2.25	20.31	0.25	eng_wiki	ita_wiki	16_16_512_4096_2	3.69	24.9	0.38
eng_wiki	ita_wiki	6_32_256_1024_2	2.59	23.99	0.27	eng_wiki	ita_wiki	16_24_256_1024_2	1.41	5.47	0.22
eng_wiki	ita_wiki	8_4_256_512_2	13.44	45.07	0.6	eng_wiki	ita_wiki	16_32_256_1024_2	1.33	2.99	0.21
eng_wiki	ita_wiki	8_4_512_2048_2	14.38	46.27	0.63	eng_wiki	ita_wiki	16_32_512_2048_2	1.38	6.65	0.22
eng_wiki	ita_wiki	8_4_1024_4096_2	15.74	47.32	0.67	eng_wiki	ita_wiki	24_2_512_256_2	3.79	24.36	0.35
eng_wiki	ita_wiki	8_6_256_256_2	13.37	44.82	0.6	eng_wiki	ita_wiki	24_2_512_512_2	3.65	23.5	0.34
eng_wiki	ita_wiki	8_6_512_1024_2	13.99	45.65	0.62	eng_wiki	ita_wiki	24_2_1024_1024_2	3.69	25.0	0.37
eng_wiki	ita_wiki	8_8_256_4096_2	12.89	44.41	0.58	eng_wiki	ita_wiki	24_4_256_2048_2	3.04	21.33	0.36
eng_wiki	ita_wiki	8_8_512_1024_2	13.97	45.77	0.62	eng_wiki	ita_wiki	24_4_512_256_2	3.64	25.14	0.37
eng_wiki	ita_wiki	8_8_1024_2048_2	15.01	46.59	0.65	eng_wiki	ita_wiki	24_4_1024_512_2	3.73	24.99	0.36
eng_wiki	ita_wiki	8_12_512_256_2	13.7	45.62	0.61	eng_wiki	ita_wiki	24_6_256_2048_2	2.86	20.14	0.35
eng_wiki	ita_wiki	8_12_512_512_2	13.59	45.17	0.61	eng_wiki	ita_wiki	24_6_512_4096_2	2.85	20.86	0.38
eng_wiki	ita_wiki	8_12_1024_512_2	14.92	46.66	0.65	eng_wiki	ita_wiki	24_6_1024_256_2	3.77	26.25	0.38
eng_wiki	ita_wiki	8_12_1024_1024_2	15.06	46.97	0.65	eng_wiki	ita_wiki	24_6_1024_512_2	3.59	26.53	0.38
eng_wiki	ita_wiki	8_16_1024_256_2	14.88	47.08	0.65	eng_wiki	ita_wiki	24_8_256_2048_2	2.93	20.88	0.36
eng_wiki	ita_wiki	8_32_256_1024_2	3.23	26.48	0.28	eng_wiki	ita_wiki	24_8_512_4096_2	2.93	22.32	0.38
eng_wiki	ita_wiki	8_32_512_2048_2	3.68	28.03	0.29	eng_wiki	ita_wiki	24_8_1024_256_2	3.57	26.02	0.38
eng_wiki	ita_wiki	12_2_256_512_2	13.02	44.15	0.59	eng_wiki	ita_wiki	24_24_256_1024_2	1.81	2.93	0.22
eng_wiki	ita_wiki	12_2_1024_4096_2	16.12	47.89	0.66	eng_wiki	ita_wiki	24_24_512_2048_2	1.38	2.46	0.22
eng_wiki	ita_wiki	12_4_256_256_2	13.52	45.23	0.61	eng_wiki	ita_wiki	24_32_256_512_2	1.48	7.86	0.23
eng_wiki	ita_wiki	12_4_256_4096_2	13.37	44.22	0.59	eng_wiki	ita_wiki	32_2_256_2048_2	2.56	18.81	0.33
eng_wiki	ita_wiki	12_4_512_1024_2	13.87	45.61	0.62	eng_wiki	ita_wiki	32_2_512_4096_2	2.24	17.43	0.33
eng_wiki	ita_wiki	12_4_1024_2048_2	15.61	47.63	0.66	eng_wiki	ita_wiki	32_2_1024_256_2	2.76	18.58	0.34
eng_wiki	ita_wiki	12_6_256_4096_2	12.64	43.68	0.58	eng_wiki	ita_wiki	32_12_256_1024_2	2.68	20.44	0.34
eng_wiki	ita_wiki	12_6_512_1024_2	14.0	45.74	0.61	eng_wiki	ita_wiki	32_16_256_1024_2	2.67	20.64	0.35
eng_wiki	ita_wiki	12_6_1024_2048_2	15.34	47.16	0.66	eng_wiki	ita_wiki	32_16_512_2048_2	2.46	19.93	0.36
eng_wiki	ita_wiki	12_8_512_512_2	14.28	46.03	0.62	eng_wiki	ita_wiki	32_32_256_512_2	1.66	3.55	0.22
eng_wiki	ita_wiki	12_8_1024_1024_2	14.57	46.41	0.63	eng_wiki	ita_wiki	32_32_256_4096_2	1.41	1.6	0.2
eng_wiki	ita_wiki	12_12_512_256_2	13.37	45.32	0.6	eng_wiki	ita_wiki	32_32_512_1024_2	1.55	2.37	0.21
eng_wiki	ita_wiki	12_12_1024_256_2	15.24	47.12	0.64	eng	mni	2_2_1024_1024_2	20.5	47.35	0.7
eng_wiki	ita_wiki	12_12_1024_512_2	14.67	46.45	0.64	eng	mni	2_2_2048_4096_2	21.32	47.97	0.7
eng_wiki	ita_wiki	12_16_256_2048_2	11.27	42.02	0.54	eng	mni	2_4_512_4096_2	20.97	47.45	0.7
eng_wiki	ita_wiki	12_16_512_4096_2	11.97	42.82	0.56	eng	mni	2_4_2048_256_2	21.57	48.18	0.7
eng_wiki	ita_wiki	12_32_256_1024_2	1.29	3.86	0.21	eng	mni	2_6_256_1024_2	19.46	45.8	0.7
eng_wiki	ita_wiki	12_32_512_2048_2	3.77	27.69	0.29	eng	mni	2_8_256_512_2	18.62	45.96	0.7
eng_wiki	ita_wiki	16_2_256_256_2	11.92	42.5	0.56	eng	mni	2_8_512_2048_2	20.65	47.48	0.7
eng_wiki	ita_wiki	16_2_256_4096_2	3.27	21.27	0.35	eng	mni	2_8_1024_4096_2	21.7	48.53	0.71

Table 17: Results for Experiment 2 (Part 5/6)

src	tgt	model	bleu	chrf	comet	src	tgt	model	bleu	chrf	comet
eng	mni	2_12_512_1024_2	20.96	48.15	0.71	eng	mni	8_6_512_1024_2	20.32	47.29	0.71
eng	mni	2_12_1024_2048_2	21.94	49.38	0.71	eng	mni	8_8_256_4096_2	18.93	41.98	0.68
eng	mni	2_16_512_256_2	20.96	47.99	0.71	eng	mni	8_8_512_1024_2	19.81	46.04	0.7
eng	mni	2_16_1024_256_2	21.51	49.88	0.71	eng	mni	8_8_1024_2048_2	21.33	48.04	0.71
eng	mni	2_16_1024_512_2	21.52	49.7	0.71	eng	mni	8_12_512_256_2	19.55	46.84	0.71
eng	mni	2_16_1024_1024_2	21.68	48.97	0.71	eng	mni	8_12_512_512_2	20.57	47.32	0.71
eng	mni	2_24_256_2048_2	18.53	45.07	0.69	eng	mni	8_12_1024_512_2	21.6	48.25	0.71
eng	mni	2_24_512_4096_2	21.14	48.14	0.7	eng	mni	8_12_1024_1024_2	21.16	47.83	0.71
eng	mni	4_2_1024_256_2	20.08	46.9	0.7	eng	mni	8_16_1024_256_2	20.48	47.36	0.71
eng	mni	4_2_1024_512_2	20.46	47.53	0.7	eng	mni	8_32_256_1024_2	4.71	18.75	0.45
eng	mni	4_2_2048_512_2	19.91	46.03	0.69	eng	mni	8_32_512_2048_2	8.73	29.97	0.54
eng	mni	4_2_2048_1024_2	20.23	46.64	0.69	eng	mni	12_2_256_512_2	14.79	39.85	0.68
eng	mni	4_4_256_1024_2	17.38	44.5	0.69	eng	mni	12_2_1024_4096_2	19.94	45.84	0.7
eng	mni	4_6_256_512_2	14.69	41.51	0.68	eng	mni	12_4_256_256_2	15.01	39.36	0.68
eng	mni	4_6_512_2048_2	20.61	47.51	0.7	eng	mni	12_4_256_4096_2	17.39	38.81	0.68
eng	mni	4_6_1024_4096_2	22.28	49.07	0.71	eng	mni	12_4_512_1024_2	19.2	44.07	0.7
eng	mni	4_8_256_256_2	17.69	44.48	0.69	eng	mni	12_4_1024_2048_2	19.98	44.33	0.69
eng	mni	4_8_1024_4096_2	22.02	49.02	0.71	eng	mni	12_6_256_4096_2	17.57	38.62	0.68
eng	mni	4_12_256_4096_2	19.78	45.44	0.7	eng	mni	12_6_512_1024_2	19.01	43.38	0.69
eng	mni	4_12_512_512_2	20.77	47.58	0.71	eng	mni	12_6_1024_2048_2	19.39	44.1	0.69
eng	mni	4_12_1024_1024_2	21.61	49.21	0.71	eng	mni	12_8_512_512_2	18.57	43.11	0.69
eng	mni	4_12_1024_2048_2	21.86	48.87	0.71	eng	mni	12_8_1024_1024_2	19.74	44.36	0.69
eng	mni	4_16_512_256_2	20.2	47.73	0.71	eng	mni	12_12_512_256_2	18.94	43.92	0.7
eng	mni	4_16_1024_256_2	21.86	48.67	0.71	eng	mni	12_12_1024_256_2	20.07	44.85	0.7
eng	mni	4_16_1024_512_2	21.74	49.41	0.71	eng	mni	12_12_1024_512_2	19.2	45.0	0.7
eng	mni	4_24_256_2048_2	19.64	45.94	0.7	eng	mni	12_16_256_2048_2	17.19	39.29	0.68
eng	mni	4_24_512_4096_2	4.15	16.81	0.42	eng	mni	12_16_512_4096_2	19.34	43.19	0.69
eng	mni	6_2_256_1024_2	16.71	43.44	0.69	eng	mni	12_32_256_1024_2	3.96	16.01	0.42
eng	mni	6_2_512_4096_2	20.56	47.24	0.7	eng	mni	12_32_512_2048_2	3.51	16.37	0.41
eng	mni	6_2_2048_256_2	19.46	45.47	0.69	eng	mni	16_2_256_256_2	12.51	36.0	0.66
eng	mni	6_4_512_2048_2	20.61	47.66	0.71	eng	mni	16_2_256_4096_2	15.8	36.51	0.66
eng	mni	6_6_256_512_2	17.38	44.22	0.69	eng	mni	16_2_512_1024_2	17.12	41.31	0.69
eng	mni	6_6_512_2048_2	20.14	46.7	0.7	eng	mni	16_2_1024_2048_2	19.83	43.59	0.69
eng	mni	6_6_1024_4096_2	21.54	48.65	0.71	eng	mni	16_4_512_512_2	18.0	41.83	0.69
eng	mni	6_8_256_256_2	17.31	44.25	0.7	eng	mni	16_4_1024_2048_2	18.93	43.39	0.68
eng	mni	6_8_512_1024_2	20.77	48.13	0.71	eng	mni	16_6_512_512_2	17.97	42.24	0.69
eng	mni	6_8_1024_2048_2	21.45	47.59	0.7	eng	mni	16_6_1024_1024_2	18.44	42.31	0.68
eng	mni	6_12_256_4096_2	19.68	45.01	0.7	eng	mni	16_8_512_256_2	17.6	41.4	0.69
eng	mni	6_12_512_256_2	20.82	48.02	0.71	eng	mni	16_8_1024_512_2	19.05	42.87	0.68
eng	mni	6_12_512_512_2	20.82	48.3	0.71	eng	mni	16_8_1024_1024_2	19.36	42.82	0.68
eng	mni	6_12_1024_1024_2	21.23	49.09	0.71	eng	mni	16_12_256_2048_2	16.53	38.3	0.67
eng	mni	6_16_1024_256_2	21.32	48.92	0.71	eng	mni	16_12_512_4096_2	18.58	41.19	0.68
eng	mni	6_16_1024_512_2	21.22	48.73	0.71	eng	mni	16_12_1024_256_2	19.05	43.73	0.69
eng	mni	6_24_256_2048_2	19.62	46.48	0.7	eng	mni	16_16_256_2048_2	17.28	38.61	0.67
eng	mni	6_24_512_4096_2	4.63	17.12	0.43	eng	mni	16_16_512_4096_2	19.24	41.86	0.68
eng	mni	6_32_256_1024_2	2.69	11.42	0.37	eng	mni	16_24_256_1024_2	2.55	7.92	0.39
eng	mni	8_4_256_512_2	15.21	41.64	0.68	eng	mni	16_32_256_1024_2	3.52	14.28	0.4
eng	mni	8_4_512_2048_2	20.79	47.27	0.71	eng	mni	16_32_512_2048_2	2.8	12.5	0.4
eng	mni	8_4_1024_4096_2	21.31	47.73	0.7	eng	mni	24_2_512_256_2	15.61	40.47	0.68
eng	mni	8_6_256_256_2	16.37	42.96	0.69	eng	mni	24_2_512_512_2	16.97	40.52	0.68

Table 18: Results for Experiment 2 (Part 6/6)

src	tgt	model	bleu	chrf	comet
eng	mni	24_2_1024_1024_2	18.43	42.35	0.68
eng	mni	24_4_256_2048_2	14.6	34.64	0.66
eng	mni	24_4_512_256_2	15.55	38.65	0.67
eng	mni	24_4_1024_512_2	18.81	42.09	0.68
eng	mni	24_6_256_2048_2	15.31	34.31	0.66
eng	mni	24_6_512_4096_2	15.34	33.19	0.65
eng	mni	24_6_1024_256_2	18.17	41.86	0.68
eng	mni	24_6_1024_512_2	18.12	41.28	0.68
eng	mni	24_8_256_2048_2	15.26	34.28	0.66
eng	mni	24_8_512_4096_2	14.54	32.52	0.65
eng	mni	24_8_1024_256_2	17.72	40.74	0.67
eng	mni	24_24_256_1024_2	3.18	12.24	0.4
eng	mni	24_24_512_2048_2	2.8	12.81	0.4
eng	mni	24_32_256_512_2	2.7	11.38	0.38
eng	mni	32_2_256_2048_2	7.27	24.19	0.61
eng	mni	32_2_512_4096_2	2.29	13.55	0.51
eng	mni	32_2_1024_256_2	17.45	39.5	0.67
eng	mni	32_12_256_1024_2	12.51	29.79	0.6
eng	mni	32_16_256_1024_2	15.71	33.21	0.6
eng	mni	32_16_512_2048_2	13.77	31.34	0.6
eng	mni	32_32_256_512_2	2.4	10.86	0.38
eng	mni	32_32_256_4096_2	1.27	8.05	0.37
eng	mni	32_32_512_1024_2	1.81	7.73	0.38

IfGPT: A Dataset in Bulgarian for Large Language Models

Svetla Koeva
DCL – IBL,
Bulgarian Academy of Sciences, Sofia, Bulgaria
svetla@dcl.bas.bg

Ivelina Stoyanova
DCL – IBL,
Bulgarian Academy of Sciences, Sofia, Bulgaria
iva@dcl.bas.bg

Jordan Kralev
DCL – IBL;
Technological University
Sofia, Bulgaria
jkralev@dcl.bas.bg

Abstract

The paper presents the large dataset **IfGPT**, which contains available corpora and datasets for Bulgarian, and describes methods to continuously expand it with unduplicated and unbiased Bulgarian data. The samples in the dataset are annotated with metadata that enable effective extraction of domain- and application-oriented datasets for fine-tuning or Retrieval Augmented Generation (RAG) of large language models (LLMs). The paper focuses on the description of the extended metadata of the **IfGPT** dataset and its management in a graph database.

1 Introduction

The large-scale transformer-based models (Vaswani et al., 2017) have significantly changed the state of the art in language processing. There are two basic steps in the development of LLMs, both of which have to do with datasets: Pre-training on large text data and subsequent fine-tuning for a specific task with suitable data.

Developing datasets for LLMs is a major challenge for languages with limited resources. These include:

Data scarcity There are few sources for compiling large datasets for pre-training and fine-tuning LLMs for languages such as Bulgarian, whose relatively low production of authentic digital texts is predetermined by the relatively small number of its speakers.[1]

Copyright restrictions It is even more difficult to find datasets that do not raise copyright issues and are available for both non-commercial and commercial use.[2]

Quality of the data Freely accessible data is often noisy and inhomogeneous and can therefore cause problems or lead to distortions. Procedures for data cleansing and selecting only high-quality texts further limit the scope of the data.

In this paper, we present the **large dataset IfGPT**,[3] which contains some already available corpora and datasets for Bulgarian, as well as methods for its continuous expansion with non-duplicated, clean Bulgarian data. The samples in the dataset are annotated with metadata that enable effective extraction of domain- and application-oriented datasets. The paper focuses on the description of the extended metadata of the **IfGPT** dataset and its management in a graph-based database.

The aim is to avoid the redundant compilation of datasets by different users and the multiple efforts for cleaning the data and to facilitate the reuse of the data for solving different application tasks. The main contribution of our work can be summarised as follows:

(a) Merging several relatively large text collections for Bulgarian into one dataset with standardised metadata description and document formats.

(b) Adding new texts to the dataset in a standardised way.

(c) Deploying and customising a set of tools in a chain for text cleaning, deduplication, detection of sensitive and biassed information to ensure the quality of the data.

(d) Providing a uniform metadata description for all documents in the datasets and organising the metadata categories in a graph representation, originally proposed for the Bulgarian National Corpus (Koeva et al., 2012) and extended to the present **IfGPT** dataset.

(e) Providing means to efficiently query metadata to find suitable text documents for a given

[1] https://datareportal.com/reports/digital-2025-bulgaria

[2] The Bulgarian Intellectual Property Rights Act of 2023 liberalises the use of texts that are accessible digitally or in digital form for automatic analysis, but some proprietary collections that are protected by copyright and are not accessible.

[3] https://ifgpt.dcl.bas.bg/en/

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 65–75, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-007

LLM fine-tuning or Retrieval Augmented Generation (RAG) task.

2 Large text datasets

Recent advances in the development of LLMs have demonstrated the effectiveness of their pre-training on large text datasets. Despite the fact that some technologies enable shorter parts of training datasets for specific domains and/or languages, the growing demand for language modelling data for most languages, including Bulgarian, remains a challenge. Here we will briefly present some of the widely used and recently created large text datasets used for pre-training.

CommonCrawl creates and maintains an open web crawl dataset. Since 2008, CommonCrawl has collected petabytes of data, including raw web page data, metadata, and text extractions. CommonCrawl is typically used to retrieve subsets of websites during a specific time period. Due to the noisy and low-quality information in web data (Luccioni and Viviano, 2021), it is necessary to clean and filter the data before using it. There are a number of filtered datasets based on CommonCrawl including Bulgarian. **OSCAR** (Open Super-large Crawled Aggregated coRpus) is a large multilingual corpus created by language classification and filtering of the CommonCrawl dataset (Abadji et al., 2022). It covers 152 languages and offers both original and deduplicated versions of the data. Similarly, the **C4** (Raffel et al., 2020) and **mC4** (Xue et al., 2020) datasets were derived from Common Crawl. These corpora were created using heuristic methods to filter out non-linguistic content (such as boilerplate or noise) and underwent extensive deduplication. While **C4** was developed for English only, **mC4** covers over 100 languages. Another related resource is **CC100** (Conneau et al., 2020), which provides monolingual data for more than 100 languages. It was created by processing CommonCrawl snapshots collected between January and December 2018.

Many of the large datasets do not contain Bulgarian, e.g. **Pile**, an 825 GB English text corpus developed for large-scale language model training (Gao et al., 2020); **MassiveText**, a collection of large English language text datasets from various sources, including websites, books, news articles and code (Rae et al., 2022), etc.

There are several studies that present available datasets and categorise them under different aspects: (1) Pre-training Corpora; (2) Instruction Fine-tuning Datasets; (3) Preference Datasets; (4) Evaluation Datasets; (5) Traditional Natural Language Processing Corpora (Liu et al., 2024; Lu et al., 2024). The **IfGPT** dataset presented here can be used as (part of) a pre-training dataset, a fine-tuning dataset (with some modifications), an evaluation dataset (with some modifications), and a traditional natural language processing dataset. However, our motivation for its compilation, management and extension is the fine-tuning of LLMs or RAG applications.

3 Data sources for IfGPT dataset

When collecting and pre-processing data for fine-tuning LLMs, the aim is to collect as much diverse Bulgarian language data as possible that is human-generated, of high quality, does not contain sensitive, false or ethically unacceptable information, is not repetitive and is accompanied by accurate information about its source and the licence for its use.

The components of the **IfGPT** dataset can be categorised into three main groups depending on the type of text, its composition and its possible uses: 1) collections of texts (corpora) that have already been created and processed and are available to us, 2) other existing datasets of Bulgarian texts that need to be reviewed, downloaded and, if necessary, the format of the texts and metadata converted to the format and metadata of the **IfGPT** dataset, 3) compilation of new datasets through targeted crawling and processing of the identified texts for filtering, cleaning, deduplicating and adding metadata.

3.1 Brief description of existing text collections (corpora)

The existing text collections include corpora created for linguistic and corpus-related studies and corpora created for various NLP projects, e.g. for training machine translation systems. The **Bulgarian National Corpus (BulNC)** contains a wide range of texts of different sizes, different media types (written and spoken), different styles, different time periods (synchronous and diachronic) and different licences. Each text in the collection is labelled with metadata (Koeva et al., 2012). BulNC was originally compiled from the Bulgarian Lexicographical Archive and the Text Archive for Written Bulgarian, which make up 55.95% of the corpus.

Later, the EMEA corpus (medical administrative texts) and the OpenSubtitles corpus (film subtitles) were added, accounting for 1.27% and 8.61% of BulNC respectively. The remaining texts were automatically crawled and include a large number of administrative texts, news from monolingual and multilingual sources, scientific texts and popular science texts. The BulNC currently contains around 420,000,000 words and more than 10,000 text samples. Each text sample is provided with a detailed metadata description in a separate file, which makes it possible to extract subcorpora from specific domains and, if permitted, to distribute them with original licences. The texts are stored both in a word-per-line format (or 'vertical' format, in which each line contains a token, its lemma, the part of speech and grammatical features) and in a raw text format.

The dataset **General News in Bulgarian** contains news from different thematic domains. The news items and their metadata were collected automatically from various (mainly Bulgarian) Internet sources: 11,840 web domains and 2,116,739 web pages. The total number of words in the collected general news in Bulgarian language amounts to 601,330,975 words, spread over 33,375,366 sentences and about 28,000 texts. A crawling platform was used for the identification and collection of monolingual data from web pages, the removal of near-duplicates at the document level, and text normalisation and cleaning (Koeva et al., 2020). The extracted texts were structured into JSON files containing extracted metadata and an automatic categorisation of the content into 185 thematic domains (ordered by probability). The main domains with the largest number of documents are: Economics; Sociology; Politics; Law; Business; Commerce; Education; Administration; School; Leisure; and History. The links to the original sources and the distribution licences (if indicated in the sources) are part of the metadata.

The corpus **Bulgarian CURLICAT (Curated Multilingual Language Resources for CEF.AT)** consists of texts from various sources (Váradi et al., 2022). The collection comprises 113,087 documents divided into seven thematic domains: Culture, Education, European Union, Finance, Politics, Economy and Science. All documents are licenced under CC-BY, CC-BY-SA and CC-BY-NC. The texts are linguistically annotated and are available in CoNLL-U Plus format.[4]

The corpus **Bulgarian MARCELL (Multilingual resources for CEF.AT in the legal domain)** consists of legislative documents divided into fifteen types (Váradi et al., 2020). The time span of the documents ranges from 1946 to 2023 and the texts were extracted from the Bulgarian State Gazette, the official gazette of the Bulgarian government, in which documents from official institutions such as the government, the Bulgarian National Assembly, the Constitutional Court, etc. are published. The Bulgarian corpus consists of 25,283 documents categorised into eleven types: Administrative Court; Agreements; Amendments, legal acts; Conventions; Decrees; Decrees of the Council of Ministers; Directives; Instructions; Laws (legal acts); Memoranda; Resolutions. The documents were annotated in CoNLL-U Plus format. The dataset comprises around 45,000,000 tokens and 3,281,000 sentences.

Our work on the datasets already available to us is currently focused on three directions: Identifying texts that are suitable for distribution (with appropriate licences and not duplicated in other selected parts); standardising the format of the texts provided in addition to the original formats, raw text format and JSONL format;[5] and, where necessary, harmonising metadata (categories and values).

3.2 Use of other available datasets

In recent years, many large datasets have been created and gradually expanded with new data, with a focus on open datasets without usage restrictions. These include CommonCrawl;[6] and its cleaned derivatives such as C4[7] and CC-100;[8] OPUS Corpora;[9] etc.

Datasets are also distributed via well-known language repositories such as **ELG**, **CLARIN**, **GitHub**, **HuggingFace**, etc. For example, at the time of writing, HuggingFace has 258 text datasets containing Bulgarian; the ELG catalogue has 388 corpora containing Bulgarian; etc.

The main problems with these are that: (a) Bulgarian and other low-resource languages are rarely included; (b) if they are, they are only a small part

[4] https://universaldependencies.org/ext-format.html

[5] https://jsonlines.org/

[6] https://commoncrawl.org

[7] https://github.com/google-research/text-to-text-transfer-transformer#c4

[8] https://data.statmt.org/cc-100/

[9] https://opus.nlpl.eu/

of the data; (c) they may already be included in the datasets available to us; (d) they may not fulfil the quality requirements both in terms of overall data quality and suitability for training; (e) their availability on the web often means that they have already been included in the LLMs.

The aim here is to avoid overlaps with texts that have already been collected and to carry out a massive textual clean-up in order to filter out malformed texts and irrelevant data. The next step is to assign as many metadata as possible and convert the documents into the standardised format.

3.3 Compilation of new datasets through targeted crawling

A regularly updated source for the provision of new text data has been identified:

(a) Repositories for scientific papers, dissertations and other research publications such as: **Bulgarian Portal for Open Science**, a platform providing free access to full texts of articles published in Bulgarian scientific journals, selected scientific books together with extensive bibliographic metadata, etc.; scientific and popular science journals and blogs; websites of universities and research institutions publishing scientific papers, dissertations, etc. from various domains.

(b) **Public administrative data** provided by the Bulgarian National Assembly (parliamentary minutes and the Government Gazette), ministries, agencies and municipalities.

(c) Data from **websites and technical documentation of companies** from various domains and with appropriate licences.

(d) **Websites of media**: newspapers, television and radio stations that publish news from various domains and have appropriate licences.

Sources that have already been used for the collection of resources (see 3.1) can be monitored and crawled to update the datasets. When adding new text samples, the same format of the text, metadata and annotations is used to ensure compatibility with the procedures for validation, data enrichment and extraction of subsets of the data. In addition, the metadata provides a reliable means of filtering data (by source, year, domain, etc.) for more efficient deduplication (see 3.4.1).

To this end, we need reliable means to assess data diversity and techniques to improve it. Particular attention should be paid to less frequent linguistic phenomena, which firstly are not well captured in smaller datasets and secondly are crucial for ensuring and maintaining linguistic diversity.

One of the biggest challenges is to find and use data with suitable licences that allow sharing of the data (as part of the dataset). Many existing datasets disregard the restrictions on sharing and consider it sufficient to provide appropriate references to the source and authorship of the text samples.

3.4 Procedures for improving the quality of the dataset

Any application that needs to reliably represent a domain requires diverse, balanced and unbiased data. The following techniques are important to provide high quality data.

3.4.1 Removing duplicates

Deduplication has been shown to improve the quality of data and the performance of LLMs, in particular by removing overlap between training and test data, allowing for more reliable evaluation (Lee et al., 2022).

The first pre-filtering step relies on metadata and involves matching texts by source, year, domain, title, author, etc. to quickly identify and remove identical text samples which come from different dataset sources. This significantly improves the efficiency of further deduplication.

The main deduplication method we implement is based on the MinHash and Locality Sensitive Hashing (LSH) algorithm, which is widely used for this purpose (Leskovec et al., 2020; Lee et al., 2022; Albalak et al., 2024) and which provides an efficient way to identify even near-duplicates. The algorithm estimates the n-gram similarity between all pairs of text samples and identifies those with high n-gram overlap.

The deduplication procedures are implemented in a pipeline to facilitate ongoing deduplication as the dataset is regularly updated with new texts.

3.4.2 Handling formatting, boilerplate, web navigation elements from texts

For the extraction of raw text from HTML documents, we used CSS selectors to mark the elements we wanted to extract. In addition, various techniques are used for raw text extraction (Koeva et al., 2020): automatic correction of hyphenated words based on vocabulary, regular expressions to filter out metadata, sentence tokenisation and language detection to filter out non-Bulgarian sentences, etc. Since there are also PDF documents, we used a

PDF to text converter to extract text data. Additional scripts were written to remove headers and footers from the PDF documents. The extracted paragraphs were merged based on a heuristic analysis of capitalisation and lexical content when a sentence crossed a paragraph boundary. Scanned and OCR-recognised PDF files were not processed due to their lower quality, and text and paragraphs written in languages other than Bulgarian (mostly English) were removed.

3.4.3 Identification of sensitive personal data, biases, etc.

A pressing ethical issue that is essential in the development of large datasets with diverse sources is the identification and removal (e.g. masking) of personally identifiable information (Kober et al., 2023). However, the identification and removal of personally identifiable information is a difficult task due to the different types and forms as well as the inconsistent definitions, especially in various data protection laws (Song et al., 2025). A number of methods have been developed, including those based on machine learning techniques (Kulkarni and Cauvery, 2021; Shahriar et al., 2024), Transformers (Johnson et al., 2020; Shahriar et al., 2024) and rule-based identification (Jaikumar et al., 2023) as well as masking or tokenisation to remove personally identifiable information. In our approach, we experimented with the MAPA anonymisation package for Bulgarian[10] and with some naive rule-based methods to detect sentences of the document with potentially sensitive information and mark their number per document in the metadata.

The increasing development of LLMs has led to consideration of the biases inherent in them, resulting in the development of a range of techniques to measure and eliminate bias, particularly in relation to social issues. The main groups of techniques that address bias include: (a) the introduction of metrics to assess and identify bias in datasets; (b) techniques to reduce bias in the pre-processing, training and post-processing stages. Gallegos et al. (2024) summarises a wide range of current research focused on better understanding and preventing the propagation of bias in LLMs. Our goal is to score the documents in our dataset according to the percentage of potentially biased or abusive sentences and include this information in the metadata for further use, text filtering, etc. In this way, we can make a selection of documents for fine-tuning without sensitive and biased content, but we can also use the data for further research on bias. Currently, the classification of potentially biased sentences is being developed.

[10] https://mapa-project.eu/

3.5 Current structure of the IfGPT dataset

The current structure of the **IfGPT** dataset in terms of the source datasets of the text samples, the domain distribution and the size is shown in Table 1. The newly compiled dataset has a standardised representation of the metadata and text formats and was subjected to the data quality improvement procedures (see 3.4). The process of expanding **IfGPT** dataset with clean data is ongoing.

Source	# texts	# tokens	Licence
MARCELL	25K	45M	PD
CURLICAT	113K	35M	CC
BulNC Admin	17K	79M	PD
BulNC Wikipedia	89K	41M	CC/GNU
BulNC Subtitles	146K	27M	OPUS

Table 1: Current structure of IfGPT (August 2025). Licences: PD – public domain, CC – Creative Commons (various), GNU – GNU Free Documentation License, other open or restrictive licenses.

The metadata description of the texts within the **IfGPT** dataset is available to search and extract subsets.[11]

4 File format

Some of the documents in the **IfGPT** dataset are already available in vertical format, in CoNLL-U Plus format or in JSON format. The metadata is included in both the CoNLL-U Plus and JSON format, while in the vertical format the metadata is available in separate associated files. All documents are also saved in raw text format before being annotated and converted to either CoNLL-U Plus or JSON format.

The metadata descriptions are in the form of attribute-value pairs. For some categories, the values are predefined, e.g. for the media type, for others, e.g. the title of the document, any value is permitted.

The IfGPT dataset is provided in JSONL format for the LLM tasks, but the other available format versions can be requested if required.

[11] https://ifgpt.dcl.bas.bg/ifgpt-dataset/

5 Metadata categories

Metadata is essential to ensure efficient and effective selection of datasets for fine-tuning and RAG for specific domains and applications. Fine-tuning of LLMs is performed as a language-dependent task, focusing on a specific language, in our case Bulgarian, and further reducing the scope to a specific domain, task, etc. This requires the selection of a dataset with relevant data to ensure successful fine-tuning. On the other hand, metadata can be used not only for the selection of datasets suitable for RAG, but also for more effective methods of filtering information in RAG based on metadata (Bruni et al., 2025). Even though we emphasise the importance of metadata, we must point out that the empirical evaluation of the efficiency of metadata descriptions is beyond the scope of this study.

All four text collections (described in 3.1) have been supplied with metadata. The metadata of the Bulgarian National Corpus is aimed at searching and retrieving information for the needs of corpus and language research in general and therefore has a complex graph-based structure of related categories (Koeva et al., 2016). The metadata for the resource General News in Bulgarian is simply a categorisation into up to six most likely thematic domains (sports, politics, history, etc.). The metadata for the other two multilingual resources that also contain Bulgarian (MARCELL and CURLICAT) are synchronised between the different languages to form a single subset of categories. All four resources have overlapping metadata, and based on our task we have defined a set of metadata that is mandatory for each document (regardless of whether there are categories with a null value) and metadata that is optional. Optional metadata is metadata that is already assigned to the document but is not part of the mandatory metadata.

The following mandatory metadata is defined for the documents:

Identifier – unique identifier of the document in all collections, created with the language code `bg` as a prefix;

Licence – the conditions for use, i.e. CC BY-SA 4.0 licence;

PublicationDate – the date of original publication of the document (if available) in ISO 8601 format;

DocumentTitle – human-readable title (name) of the document;

Source – the name of the organisation that published the source document, i.e. journal, publisher, blog, website, etc.;

Medium – whether the document is text, audio, image or video;

Url – the original individual address where the document was retrieved from, if applicable;

Domain – classification of a specific thematic domain selected from a predefined list of 24 domains; up to six domains can be listed;

Keywords – extracted terms that specify the document; up to six keywords can be listed;

NumberWords – the total number of words in the document;

NumberSentences – the total number of sentences in the document;

NumberTokens – the total number of tokens in the document;

PersonallyIdentifiableInformation – the percentage of tokens in the total number of tokens in the document;

BiasedInformation – the percentage of tokens in the total number of tokens in the document.

The following metadata is optional for the documents:

Author – name(s) of the person(s) who created the text in the source document;

Style – the literary style of the text in the document, selected from a predefined list: Fantasy, Administrative, Legal, Journalism, etc.;

Type – specifies the type of the source document (e.g. book, chapter, essay, newspaper article, blog post, etc.);

Subdomain – a further classification of documents into narrower categories, e.g. scientific domains for the field of science or cultural domains for the field of culture; a subdomain is linked to a specific domain;

TranslatedDocument – whether the document was originally created in Bulgarian or whether it has been translated;

CollectionDate – the date of collection of the document in ISO 8601 format;

LicenseLink – the link to the licence on the source's website, if available;

NumberParagraph – the total number of paragraphs in the document;

TaskCategories - the applications (selected from a predefined list) for which the template was developed or is suitable, e.g. for question-answering.

Some of the metadata values are extracted automatically. The main techniques for automatically extracting metadata are: (a) metatextual techniques, which consist of extracting information from the HTML markup of the original files; and (b) textual techniques, which consist of text analysis and heuristics using a set of language resources. The following metadata values are automatically extracted from the HTML sources: Author, DocumentTitle, PublicationDate. The classification information includes the thematic domain of the texts, their genre and type as well as the results of the text analysis. In some cases, the source may contain classification labels according to an assumed domain and/or genre classification of the source, e.g. texts on a news website may be divided into editorials and articles of different domains – business, sports, etc.

Some metadata values are generated automatically. These are statistical information resulting from the processing of the text that includes the number of words, tokens, sentences, etc. Administrative metadata such as the document identifier, language code and source are also generated.

In order to improve the quality and quantity of the metadata used to describe each text entry, several procedures are defined. These procedures aim to identify contextually relevant descriptors to fill in missing values in the metadata. The reasons for incomplete data are manifold: in some cases, the data is not collected (by the users/authors), the website where the text is stored does not store certain types of data, it may be difficult or impossible to extract it from the online source, or it may be the result of data integration errors.

The task can be performed as a multi-class classification using heuristics, statistical methods or machine learning. It has been pointed out that traditional statistical methods for data imputation often do not provide an accurate and comprehensive description as they do not analyse the semantic context and relationships within the data (Jin et al., 2025). Mei et al. (2021) propose the use of a pretrained language model to assign metadata based on semantic features of the text and its description. Alyafeai et al. (2025) uses LLMs to automatically extract metadata from scientific articles by analysing context length and few-shot learning.

So far, we use more traditional methods for extracting metadata – statistical and rule-based, depending on the source of the document, the original format of the document (PDF, HTML, etc.) and the structure of the document itself. As we want to harmonise the metadata of existing datasets and new incoming texts, extending and standardising the metadata of available documents may require re-crawling the sources and repeating the text extraction process. We will upgrade the methods and tolls we use (Koeva et al., 2020) with the functionalities of applications like Trafilatura (Alyafeai et al., 2025), Maker,[12] etc.

6 Metadata management

Graph databases are designed to efficiently process large amounts of interconnected data. They can be scaled horizontally by adding more nodes to the database, while maintaining performance even for complex queries. The most commonly used graph databases are Neo4J, Microsoft Azure Cosmos DB, ArangoDB, TigerGraph and Amazon Neptune. Neo4J[13] is one of the most popular graph databases due to its high performance, support for the Cypher query language (Francis et al., 2018) and strong community support.

To effectively utilise the properties of a graph database when storing metadata, a schema is designed that captures the most important entities and their connections. The nodes of the metadata schema are defined as follows:

Document nodes with the properties: **Identifier, Title, Source, Domain, Author, Licence**, etc.;

Domain nodes with the properties **Name** and **Parent_category**;

Author nodes with the properties **Name** and optional details such as **Biography**;

Source nodes with the properties **Name** and **Url**;

Licence nodes with a single property **Type**.

The graph edges, which represent the relations between the nodes, are defined as follows:

Document-Domain of type BELONGS_TO;

Domain-Domain of type SUBCATEGORY_OF;

Document-License of type LICENSED_WITH;

Document-Author of type WRITTEN_BY;

Document-Source of type PUBLISHED_IN.

[12] https://github.com/datalab-to/marker
[13] https://neo4j.com/

```
CREATE (d:Document {id: "bg-bnc-2011040848215", title: "Ото
Варбуpг", author: "ChuispastonBot", source:
"bg.wikipedia.org", publication_date: "2011-04-08", domain:
"SCIENCE", subdomain: "BIOLOGY", license: "CC-BY-SA"})

CREATE (c1:Domain {name: "SCIENCE"})
CREATE (c2:Domain {name: "BIOLOGY"})

CREATE (a:Author {name: "ChuispastonBot"})
CREATE (s:Source {name: "bg.wikipedia.org", url:
"http://bg.wikipedia.org/"})
CREATE (l:License {type: "CC-BY-SA"})

// Create relationships
MATCH (d:Document {id: "bg-bnc-2011040848215"}), (c2:Domain
{name: "BIOLOGY"})
CREATE (d)-[:BELONGS_TO]->(c2)

MATCH (c2:Category {name: "BIOLOGY"}), (c1:Domain {name:
"SCIENCE"})
CREATE (c2)-[:SUBCATEGORY_OF]->(c1)

MATCH (d:Document {id: "bg-bnc-2011040848215"}), (a:Author
{name: "ChuispastonBot"})
CREATE (d)-[:WRITTEN_BY]->(a)

MATCH (d:Document {id: "bg-bnc-2011040848215"}), (s:Source
{name: "bg.wikipedia.org"})
CREATE (d)-[:PUBLISHED_IN]->(s)

MATCH (d:Document {id: "bg-bnc-2011040848215"}), (l:License
{type: "CC-BY-SA"})
CREATE (d)-[:LICENSED_UNDER]->(l)
```

Example 1: Processing a document with Cypher QL

Integrating the datasets into a vector database such as ChromaDB[14] can improve the efficiency of storing and querying vector representations of text data, which is critical for RAG technology. The conversion of text data into vector representations can be done using embeddings generated by specialised models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) or other transformer-based architectures. The purpose of these vectors is to capture semantic information about the text and use it for similarity searches.

To integrate ChromaDB into a graph database, the unique identifier is stored for each document or vector representation, depending on the granularity required for a particular task. An example workflow for processing Bulgarian texts with ChromaDB is presented below:

→ Vectorise the text using a selected embedding model. Save these vectors in ChromaDB with unique IDs.

→ Store metadata in the Neo4J graph database by creating nodes for sentences or documents and storing metadata such as Author, Source and Domain.

→ Also save the vector ID from ChromaDB as a property of the node.

→ Query similar documents using ChromaDB to find the closest vectors to a given query vector and retrieve the IDs of these vectors. Use this to query the graph database for the corresponding metadata.

[14] https://www.trychroma.com/

7 Conclusion

The most important results reported in this paper include the compilation of the **IfGPT** dataset for Bulgarian and the development of a metadata schema with graph-structured categories that enables efficient searching in the metadata. We also provide an online search interface in the metadata that enables the identification of smaller datasets tailored to specific domains and applications.

The metadata description of the **IfGPT** dataset contains a large number of categories that describe the text samples on different levels. Some of the most important metadata categories for the compilation of domain- and application-specific datasets are the following:

Domain information: A set of characteristics used to comprehensively describe the domain of the text was produced, including style, domain, subdomain. The source can also provide information about the domain, e.g. scientific journals in different domains or subsections of a news source.

Keywords: A schematic description of the content of the text sample can be created automatically based on the title, abstract (if available) or full text.

Sensitive personal data and biases: The parts containing sensitive personal data and biases are not removed or replaced by neutral data, but the percentage of such content in a document is calculated and can thus vary the strictness of the criteria for exclusion from certain datasets.

Using a graph database to store metadata offers several advantages over traditional relational databases or file-based systems. One of the main advantages is the ability to effectively model complex relationships between linguistic entities.

To summarise, a suitable dataset such as **IfGPT** – as large as possible, equipped with rich metadata for efficient search and retrieval of suitable documents, clearly defined tasks and thematic domains, and adequately managed with a graph database integrated with a database of embeddings – will enable fast and efficient fine-tuning of LLMs and Retrieval Augmented Generation.

Acknowledgment

The present study is carried out within the project Infrastructure for Fine-tuning Pre-trained Large Language Models, Grant Agreement No. ПВУ – 55 from 12.12.2024 /BG-RRP-2.017-0030-C01/.

Limitations

The main practical constraints involve the lack of extensive and diverse sources for collecting texts from specialised domains in Bulgarian. Additionally, specialised texts are often distributed in PDF format, which presents challenges for maintaining high text quality in the data.

For metadata, automatic collection may be inadequate, as online sources often provide limited information about the text. Conversely, manual metadata description is inefficient in terms of human effort and time. As high-quality metadata is important for correct dataset selection, some evaluation metrics for automatically assigned metadata, ensuring its completeness and consistency, need to be developed.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A Survey on Data Selection for Language Models. *Transactions on Machine Learning Research*.

Zaid Alyafeai, Maged S. Al-Shaibani, and Bernard Ghanem. 2025. MOLE: Metadata extraction and validation in scientific papers using llms. *arXiv e-prints*, page page arXiv: 2505.19800.

Davide Bruni, Marco Avvenuti, Nicola Tonellotto, and Maurizio Tesconi. 2025. AMAQA: A metadata-based qa dataset for rag systems. *arXiv e-prints*, page page arXiv: 2505.13557.

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.

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.

Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An Evolving Query Language for Property Graphs. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, page 1433–1445, New York, NY, USA. Association for Computing Machinery.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and Fairness in Large Language Models: A Survey. *Computational Linguistics*, 50(3):1097–1179.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling.

Jaikishan Jaikumar, Mohana, and Pavankumar Suresh. 2023. Privacy-Preserving Personal Identifiable Information (PII) Label Detection Using Machine Learning. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–5.

Can Jin, Tong Che, Hongwu Peng, Yiyuan Li, Dimitris N. Metaxas, and Marco Pavone. 2025. Learning from teaching regularization: generalizable correlations should be easy to imitate. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Alastair E.W. Johnson, Lucas Bulgarelli, and Tom J. Pollard. 2020. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *Proceedings of the ACM Conference on Health, Inference and Learning, Toronto, Ontario, Canada 2020*, pages 214–221.

Maria Kober, Jordan Samhi, Steven Arzt, Tegawendé F. Bissyandé, and Jacques Klein. 2023. Sensitive and personal data: What exactly are you talking about? In *MOBILESoft*, pages 70–74.

Svetla Koeva, Nikola Obreshkov, and Martin Yalamov. 2020. Natural language processing pipeline to annotate Bulgarian legislative documents. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6988–6994, Marseille, France. European Language Resources Association.

Svetla Koeva, Ivelina Stoyanova, Svetlozara Leseva, Rositsa Dekova, Tsvetana Dimitrova, and Ekaterina Tarpomanova. 2012. The Bulgarian National Corpus: Theory and Practice in Corpus Design. *Journal of Language Modelling*, (1):65–110.

Svetla Koeva, Ivelina Stoyanova, Maria Todorova, Svetlozara Leseva, and Tsvetana Dimitrova. 2016. Metadata extraction, representation and management within the Bulgarian National Corpus. In *Priceedings of the 4th Workshop on Challenges in the Management of Large Corpora*, pages 33–39. ELDA.

Poornima Kulkarni and N. K. Cauvery. 2021. Personally Identifiable Information PII Detection in the Unstructured Large Text Corpus using Natural Language Processing and Unsupervised Learning Technique. *International Journal of Advanced Computer Science and Applications*, 12(9).

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2020. *Mining of Massive Datasets*, 3rd edition. Cambridge University Press.

Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. 2024. Datasets for Large Language Models: A Comprehensive Survey. *arXiv e-prints*, page arXiv:2402.18041.

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 e-prints*, page arXiv:1907.11692.

Yuting Lu, Chao Sun, Yuchao Yan, Hegong Zhu, Dongdong Song, Qing Peng, Li Yu, Xiaozheng Wang, Jian Jiang, and Xiaolong Ye. 2024. A Comprehensive Survey of Datasets for Large Language Model Evaluation. In *2024 5th Information Communication Technologies Conference (ICTC)*, pages 330–336.

Alexandra Luccioni and Joseph Viviano. 2021. What's in the Box? An Analysis of Undesirable Content in the Common Crawl Corpus. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 182–189, Online. Association for Computational Linguistics.

Yinan Mei, Shaoxu Song, Chenguang Fang, Haifeng Yang, Jingyun Fang, and Jiang Long. 2021. Capturing Semantics for Imputation with Pre-trained Language Models. In *Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 2021*, pages 61–72.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling language models: Methods, analysis & insights from training gopher. *arXiv e-prints*, page arXiv:2112.11446.

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.

Md Hasan Shahriar, Anne V. D. M. Kayem, David Reich, and Christoph Meinel. 2024. Identifying personal identifiable information (PII) in unstructured text: A comparative study on transformers. In *Database and Expert Systems Applications: 35th International Conference, DEXA 2024, Naples, Italy, August 26–28, 2024, Proceedings, Part II*, page 174–181, Berlin, Heidelberg. Springer-Verlag.

Qiurong Song, Yanlai Wu, Rie Helene (Lindy) Hernandez, Yao Li, Yubo Kou, and Xinning Gui. 2025. Understanding Users' Perception of Personally Identifiable Information. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 26 April - 1 May 2025*. Association for Computing Machinery, New York.

Tamás Váradi, Svetla Koeva, Martin Yamalov, Marko Tadić, Bálint Sass, Bartłomiej Nitoń, Maciej Ogrodniczuk, Piotr Pęzik, Verginica Barbu Mititelu, Radu Ion, Elena Irimia, Maria Mitrofan, Vasile Păiș, Dan Tufiș, Radovan Garabík, Simon Krek, Andraz Repar, Matjaž Rihtar, and Janez Brank. 2020. The MARCELL Legislative Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3761–3768, Marseille, France. European Language Resources Association.

Tamás Váradi, Bence Nyéki, Svetla Koeva, Marko Tadić, Vanja Štefanec, Maciej Ogrodniczuk, Bartłomiej Nitoń, Piotr Pęzik, Verginica Barbu Mititelu, Elena Irimia, Maria Mitrofan, Dan Tufiș,

Radovan Garabík, Simon Krek, and Andraž Repar. 2022. Introducing the CURLICAT Corpora: Seven-language Domain Specific Annotated Corpora from Curated Sources. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 100–108, Marseille, France. European Language Resources Association.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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. *CoRR*, abs/2010.11934.

Modular Training of Deep Neural Networks for Text Classification in Guarani

José Luis Vázquez Noguera
Centro de Investigación
Universidad Americana
Asunción, Paraguay
jose.vazquez@ua.edu.py

Carlos U. Valdez
Facultad de Ciencias y Tecnología
Universidad Autónoma de Asunción
Asunción, Paraguay
cavaldez@uaa.edu.py

Julio César Mello-Román
Facultad Politécnica
Universidad Nacional de Asunción
San Lorenzo, Paraguay
juliomello@pol.una.py

Marvin M. Aguero
Facultad de Ciencias y Tecnología
Universidad Autónoma de Asunción
Asunción, Paraguay
marvin-aguero@outlook.com

José D. Colbes
Facultad Politécnica
Universidad Nacional de Asunción
San Lorenzo, Paraguay
jcolbes@pol.una.py

Sebastián A. Grillo
Facultad de Ciencias y Tecnología
Universidad Autónoma de Asunción
Asunción, Paraguay
sgrillo@uaa.edu.py

Abstract

We present a modular training approach for deep text classification in Guarani, where networks are split into sectors trained independently and later combined. This sector-wise backpropagation improves stability, reduces training time, and adapts to standard architectures like CNNs, LSTMs, and Transformers. Evaluated on three Guarani datasets—emotion, humor, and offensive language—our method outperforms traditional Bayesian-optimized training in both accuracy and efficiency.

1 Introduction

Natural language processing (NLP) for low-resource languages has gained attention due to the need for more inclusive technologies (Joshi et al., 2020). Guarani, an indigenous language spoken by over eight million people in Paraguay and neighboring countries, remains underrepresented in digital resources.It lacks open corpora, standard models, and suffers from frequent code-switching with Spanish (Estigarribia, 2016), which complicates data collection. These particularities of the Guarani language, coupled with the scarcity of labeled data and pretrained modules, make it challenging to train deep neural networks that generalize well to downstream tasks such as sentiment analysis, which are standard benchmark tasks (Mao et al., 2023) for high-resource languages like English.

Some efforts in low-resource NLP for Guarani have focused on corpus creation and benchmarking. Chiruzzo et al. (2020) expanded initial Guarani-Spanish sentence pairs into larger parallel collections, later unified and quality controlled as the `Jojajovai` corpus (Chiruzzo et al., 2022). Preliminary Guarani BERT (Devlin et al., 2019) variants (including continuous-pretrained and trained from scratch) have been trained on Wikipedia-derived texts containing only ∼800K tokens (Agüero-Torales et al., 2023), and Guarani was added to large multilingual initiatives such as 'No Language Left Behind' (NLLB Team et al., 2022) and `Google Translate` (Bapna et al., 2022). With regard to the text classification task, there are some works with diverse results, mainly for affective computing such as (i) (Agüero-Torales et al., 2023) explores various deep neural text classification techniques for multidimensional affective analysis; and (ii) sentiment analysis (Ríos et al., 2014), covering approaches that range from lexicon-based or traditional machine learning models (bag-of-words) to more sophisticated methods such as fine-tuning multilingual transformer models (Vaswani et al., 2017).

On the other hand, traditional text classification approaches in high-resource settings rely on end-to-end backpropagation over large corpora and big pretrained embeddings. When applied to Guarani, these methods tend to overfit quickly or fail to converge, since the number of tunable parameters far exceeds the available supervision. Recent work on low-resource NLP has mitigated these issues through transfer learning and cross-lingual embeddings (e.g. Schuster et al. (2019)), or adapting models trained in related languages or synthetic data (Lucas et al., 2024). However, these strategies remain monolithic: they update most network parameters at once, risking catastrophic forgetting of pretrained knowledge or uneven adaptation across layers (Kirkpatrick et al., 2017; Roy, 2024).

In parallel, modular and layer-wise training has been proposed in other domains (e.g. vision) to control the capacity of deep architectures (Tabrizi

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 76–81, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-008

et al., 2024). By isolating each layer (or 'sector') and optimizing its weights separately, these methods reduce the dimensionality of each learning step, reducing overfitting, and accelerating convergence (Belilovsky et al., 2020). However, to our knowledge, no prior work has applied a fully sector-wise backpropagation scheme to text classification in a truly low-resource language.

This work is based on the layer-wise loss assignments approach for layer-wise training (Belilovsky et al., 2020, 2019), which trains each layer using an auxiliary coupled model that can have several layers. Our approach decomposes a deep network into successive parameterized sectors, each trained as a shallow subnetwork on intermediate representations. We then recombine the trained sectors into a full model, preserving both pretrained knowledge and local adjustments. This sector-wise backpropagation delivers the following benefits:

- It constrains the number of parameters updated at each step, resulting in more stable training curves on small Guarani datasets.
- It preserves cross-sector knowledge transfer by propagating learned representations forward between stages.
- It consistently integrates with any architecture built from standard layers (e.g., convolutional (LeCun et al., 1989), Long Short-Term Memory (Hochreiter and Schmidhuber, 1997, LSTM) or transformers (Vaswani et al., 2017)), allowing the adaptation of existing models.

We validate our proposal on three Guarani corpora for affective computing (Agüero-Torales et al., 2023), namely: i) *gn-humor-detection*, ii) *gn-offensive-language-identification*, and iii) *gn-emotion-recognition*. In experiments, our sector-wise method outperforms conventional end-to-end training and standard baselines by significant margins. The remainder of this paper is structured as follows. Section 2 details our sector-wise optimization algorithm. Section 3 presents the experimental results, and Section 4 concludes our work.

2 Sector-wise Backpropagation

The modular optimization applied in this work is based on the concept of *sector*. A sector consists of a parameterized layer and all subsequent non-parameterized layers until the next parameterized layer. For example, in a network with architecture C_1–P_1–P_2–C_2–P_3–C_3 (where C_i are fully connected layers and P_i are pooling layers), the sectors would be:

$S_1 = C_1$–P_1–P_2, $S_2 = C_2$–P_3, $S_3 = C_3$.

Given a network D and a training set X, for each epoch, the method proceeds in three main steps:

1. For each sector S_i (excluding the last), construct a shallow network N_i composed of S_i and an output layer identical to that of D.
2. Train each N_i using transformed instances $f_{i-1}(x)$, where $f_0(x) = x$, and we define $f_i(x)$ as the output of N_i with its output layer removed.
3. Rebuild D by stacking the trained sectors and removing the auxiliary output layers, except for the final one.

In the earlier example, the auxiliary networks created would be:

$N_1 = C_1$–P_1–P_2–C_3', $N_2 = C_2$–P_3–C_3,

where C_3' replicates C_3. N_1 is trained on $x \in X$, and N_2 is trained on transformed outputs $f_1(x)$. Algorithm 1 formalises the proposal.

Algorithm 1 Sector-wise Local Backpropagation and Network Reconstruction

Initialize: Architecture D, sectors $S_1, \ldots, S_n$, null network R_0
while stop condition not met **do**
  Sector Backpropagation
  for $i = 1$ to $n - 1$ **do**
    Create N_i by adding a layer similar to the last layer of D on top of sector S_i
    Train N_i for one epoch using instances $f_{i-1}(x)$ for $x \in X$, with the same label as x
    Compute $f_i(x)$ for each $x \in X$ by evaluating the penultimate layer output of N_i
  end for
  Network Reconstruction
  $R_0 \leftarrow \varnothing$
  for $i = 1$ to $n - 2$ **do**
    Extract S_i from trained N_i preserving learned parameters
    Connect S_i to R_{i-1} according to D, forming R_i
  end for
  Connect R_{n-2} to N_{n-1} according to D, forming R_{i-1}
end while
return R_{i-1}

3 Results

Experiments were conducted on three datasets (over their train-dev-test splits): *gn-humor-detection* (fun and no-fun classes), *gn-offensive-language-identification* (offensive and no-offensive

classes), and *gn-emotion-recognition* (happy, angry, sad and other classes) (Agüero-Torales et al., 2023); using 10-fold cross-validation. Three model architectures were tested on each dataset: a **1D convolutional network** (Omernick and Chollet, 2019; Waibel et al., 1989), a **transformer-based model** (Nandan, 2020; Vaswani et al., 2017), and a **bidirectional LSTM** (Chollet, 2020; Schuster and Paliwal, 1997).

Each model was trained under three configurations: i) Standard backpropagation with fixed hyperparameters, ii) Backpropagation with Bayesian hyperparameter optimization and iii) Sector-based backpropagation (the proposed method).

For configurations 1 and 3, training was performed using a *batch size of 32*, *learning rate of 0.001*, the *Adam optimizer*, and *sparse categorical cross-entropy* loss. For configuration 2, Bayesian optimization was applied with the following domains: i) optimizer $\in$ {`adam`, `rmsprop`, `sgd`}, ii) learning rate $\in$ (1e-5, 1e-1) with a log-uniform distribution and iii) Batch size $\in$ [16, 128].

Table 1: Average accuracy on the gn-humour-detection dataset as the number of training epochs increases. Model 1 is a 1D ConvNet, model 2 is a Transformer, and model 3 is a Bidirectional LSTM.

Mod.	Epoch	Simp.	Bayes.	Prop.
1	2	**71.27**	**71.27**	70.27
1	4	69.92	70.38	**71.46**
1	6	70.19	69.95	**71.27**
1	8	65.58	68.99	**71.22**
1	10	66.12	68.78	**71.76**
2	2	71.27	71.27	**73.28**
2	4	71.27	71.25	**73.52**
2	6	71.82	71.27	**74.09**
2	8	62.33	71.27	**73.55**
2	10	59.62	71.27	**73.98**
3	2	64.54	64.66	**68.92**
3	4	64.85	66.02	**69.16**
3	6	58.27	65.39	**70.46**
3	8	63.04	65.18	**69.40**
3	10	64.23	65.15	**70.54**

Table 1 presents the corresponding results for the *gn-humor-detection* dataset. They are grouped according to the models (first column), considering different epochs (second column), followed by the average accuracy for each configuration. Considering each model, the transformer-based one achieved the highest accuracy among the others. More interestingly, our proposal obtained a better performance in nearly all cases (except for model 1 with 2 epochs). In terms of accuracy, the best configuration recorded (74.09%) is the transformer-based architecture when trained with the proposal for 6 epochs. Moreover, for the first two configurations, the average accuracy generally decreases slightly as the number of epochs increases. This behaviour does not appear in our proposal.

Table 2: Average accuracy on the gn-offensive-language-identification dataset as the number of training epochs increases. Model 1 is a 1D ConvNet, model 2 is a Transformer, and model 3 is a Bidirectional LSTM.

Mod.	Epoch	Simp.	Bayes.	Prop.
1	2	83.87	84.22	**85.12**
1	4	80.41	78.96	**85.02**
1	6	82.72	82.35	**86.31**
1	8	81.11	81.66	**87.00**
1	10	70.28	81.27	**86.94**
2	2	83.87	84.15	**89.84**
2	4	84.10	82.42	**89.59**
2	6	83.40	83.96	**89.77**
2	8	82.40	85.97	**90.09**
2	10	82.32	86.89	**89.95**
3	2	86.87	**88.32**	87.72
3	4	70.74	87.81	**88.41**
3	6	85.71	88.04	**88.20**
3	8	86.25	88.00	**88.02**
3	10	87.48	88.44	**89.51**

For the *gn-offensive-language-identification* dataset, the results are presented in Table 2. In general, the average accuracies are higher than in the first dataset (>80% in almost all combinations). As before, our proposal achieved better performance in nearly all cases (except for model 3 with 2 epochs), and by a significantly larger margin for the 1D ConvNet and transformer-based models. In this dataset, the best configuration recorded (90.09%) is the transformer-based architecture when trained with the proposal for eight epochs.

Table 3: Average accuracy on the gn-emotion-recognition dataset as the number of training epochs increases. Model 1 is a 1D ConvNet, model 2 is a Transformer, and model 3 is a Bidirectional LSTM.

Mod.	Epoch	Simp.	Bayes.	Prop.
1	2	37.78	48.92	**55.43**
1	4	41.27	49.30	**55.05**
1	6	49.84	50.98	**55.43**
1	8	50.16	50.06	**55.97**
1	10	45.71	49.21	**56.10**
2	2	37.78	36.00	**48.29**
2	4	45.08	39.87	**51.71**
2	6	47.30	47.62	**55.27**
2	8	47.62	47.84	**55.87**
2	10	51.75	53.08	**56.03**
3	2	45.71	52.06	**55.17**
3	4	50.19	52.48	**56.92**
3	6	51.83	52.86	**58.35**
3	8	52.70	51.30	**57.75**
3	10	52.06	53.33	**57.84**

Table 3 shows the results for the last dataset,

gn-emotion-recognition. In this case, the accuracy values are substantially lower than those presented in Tables 1 and 2; therefore, it is the most challenging dataset. Another interesting point is that, for all epoch values, the results for the bidirectional LSTM-based models are superior to those of the other models. As with the previous datasets, our proposal consistently outperforms the other configurations. The best accuracy (58.35%) corresponds to the bidirectional LSTM model with the proposal, trained for six epochs.

Figures 1, 2, and 3 illustrate the average execution times observed across models. The results suggest that execution time is more strongly influenced by the network architecture than by the dataset itself. For the 1D ConvNet and bidirectional LSTM architectures, sector-based training achieved execution speeds approximately two and three times faster, respectively, compared to standard backpropagation. The proposal yielded speedups of up to 32× in relation to traditional backpropagation with Bayesian optimization. In the case of the Transformer architecture, sector-based training incurred an execution time up to 10% longer than traditional training; however, with Bayesian optimization, it demonstrated a 12× improvement in efficiency.

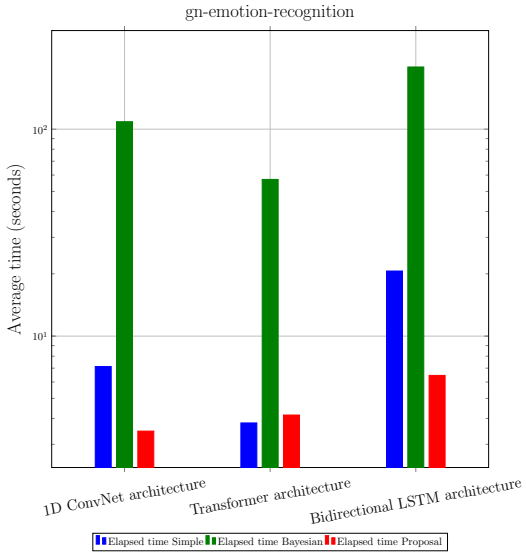


Figure 1: Average execution time for the gn-emotion-recognition dataset.

4 Conclusion

The experiments as a whole showed three notable advantages of sector training over traditional methods for text classification in Guarani using deep architectures. Firstly, for each dataset and algorithm, the highest average accuracy was always achieved by sector training during some epoch. This advantage ranged from less than 1% to almost 6% compared to the best value achieved by traditional methods. Second, the average accuracy is more stable for sector training, which does not show significant declines in later epochs, as can happen with traditional methods. Finally, the greatest advantage identified is the efficiency in execution time of sector training, which was not always lower than traditional simple backpropagation, but was nevertheless 12 to 32 times less costly than traditional backpropagation with Bayesian optimisation and with superior accuracy. This is noteworthy because traditional backpropagation with Bayesian optimisation represents the best traditional configuration in terms of average accuracy.

As future work, we plan to evaluate the approach on multi-class classification tasks with alternative loss functions, extend experiments to more tasks and languages, and analyze its scalability with different sector sizes and smaller datasets.

Limitations

The evaluation was restricted to three small Guarani affective computing datasets, which may limit generalization to other tasks or languages. Moreover, the scalability of sector-wise backpropagation to larger architectures and broader benchmarks remains to be explored.

Disclaimer

During the preparation of this work the authors used generative tools in order to fix misspellings and improve writing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Code Availability

The code for reproducing the experiments presented in this paper is publicly accessible at `https://gitlab.com/pinv01-401/dloptimizer`.

Acknowledgments

This work was supported by the CONACYT, Paraguay, under Grant PINV01-401.

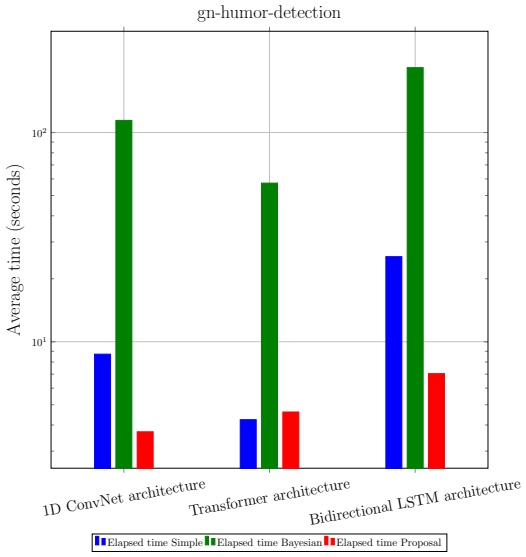


Figure 2: Average execution time for the gn-humor-detection dataset.

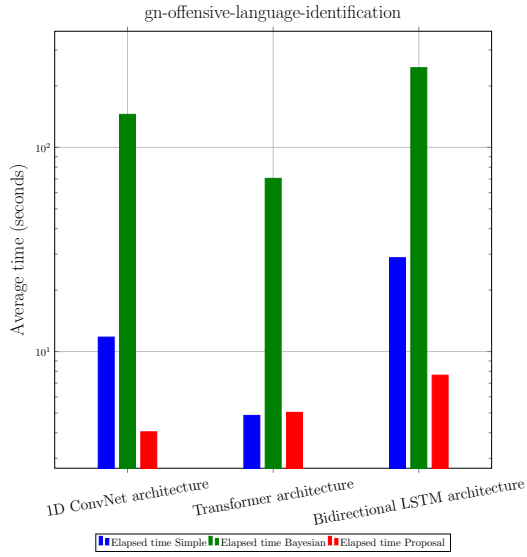


Figure 3: Average execution time for the gn-offensive-language-identification dataset.

References

Marvin M Agüero-Torales, Antonio G López-Herrera, and David Vilares. 2023. Multidimensional affective analysis for low-resource languages: A use case with guarani-spanish code-switching language. *Cognitive Computation*, 15(4):1391–1406.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, Theresa Breiner, Vera Axelrod, Jason Riesa, Yuan Cao, Mia Xu Chen, Klaus Macherey, Maxim Krikun, Pidong Wang, Alexander Gutkin, Apurva Shah, Yanping Huang, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2022. Building machine translation systems for the next thousand languages.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. 2019. Greedy layerwise learning can scale to imagenet. In *International conference on machine learning*, pages 583–593. PMLR.

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. 2020. Decoupled greedy learning of cnns. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 748–758. PMLR.

Luis Chiruzzo, Pedro Amarilla, Adolfo Ríos, and Gustavo Giménez Lugo. 2020. Development of a Guarani - Spanish parallel corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2629–2633, Marseille, France. European Language Resources Association.

Luis Chiruzzo, Santiago Góngora, Aldo Alvarez, Gustavo Giménez-Lugo, Marvin Agüero-Torales, and Yliana Rodríguez. 2022. Jojajovai: A parallel Guarani-Spanish corpus for MT benchmarking. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2098–2107, Marseille, France. European Language Resources Association.

François Chollet. 2020. Bidirectional lstm on imdb. `https://keras.io/examples/nlp/bidirectional_lstm_imdb/`.

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.

Bruno Estigarribia. 2016. Guaraní aquí, jopara allá. reflexiones sobre la (socio)lingüística paraguaya, written by penner, hedy. *Journal of Language Contact*, 9(2):397 – 403.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell.

2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551.

Agustín Lucas, Alexis Baladón, Victoria Pardiñas, Marvin Agüero-Torales, Santiago Góngora, and Luis Chiruzzo. 2024. Grammar-based data augmentation for low-resource languages: The case of Guarani-Spanish neural machine translation. 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 6385–6397, Mexico City, Mexico. Association for Computational Linguistics.

Rui Mao, Qian Liu, Kai He, Wei Li, and Erik Cambria. 2023. The biases of pre-trained language models: An empirical study on prompt-based sentiment analysis and emotion detection. *IEEE Trans. Affect. Comput.*, 14(3):1743–1753.

Apoorv Nandan. 2020. Text classification with transformer. `https://keras.io/examples/nlp/text_classification_with_transformer/`.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Mark Omernick and François Chollet. 2019. Text classification from scratch. `https://keras.io/examples/nlp/text_classification_from_scratch/`.

Kaushik Roy. 2024. *Lifelong Learning with Neural Network*. Ph.D. thesis, MONASH University.

Adolfo A. Ríos, Pedro J. Amarilla, and Gustavo A. Giménez Lugo. 2014. Sentiment categorization on a creole language with lexicon-based and machine learning techniques. In *2014 Brazilian Conference on Intelligent Systems*, pages 37–43.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. 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 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

Melika Sadeghi Tabrizi, Ali Karimi, Ahmad Kalhor, Babak N Araabi, and Mona Ahmadian. 2024. Layer-wise learning of cnns by self-tuning learning rate and early stopping at each layer. In *35th British Machine Vision Conference 2024, BMVC 2024, Glasgow, UK, November 25-28, 2024*. BMVA.

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.

Alex Waibel, Manfred Hanazawa, Gregory Hinton, Kevin Shikano, and Kevin J. Lang. 1989. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339.

Roman Urdu as a Low-Resource Language: Building the First IR Dataset and Baseline

Umer Butt[1,2,3] **Stalin Varanasi**[1,2] **Günter Neumann**[1,2]
[1]Saarland Informatics Campus, D3.2, Saarland University, Germany
[2] German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[3] Sequire technology GmbH, Saarbrücken, Germany
umer.butt@sequire.de, stalin.varanasi@dfki.de, guenter.neumann@dfki.de

Abstract

The field of Information Retrieval (IR) increasingly recognizes the importance of inclusivity, yet addressing the needs of low-resource languages, especially those with informal variants, remains a significant challenge. This paper addresses a critical gap in effective IR systems for Roman Urdu, a romanized version of Urdu i.e a language with millions of speakers, widely used in digital communication yet severely underrepresented in research and tooling. Roman Urdu presents unique complexities due to its informality, lack of standardized spelling conventions, and frequent code-switching with English. Crucially, prior to this work, there was a complete absence of any Roman Urdu IR dataset or dedicated retrieval work. To address this critical gap, we present the first-ever large-scale IR MS-marco translated dataset specifically for Roman Urdu, created through a multi-hop pipeline involving English-to-Urdu translation followed by Urdu-to-Roman Urdu transliteration. Using this novel dataset, we train and evaluate a multilingual retrieval model, achieving substantial improvements over traditional lexical retrieval baselines (**MRR@10**: **0.19** vs. 0.08; **Recall@10**: **0.332** vs. 0.169). This work lays foundational benchmarks and methodologies for Roman Urdu IR especially using the transformer based models, significantly contributing to inclusive information access and setting the stage for future research in informal, Romanized, and low-resource languages.

1 Introduction

Advancements in Information Retrieval (IR) have predominantly served high-resource languages, largely due to the availability of extensive training data and well-optimized models. As a result, informal and low-resource languages remain largely excluded from the benefits of modern IR systems.

Urdu is spoken by over 70 million people in South Asia and remains an important medium for written and verbal communication, especially in Pakistan and parts of India. Despite its widespread use, Urdu is underrepresented in digital language technologies due to challenges such as its Perso-Arabic script, right-to-left writing direction, and complex morphology, issues that are less severe in high-resource languages like Arabic or Chinese due to better tooling and research support.

Alongside standard Urdu, Roman Urdu (Urdu written in the Latin script) has become the dominant form of informal communication on platforms like Instagram, WhatsApp, and social media. Its popularity stems from practical constraints, such as the lack of easy-to-use Urdu keyboards and familiarity with Latin characters. (Safdar et al., 2020) However, Roman Urdu poses its own set of challenges, including inconsistent spelling, informal grammar, and frequent code-switching, making it especially difficult for information retrieval (IR) systems.

A key barrier in developing effective IR systems for both Urdu and Roman Urdu is the lack of large-scale, labeled datasets. Manual creation is often impractical, and while machine translation offers a scalable alternative, it can introduce semantic drift or misalignment. Recent work has begun addressing these issues for Urdu, but Roman Urdu remains largely overlooked. This work addresses the gap by constructing the first large-scale Roman Urdu IR benchmark. Our approach builds upon prior efforts in multilingual IR and transliteration. Following the methodology introduced in multilingual mMARCO (Nguyen et al., 2016a), we begin by translating the English MS MARCO dataset into Urdu as described by (Butt et al., 2025a), using a state-of-the-art translation model IndicTrans2 (Ramesh et al., 2022). To convert this Urdu data into Roman Urdu, we leverage a high-accuracy transliteration model as proposed in (Butt et al., 2025b) to outperform traditional approaches us-

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 82–87, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-009

ing transformer-based architectures and masked language modeling. Note that we go through this hopping process because there does not exist any open access model for direct translation of English to Roman-Urdu.

Our main contributions are:

- **Construction of the First Roman Urdu IR Dataset:** We generate a large-scale Roman Urdu version of MS MARCO via a multi-step translation and transliteration pipeline, maintaining semantic alignment with the original data.
- **Development of a Roman Urdu IR Model:** We fine-tune a multilingual IR model on the new dataset and demonstrate that it significantly outperforms the baseline, which struggles with informal and inconsistent spellings in Roman Urdu.
- **Scalable and Reusable Methodology:** Our approach provides a practical framework that can be adapted for other low-resource or Romanized scripts facing similar linguistic challenges.
- **Public Release of Resources:** To support future research, we make our Roman Urdu dataset, fine-tuned model, and code publicly available on Hugging Face and GitHub.[1] [2] [3]

2 Background on Romanization & Roman Urdu

Many multilingual communities, particularly across South Asia and Africa, use the Latin script (i.e., English alphabet) to write their native languages, a process known as romanization. This practice emerged from early limitations in computing and mobile technologies, where keyboards and software lacked support for non-Latin scripts. As a result, speakers of languages such as Urdu, Hindi, Bengali, and Arabic began using Roman characters to represent their native words. The trend was further reinforced by the global rise of the internet, where English remains dominant, making Romanized writing a convenient and accessible alternative for digital communication.

[1] https://huggingface.co/Mavkif/roman-urdu-mt5-mmarco

[2] https://huggingface.co/datasets/Mavkif/roman-urdu-msmarco-dataset

[3] https://github.com/UmerTariq1/MS-Marco-Translation-and-IR

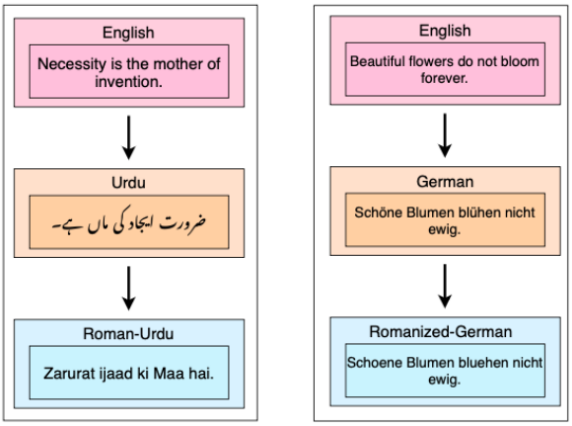


Figure 1: Examples of Romanization in English → Urdu → Roman-Urdu and English → German → Romanized-German.

Among these, Roman Urdu has seen especially widespread use across social media, e-commerce, online news, and informal messaging. Unlike standardized Latin-based scripts, however, Roman Urdu lacks any formal spelling conventions, leading to highly inconsistent, user-dependent, and phonetic spellings. The same word may appear in multiple forms based on how a speaker hears or pronounces it.

This lack of orthographic standardization poses significant challenges for NLP and IR models, which must account for noisy spelling, informal grammar, and frequent code-switching with English. As a result, building robust Roman Urdu datasets and retrieval models requires not just large-scale training data, but approaches that can handle spelling variation and contextual ambiguity in a low-resource setting. An example of a sentence in English, Urdu and Roman-Urdu and an an example of a sentence in English, Germany and Hypothethical Romanized-German language (to show example of how it would look) is given in 1

3 Related Work

Roman Urdu, despite being widely used online, has remained almost completely absent from retrieval research. Prior efforts have focused mostly on sentiment classification or dictionary creation (Smat26, 2023; Zahid et al., 2020), with no standardized datasets or IR models available for this variant. The lack of relevance-labeled resources and the informal nature of the script, non-standard spelling, code-switching with English, and inconsistent grammar, pose serious challenges for build-

ing retrieval models. This work is the first to directly address these challenges at scale. (Safdar et al., 2020)

In contrast, Urdu IR has recently seen progress through translation-based methods. (Butt et al., 2025a) translated the MS MARCO dataset (Nguyen et al., 2016b) into Urdu using IndicTrans2, producing over 8.8 million passages and 500,000+ queries. This dataset enabled fine-tuning of a multilingual mT5 reranker (Xue et al., 2021) and showed substantial performance gains over zero-shot and BM25 (Robertson and Walker, 1994) baselines. IndicTrans2 was chosen for its strong performance over Google Translate and OPUS-MT (Tiedemann, 2012), achieving a chrF++ score of 68.2.

Transliteration has also been studied in Roman Urdu, mainly using the Roman-Urdu-Parl corpus (Alam and Hussain, 2022). Early models used RNNs (Elman, 1990), but recent work by (Butt et al., 2025a) introduced a transformer-based transliteration model with MLM pretraining, significantly improving cross-domain robustness. That model forms the core of the Urdu→Roman Urdu step in our multi-hop pipeline.

Although multilingual IR research has produced general-purpose models like mBERT (Devlin et al., 2018), XLM-R (Conneau, 2019), LaBSE (Feng et al., 2020), and mT5 (Xue et al., 2021), these models struggle on underrepresented scripts like Roman Urdu due to minimal pretraining exposure. Efforts like mMARCO translated MS MARCO into 13 languages (Nguyen et al., 2016a) but excluded Urdu and Roman Urdu. Other multilingual IR benchmarks (e.g., MIRACL (Yu et al., 2021), Mr.TyDi (Clark et al., 2020)) offer limited or no coverage of these languages.

Our work bridges this gap by combining translation and transliteration to construct a Roman Urdu version of MS MARCO and training the first neural IR model for this script. It serves as a foundational benchmark for retrieval in Romanized low-resource languages.

4 Experimental Setup

4.1 Dataset Creation

We create our Roman Urdu IR dataset via a multi-hop translation process starting from the English MS MARCO dataset. Initially, we translate MS MARCO passages and queries into Urdu using the IndicTrans2 translation model, chosen for its strong performance on South Asian languages as previously demonstrated by (Butt et al., 2025a). The Urdu dataset comprises over 8.8 million passages and 500,000+ queries, serving as a reliable intermediate step.

English	Urdu (اردو) Translation	Roman-Urdu Transliteration
Query: what fruit is native to australia	**Query:** آسٹریلیا کا کون سا پھل مقامی ہے	**Query:** Australia ka kon sa phal muqami hai
Relevant Passage : Passiflora herbertiana. A rare passion fruit native to Australia. Fruits are green-skinned, white fleshed, with an unknown edible rating. Some sources list the fruit as edible, sweet and tasty.	**Relevant Passage:** پاسفلاورا ہربرٹیانا: آسٹریلیا کا ایک نایاب پیشن فروٹ۔ پھل سبز جلد والے سفیدگوشت والے ہوتے ہیں جن کی خوردنی درجہ بندی نامعلوم ہوتی ہے۔ کچھ ذرائع اس پھل کو خوردنی میٹھا اور لذیذ کے طور پر درج کرتے ہیں	**Relevant Passage :** pasflora harbrtiana. Australia ka aik nayab passion fruit. phal sabz jald walay safaid gosht walay hote hain jin ki khordani darja bandi namaloom hoti hen. kuch zaraye is phal ko khordani meetha aur lazeez ke taur par darj karte hain.
Non-Relevant Passage : The kola nut is the fruit of the kola tree, a genus (Cola) of trees that are native to the tropical rainforests of Africa.	**Non-Relevant Passage :** کولا نٹ کولا درخت کا پھل ہے جو درختوں کی ایک جینس کولا ہے جو افریقہ کے اشنکٹبندیی برساتی جنگلات سے تعلق رکھتی	**Non-Relevant Passage :** cola nutt cola darakht ka phal hai jo darakhton ki ek jeans cola hai jo Africa ke ashnktabndi barsati janglaat se taalluq rakhti ha

Figure 2: Example query and passage pair in English, Urdu (translated), and Roman Urdu (transliterated).

Next, we transliterate this whole Urdu dataset into Roman Urdu using a previously developed transliteration model (Butt et al., 2025b), which employs a transformer-based architecture (m2m100 (Fan et al., 2021)) fine-tuned on the Roman-Urdu-Parl corpus and augmented with Masked Language Modeling (MLM) pretraining. This step ensures robust handling of spelling variations common in Roman Urdu. This results in the Roman-Urdu version of the whole publically available English MS-Marco passage ranking dataset.

4.2 Potential Issues With Translated/Transliterated Dataset

Although machine translation provides a scalable solution, it can also lead to semantic drift and context loss. These issues are especially pronounced in multi-hop pipelines like ours (English → Urdu → Roman Urdu), where small inconsistencies can compound and negatively affect retrieval performance. Despite these challenges, the resulting dataset is the first large-scale Roman Urdu resource for information retrieval, making it a valuable foundation for future work.

An illustrative example of this semantic misalignment is shown in Figure 2.

4.3 Retrieval Model

We adopt a two-stage retrieval pipeline. First, a BM25 index serves as the base retriever, returning the top k=1000 candidates per query to ensure high recall. These candidates are then re-ranked using a multilingual reranker based on the mT5 architecture, which has been shown effective in multilingual IR tasks such as mMARCO.

Following the same approach as previously shown in (Butt et al., 2025a), we fine-tuned the

mMARCO model on the whole Roman Urdu dataset. While mT5 is pretrained on a diverse set of languages, it does not include Roman Urdu, making fine-tuning necessary to adapt to the script's informal and non-standard characteristics.

We frame retrieval as a binary relevance classification task in a sequence-to-sequence setup. For each query–passage pair, the model is trained to generate "yes" for relevant passages and "no" otherwise. At inference time, we compute a softmax over the generated tokens to obtain relevance scores for reranking.

The training configuration mirrors that of the mmarco model: a learning rate of 0.001, dropout of 0.1, and an effective batch size of 128 (batch size 32 with gradient accumulation over 4 steps). This consistent setup enables meaningful comparison between retrieval performance in Urdu and Roman Urdu, isolating the effects of linguistic representation.

4.4 Evaluation

We evaluate retrieval performance using standard IR metrics that reflect different aspects of effectiveness. We report **MRR@10**, **Recall@10**, **MAP@10**, **NDCG@10**, and **Precision@10** to provide a comprehensive view of overall ranking quality and the system's ability to surface relevant results.

Zero-shot multilingual models were not used as they fail to comprehend Roman Urdu's informal structure and inconsistent spelling. Since no prior baselines for Roman Urdu IR exist, we use **BM25** as the only meaningful comparison. Our significantly outperforming reranker demonstrates the effectiveness of the transliteration pipeline and model fine-tuning.

5 Results Discussion

We present the performance of our Roman Urdu IR model in Table 1, comparing our fine-tuned multilingual reranker against the BM25 baseline.

The reranker consistently outperforms BM25 across all metrics, achieving an MRR@10 of 0.1903 and Recall@10 of 0.3326, which is more than double the baseline values. Significant gains are also observed in MAP, NDCG, and Precision, indicating improvements in both early ranking and overall retrieval quality. This improvement is particularly notable given the noisy and inconsistent nature of Roman Urdu, which poses a challenge for lexical methods like BM25 that rely on exact token overlap. In contrast, the reranker benefits from contextual modeling and cross-lingual knowledge.

Compared to earlier results in Urdu IR (Butt et al., 2025a), the Roman Urdu model performs slightly lower (e.g., Urdu MRR@10: 0.248), which is expected due to the added noise introduced during transliteration. Nonetheless, the performance remains strong considering the informal nature of the script and lack of standardization.

These results validate our multi-hop pipeline and establish both the dataset and model as practical baselines for future research on retrieval in Romanized, informal, and low-resource languages.

Metric	BM25	Our Fine-tuned Reranker
MAP@10	0.0502	0.1262
MRR@10	0.0846	0.1903
NDCG@10	0.1218	0.2572
Precision@10	0.0177	0.0347
Recall@10	0.1699	0.3326

Table 1: Retrieval performance comparison on Roman Urdu MS MARCO (6980 queries).

6 Future Work and Conclusion

This paper presented the first large-scale Roman Urdu Information Retrieval dataset and benchmark, showing that fine-tuning a multilingual reranker substantially outperforms traditional methods. This approach effectively addresses the informal spelling and lack of standardization in Roman Urdu.

Future work could focus on reducing error propagation in the data pipeline, improving the transliteration model with more diverse data, and exploring subword or phonetic representations to better handle spelling variations. Our pipeline could also be extended to other Romanized scripts like Arabizi or Roman Hindi, broadening its application and fostering digital inclusion. This work provides a solid foundation and valuable resources for future research in Roman Urdu information retrieval

Limitations

While our work establishes the first large-scale Urdu and Roman Urdu resources for information retrieval, several limitations should be noted. First, the translation and transliteration pipeline introduces potential sources of semantic drift and context loss. These effects are particularly pronounced in multi-hop translation (English → Urdu → Ro-

man Urdu), where small inconsistencies can compound and reduce retrieval accuracy.

Second, our evaluation is limited to the MS MARCO-derived dataset. Although this provides a strong and widely used benchmark, it does not fully capture the diversity of information needs or linguistic phenomena in real-world Urdu and Roman Urdu usage.

Finally, due to time and resource constraints, we focused on establishing reliable baselines rather than exploring advanced modeling techniques or large-scale hyperparameter tuning. We view this work as a foundation for future improvements, such as expanding coverage to other domains, experimenting with alternative translation models, and refining retrieval strategies.

Broader Impact Statement

This work aims to improve digital information access for speakers of Roman Urdu, an informal and widely used script that has been historically ignored in language technology research. By providing the first large-scale dataset and retrieval model for Roman Urdu, our work contributes to a more inclusive digital ecosystem, enabling better access to search and knowledge for communities that rely on non-standardized, Romanized scripts.

Given the widespread use of Roman Urdu in South Asia, especially among younger and less formally educated populations, this research could help bridge digital inequality and support more equitable participation in online information spaces. Furthermore, our open-source release of models and datasets encourages transparency and reuse for similar languages and regions.

However, we also acknowledge that increased access to search and retrieval tools in informal scripts may be leveraged in unintended ways, such as misinformation or targeted advertising. Mitigating these risks requires responsible deployment and careful contextualization of the technology. We encourage future researchers and practitioners to work in collaboration with local communities to ensure that such tools are developed and used ethically.

Acknowledgement

We gratefully acknowledge the German Research Center for Artificial Intelligence (DFKI) for providing hardware and a supportive research environment. This work was also supported by the German Federal Ministry of Education and Research (BMBF) as part of the TRAILS project (01IW24005).

References

Mehreen Alam and Sibt Ul Hussain. 2022. Roman-urdu-parl: Roman-urdu and urdu parallel corpus for urdu language understanding. *Transactions on Asian and Low-Resource Language Information Processing*, 21(1):1–20.

Umer Butt, Stalin Veranasi, and Günter Neumann. 2025a. Enabling low-resource language retrieval: Establishing baselines for urdu ms marco. In *European Conference on Information Retrieval*, pages 282–289. Springer.

Umer Butt, Stalin Veranasi, and Günter Neumann. 2025b. Low-resource transliteration for roman-urdu and urdu using transformer-based models. *arXiv preprint arXiv:2503.21530*.

Jonathan H Clark, Eunsol Pfeiffer, Tom Kwiatkowski, Michael Collins, Kristina Toutanova, Patrick Lewis, Aishwarya Joshi, Pradeep Rajpurkar, and Luke Zettlemoyer. 2020. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

A Conneau. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016a. Ms marco: A human generated machine reading comprehension dataset. In *Proceedings of the 2016 workshop on machine reading for question answering*, pages 180–186.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng.

2016b. Ms marco: A human-generated machine reading comprehension dataset.

Anoop K Ramesh, Deepak Raj, Dayal Tang, et al. 2022. Indictrans2: An improved neural translation model for indic languages. *arXiv preprint arXiv:2205.13431*.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. *Citeseer*.

Zanab Safdar, Ruqia Safdar Bajwa, Shafiq Hussain, Haslinda Binti Abdullah, Kalsoom Safdar, and Umar Draz. 2020. The role of roman urdu in multilingual information retrieval: A regional study. *The Journal of Academic Librarianship*, 46(6):102258.

Smat26. 2023. Roman urdu sentiment dataset. https://www.kaggle.com/datasets/smat26/roman-urdu-dataset.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

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. *arXiv preprint arXiv:2010.11934*.

Guangwei Yu, Jing Liu, Jialu Tang, Zujie Li, Shuming Bi, Yubin Pan, Peiran Huang, Bolin He, Jianhua Zhou, Xiao Zhang, et al. 2021. Miracl: Multimodal retrieval augmented with contrastive in-batch negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1481–1491.

Rabail Zahid, Muhammad Owais Idrees, Hasan Mujtaba, and Mirza Omer Beg. 2020. Roman urdu reviews dataset for aspect based opinion mining. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, pages 138–143.

The Brittle Compass: Navigating LLM Prompt Sensitivity in Slovak Migration Media Discourse

Samuel Harvan
Central European University,
Kempelen Institute
of Intelligent Technologies
samuel.harvan@intern.kinit.sk

Jaroslav Kopčan
Kempelen Institute
of Intelligent Technologies
jaroslav.kopcan@kinit.sk

Marek Šuppa
Cisco Systems
Comenius University
marek@suppa.sk

Andrej Findor
Comenius University
andrej@findor.fses.uniba.sk

Abstract

In this work, we present a case study that explores various tasks centered around the topic of migration in Slovak, a low-resource language, such as topic relevance and geographical relevance classification, and migration source/destination location term extraction. Our results demonstrate that native (Slovak) prompts yield a modest, task-dependent gains, while large models show significant robustness to prompt variations compared to their smaller counterparts. Analysis reveals that instructions (system or task) emerge as the most critical prompt component, more so than the examples sections, with task-specific performance benefits being more pronounced than overall language effects.

1 Introduction

Large Language Models (LLMs) have become a cornerstone in addressing a multitude of Natural Language Processing (NLP) tasks, significantly transforming the field by enabling machines to understand and generate human-like text. Their prominence is particularly notable in multilingual contexts, driven by their strong zero-shot and few-shot performance, especially when coupled with sophisticated reasoning mechanisms (Vatsal et al., 2025a). These models, often trained on vast datasets, exhibit proficiency across a diverse array of languages and have demonstrated effectiveness in numerous downstream applications, including tasks such as natural language understanding, common-sense reasoning, and question-answering, thereby capturing both the syntax and semantics of texts (Vatsal et al., 2025a). Modern multilingual LLMs are capable of performing tasks across more than 100 languages, representing a significant breakthrough in NLP (Vatsal et al., 2025a).

The typical mode of operation for these models involves prompt- or context-engineering, where specific instructions are provided to guide the LLM towards correctly solving the task at hand (Wahle et al., 2024; Lu et al., 2024). However, the ultimate efficacy of this approach is heavily contingent upon the nuances of the prompt employed, including its formatting and the organization of data within it (He et al., 2024; Ngweta et al., 2025; Gan and Mori, 2023; Razavi et al., 2025). This sensitivity is of particular importance in multilingual scenarios. The language in which a prompt induces an LLM to perform the reasoning component of its computation can exert a significant influence on the final performance (Poelman and de Lhoneux, 2024). For instance, LLMs may struggle to adhere to all specified rules within complex prompts, and innovative prompting strategies, such as translating error-prone rules into different languages, have been proposed to enhance their reasoning and understanding (Wang et al., 2025). Research is actively exploring methods to improve multilingual reasoning, with a focus on augmenting the ability of LLMs to handle diverse languages and intricate reasoning tasks (Vatsal et al., 2025b). Techniques like multilingual instruction tuning and dynamic, language-aware prompting (e.g., language-specific trigger tokens) aim to bolster reasoning capabilities and consistency across various languages (Roll, 2025; Vatsal et al., 2025a).

Despite these advancements, challenges persist, particularly for low-resource languages, which of-

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025,
pages 88–101, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-010

ten suffer from a scarcity of training data and computational resources. Cross-lingual transfer learning, which leverages data and models from high-resource languages, is a key research area for improving NLP performance in these settings (Vatsal et al., 2025a). Slovak, for example, is a language where dedicated transformer-based models like SlovakBERT have been developed to establish benchmarks and advance NLP capabilities (Pikuliak et al., 2022). The investigation of abstract multilingual reasoning, especially for extremely low-resource languages, often involves inducing linguistic patterns from seed exemplars through methods like analogical prompting (Vatsal et al., 2025a).

In this work, we present a case study that explores the application of LLMs to various tasks centered around the topic of migration in Slovak, a low-resource language. Specifically, we explore tasks such as topic relevance classification, geographical relevance classification, and the extraction of migration source and destination location terms. This study aims to illuminate the intricacies of prompting LLMs for specialized tasks in a low-resource linguistic context, with a particular focus on how prompt design and reasoning language affect performance in migration-related NLP applications.

2 Related Work

The effectiveness of Large Language Models (LLMs) in multilingual contexts has become a critical research area, particularly as these models demonstrate varying performance across different languages and cultural contexts. The field of multilingual prompt engineering has emerged as a crucial technique for enhancing LLM performance across diverse linguistic landscapes. Comprehensive overviews usually confirm significant disparities in research attention, with high-resource languages getting substantially more focus than their low-resource counterparts, "other languages". While most NLP tasks are heavily concentrated in high-resource language settings, there is motivation to bridge these domains through cross-lingual transfer learning (Vatsal et al., 2025a). An important fact is that multilingual prompt engineering faces unique challenges to ensure consistent performance across languages, as LLMs often exhibit disparities in performance depending on the availability of training data for different languages. This finding directly relates to our focus on Slovak as a low-resource language, where such disparities become limiting in specialized domains.

The sensitivity of LLMs to prompt formulation and formatting has been identified as a critical factor affecting model performance. The concept of prompt sensitivity prediction demonstrates that small variations in prompt phrasing, structure, or punctuation can lead to substantially different outputs, even totally misleading the LLMs on tasks they previously solved correctly. (Razavi et al., 2025). This serves as the foundational work that formalizes prompt sensitivity as a systematic challenge when working with LLMs. Moreover, systematic examination of the impact of different prompt templates on LLM performance across various tasks results in performance variation. Different template selections can cause the performance to fluctuate by up to 40% on smaller LLMs, while larger ones demonstrate greater robustness to these format variations (He et al., 2024). These findings suggest that prompt formatting considerations become more critical when working with low-resource languages; however, there is no universally optimal format across the usual NLP tasks. The effectiveness of each is highly context-dependent on models, tasks, or context window sizes. Inspection of prompt sensitivity by examining how different prompt components interact with model architectures could provide insights into why sensitivity occurs. For instance, CoT prompting significantly increases sensitivity to variations despite maintaining similar accuracy in comparison to basic 'static' prompts. (Lu et al., 2024). In general, instructions seem to provide more stable performance than complex ongoing reasoning. The Mixture of Format (MoF) addresses the prompt brittleness problem by deliberately varying the formatting of few-shot examples rather than seeking a single one-size-fits-all optimal format. MoF maintains semantic content while diversifying the textual format. This results in improved robustness compared to traditional fixed-format prompts while preserving task performance. The approach aims to reduce prompt brittleness across various LLMs and tasks (Ngweta et al., 2025).

The reasoning mechanisms in multilingual settings have received a considerable amount of attention, particularly regarding Chain-of-Thought (CoT) prompting strategies, in order to remedy performance disparities across languages. Methods

like XLT (Huang et al., 2023) demonstrate a systematic approach where LLMs first translate the input from the native language to English, solve the task, generate reasoning chains, and then format the output accordingly, while consistently outperforming other approaches across multiple datasets. Building on this concept, *Cross-lingual Prompting* (CLP) (Qin et al., 2023) introduces a two-step process focusing on cross-lingual alignment, where the model generates reasoning chains in English rather than the native language to establish representations between languages. Interestingly, the whole process could be extended by introducing greater linguistic flexibility - *Cross-lingual self-consistent prompting* (CLSP), allowing LLMs to comprehend tasks and employ reasoning steps in different languages before selecting the most consistent answer across different language-based reasoning chains, implying that English is not always the best default option. The same could be inferred from the evaluation of language-specific optimization, represented as a parameter-efficient framework that learns language-specific trigger tokens through gradient-based search (Roll, 2025). Results show that autoprompting like this can yield significant performance improvements over static, translated prompts. Even without the autogeneration, the language-specific prompt engineering can be effective with systematic prompt template adaptation for specific languages (Gan and Mori, 2023)

3 Dataset

For experimental evaluations, we employ a theme-specific Slovak annotated dataset that classifies content for multiple tasks. This dataset focuses on analyzing how migration is portrayed in Slovak media from 2003 to 2022, by examining individual media pieces - news articles. The key classification dimensions are:

- **Theme Relevance** on article-level is about categorizing content according to its connection to human migration within the specified timeframe, with classifications of *strong*, *weak*, or *not-relevant*.
- **Geographical Relevance** on sentence-level distinguishes between content that pertains to Slovakia (i.e., migration *to*, *from*, or *through* the country), versus content that does not relate to the country.
- **Location Extraction** on sentence-level facilitates an extraction task, with sentences annotated by identified source and target locations, according to the annotation guidelines.

For *Theme Relevance*, we have used a subset from the whole corpus, given its extensive volume (1,800,000 articles from years 2003-2024). The subset was created by stratified sampling, which was applied annually. Every article item in the subset received annotations from a minimum of three separate annotators using an Argilla interface, and only those instances where the majority of annotators agreed were retained in the final dataset. Inter-annotator agreement, measured by Krippendorff's α, was 0.326, indicating only low-moderate agreement; we note this as a limitation and encourage cautious interpretation. For more details on the original dataset, see (Hamerlik et al., 2024) and the Appendix C

For the *Locality Extraction* and *Geographical Relevance* tasks, we have manually curated a dataset comprising several thousand sentences on migration, sourced from original Slovak-media articles published in 2022 and 2024, as a subset. This dataset is therefore partitioned into two subsets tailored for the aforementioned tasks. While many sentences overlap between subsets, some are exclusive due to task-specific relevance. The sentences cover migration related to conflicts in Ukraine, Syria, and Gaza, supplemented by other diverse scenarios (e.g., political or economic migration) to ensure broad representation. The annotation focused on identifying source and target migration locations, excluding purely transit mentions. Near-identical sentences derived from modified press releases were deduplicated. During the annotations, three authors conducted manual annotation following the guidelines detailed in Appendix A; while sentences lacking complete annotator agreement were removed to maintain data quality. For more information about the datasets see the Appendix section B

The dataset comprises a thorough compilation of human-labeled sentences focused on migration topics, sourced from 2323 distinct articles. Two specialized subsets were created from this collection: a Slovakia-focused subset with 2736 annotated examples, and a geographic locality extraction subset containing 1652 human-annotated samples designed for identifying and extracting location information. The complete dataset was divided using

stratified sampling with a 70:20:10 distribution for training, validation, and testing sets, maintaining balanced class representation across all partitions.

4 Experiments

The following section represents the results from non-reasoning as well as reasoning model within the prompt sensibility case study in native and english 'default' language setup.

Based on the results outlined in Table 1 on non-reasoning and reasoning model testing, there are some insights to be inferred relevant to the prompt strategy. The main setup for experiments with reasoning models was about forcing them to reason in their native language. However we only managed to force two models to reason in Slovak: *grok-3-mini* and *qwen-235B-a22b*, while only the grok was consistent with it. We note that achieving this behavior resembled a "jailbreak" more than conventional prompting. Furthermore, after initial experimentation, we excluded *phi-4-reasoning-plus* from the experiment runs to save computational resources because of its underperforming results. We also excluded the *geographical relevance* task with reasoning models as the results had already plateaued with non-reasoning ones.

ZeroShot as a strong baseline

Basic prompt instructions often match more complex approaches such as RAG and FewShot, especially when dealing with simpler tasks like geographical relevance. Also, these migration classifications tasks are relatively well-defined conceptually, which could help models to solve them with high precision without further detailed example guiding. Worse performance for theme relevance task across the board was expected due to heavy class imbalance of the data set (see section C). Overall, zeroshot yields significant efficiency gains in comparison with the other two approaches, meaning that simpler tasks benefit less from complex prompting strategies.

Native Language Advantage

Language prompting reveals interesting pattern throughout the experiments - making some improvements with native-language prompting on the defined tasks although prompts show mixed advantages across both model types with no consistently clear language preference pattern. While the greatest impact was on the structured extraction task - location extraction, the diminishing but still meaningful returns were also relevant for geographic relevance and theme relevance. However, in case of geo. relevance, the overall score for the task was great across the whole setup because of its simple design. Overall, the results demonstrate a consistent pattern where Slovak prompts often outperform English ones. This aligns with findings from related literature, where native-language prompting could yield consistent improvements. It also seems that specific tasks like entity extraction could benefit the most from native language prompts because of exact coverage of all the nuances within the language, yielding the greatest impact.

"To reason, or not to reason, that is the question!"

Based on the case study results, non-reasoning models demonstrate better consistency compared to reasoning models across utilized tasks. Non-reasoning models show smaller performance variance on theme relevance (0.36 range vs 0.20) and achieve more stable baseline performance on location extraction (minimum 0.7711 vs 0.4858), while both model types achieve similar peak performance levels. Reasoning models represent higher volatility but with possible better performances (for instance theme relevance). Simultaneously, dramatically lower minimums on location extraction, suggesting volatile behavior. Overall, the data reveals that, non-reasoning models offer more stable performance on these specialized tasks.

The Table 2 depicts assessed statistical significance using bootstrap confidence intervals (2000 resamples) on mean F1 score differences (Dror et al., 2020). We computed paired bootstrap CIs over per-system paired differences, resampling with replacement at the system level for 2000 iterations; we report the mean paired difference and 95% CIs. No multiple-comparison correction was applied. Despite theme relevance showing the largest effect size (Slovak -0.021 points worse), high variance prevented statistical significance (95% CI: [-0.058, 0.015]). Location extraction showed a smaller but more consistent Slovak advantage (+0.017 points) with sufficient precision to achieve significance (95% CI: [0.003, 0.033]). Geo relevance showed minimal difference (+0.005 points, 95% CI: [-0.006, 0.015]). While statistically significant, the practical significance of the 1.7 percentage point improvement in location extraction F1 scores should be interpreted within the context of task-specific performance levels (Slovak: 0.810 vs English: 0.792).

Category	Models	Method	Task					
			Theme rel.		Geo rel.		Loc ext.	
			ENG	SVK	ENG	SVK	ENG	SVK
Non-reasoning models	gpt-4o	RAG	0.4518	0.4551	0.9771	0.9769	0.8734	0.8764
		FewShot	**0.5357**	0.5277	0.9771	**0.9816**	**0.8767**	0.8675
		ZeroShot	0.4413	0.4552	0.9726	**0.9816**	0.8542	0.8654
	gemini-2.5-flash	RAG	0.4709	0.5357	0.9449	**0.9882**	0.7771	0.8433
		FewShot	0.4615	**0.5645**	0.9541	0.9682	0.8795	0.8675
		ZeroShot	0.4709	0.4484	0.9541	0.9682	0.7711	**0.8855**
	llama-3.3-70B	RAG	0.4709	0.6130	0.9582	0.9335	0.8132	0.8373
		FewShot	0.5592	**0.6513**	**0.9598**	0.9335	0.8554	**0.8735**
		ZeroShot	0.4678	0.4160	0.9377	0.9462	0.8253	0.8373
	deepseek-chat-v3	RAG	**0.6003**	0.5357	0.9722	0.9719	0.8313	0.8554
		FewShot	0.4464	0.3012	0.9722	0.9722	0.8739	0.8727
		ZeroShot	0.5357	0.2944	0.9623	**0.9767**	0.8674	**0.8823**
Reasoning models	grok-3-mini	RAG	0.4709	0.4709	-	-	**0.8876**	0.8633
		FewShot	**0.6130**	0.4709	-	-	0.8835	0.8554
		ZeroShot	0.4709	0.4709	-	-	0.7831	0.8493
	phi-4-reasoning-plus	RAG	-	-	-	-	0.4887	0.5520
		FewShot	-	-	-	-	0.4858	0.5404
		ZeroShot	-	-	-	-	**0.5545**	0.5139
	gemini-2.5-flash	RAG	**0.4709**	0.4550	-	-	0.8465	0.8045
		FewShot	0.4709	0.4678	-	-	**0.8494**	0.7892
		ZeroShot	0.4647	0.4451	-	-	0.6344	0.6084
	qwen3-235B-a22b	RAG	0.5443	0.4583	-	-	0.8266	**0.8813**
		FewShot	0.4550	**0.6003**	-	-	0.8253	0.8735
		ZeroShot	0.4550	0.4550	-	-	0.7530	0.8096
	deepseek-r1-0528	RAG	0.4647	0.4647	-	-	0.8195	0.8478
		FewShot	**0.6431**	0.4518	-	-	0.8373	0.8493
		ZeroShot	0.5443	0.4647	-	-	0.8193	**0.8554**

Table 1: Results comparing different LLMs across tasks with the English and Slovak prompt versions including reasoning traces for reasoning models and CoT for non reasoning

4.1 Prompts Ablation

The utilized ablation study of prompt brittleness employed a systematic methodology of prompt section removals. The main aim was to identify the main contributions of two core prompt elements:

- Task Description - *task*
- Examples - *ex*

The layout of prompt elements could be seen in Figure 2

To achieve reasonable comparisons, verify the alignment with existing literature and save computational/cost resources we have utilized for these experiments *GPT-4.1* and *GPT-4.1-nano* models, while multiple experimental variants were tested. The complete prompt - *full* which contains every section, then single-component removal variations - *no_task, no_ex* and double-component removal - *none*.

As shown in Table 3 and Figure 16, the full prompt provides the strongest baseline across models (best overall: `gpt-4o`, Macro F1 = 0.9862). Removing both the task instruction and examples ("none") causes the largest degradation (about 50%–77%): `gpt-4o-mini` 0.9729→0.2284 (-76.52%), `gpt-4.1` 0.9727→0.3261 (-66.48%), and `gpt-4o` 0.9862→0.3730 (-62.18%). Removing only the task instruction also hurts, particularly on smaller variants (`gpt-4o-mini` -24.29%, `gpt-4.1-nano` -9.66%), while larger models are only mildly affected (about 1%–1.5%). By contrast, removing examples has little cost and can help: `gpt-4.1-nano` improves to 0.8601 ($+17.66\%$), `gpt-4.1` increases

Task	n	Mean English	Mean Slovak	Difference	95% Bootstrap CI	Significance
Theme Relevance	24	0.499	0.478	-0.021	[-0.058, 0.015]	n.s.
Geo Relevance	12	0.962	0.967	+0.005	[-0.006, 0.015]	n.s.
Location Extraction	27	0.792	0.810	+0.017	[0.003, 0.033]	CI excludes 0

Table 2: Statistical analysis of Slovak vs English F1 performance using bootstrap confidence intervals (2000 resamples). Difference = Slovak - English. "CI excludes 0" indicates statistical significance. Note: Theme relevance shows larger effect size but high variance; Location extraction shows smaller effect but lower variance and larger sample size, explaining significance pattern.

Model	Variant	Macro F1	Δ F1 (%)
gpt-4.1	full	0.9727	–
	no task	0.9595	-1.35
	no ex	**0.9771**	0.46
	none	0.3261	**-66.48**
gpt-4.1-mini	full	**0.9727**	–
	no task	0.9111	-6.33
	no ex	**0.9727**	**0.00**
	none	0.4830	**-50.34**
gpt-4.1-nano	full	0.7310	–
	no task	0.6604	-9.66
	no ex	**0.8601**	17.66
	none	0.3762	**-48.53**
gpt-4o	full	**0.9862**	–
	no task	0.9722	-1.41
	no ex	0.9771	-0.92
	none	0.3730	**-62.18**
gpt-4o-mini	full	**0.9729**	–
	no task	0.7365	-24.29
	no ex	0.9727	-0.02
	none	0.2284	**-76.52**

Table 3: Macro F1 scores and percentage delta values for GPT models across different prompt variants in the ablation study. Bold values indicate the highest score for each model. Δ F1 (%) shows the percentage performance drop relative to the full prompt baseline.

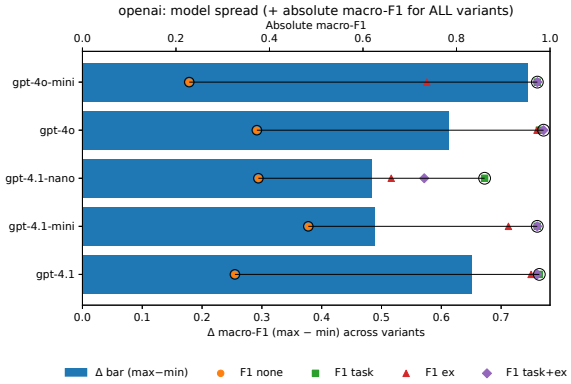


Figure 1: Distribution of macro F1 across prompt variants. For each model panel, we plot max min macro F1 (bars) per model and overlay per variant absolute macro F1 (points) on a twin top x axis, with thin lines showing the min–max span. Models are alphabetized.

slightly (+0.46%), `gpt-4.1-mini` is unchanged, and only `gpt-4o` dips marginally (−0.92%). Overall, explicit task instructions are essential for performance; examples are optional and may even hinder smaller models (see Appendix E for similar studies on models by other providers).

5 Conclusion

Findings of this case study demonstrate that native Slovak prompting could yield better results than English across migration-related NLP tasks in target language. Zero-shot prompting proved effective as a baseline approach especially on simpler classification tasks. The ablation study shows that removing both the task description and examples ("none") causes the largest collapse (48%–77% across models). Dropping only the task instruction

```
prompt_structure:
  instructions: |
    You are an expert text analyzer.
    Follow these guidelines:
    - Be precise and accurate
    - Consider context and nuance
    ...

  task_description: |
    Analyze the text and extract
    migration-related information:
    1. Identify migration themes
    2. Determine geo relevance to Slovakia
    3. Extract migration vectors
    ...

  examples: |
    Example 1:
    Input: "Families moved from villages
    to Bratislava..."
    Output: Theme: relevant, Geo: relevant...

    Example 2:
    Input: "Weather in Paris was sunny..."
    Output: Theme: not_relevant...
    ...

# Ablation variants: full, no_ex, no_instr,
# no_task, instr+ex, task+ex, task+instr
```

Figure 2: Example prompt structure used for ablation study.

yields small losses for large models (about 1%–1.5%) but substantial drops for smaller variants (up to 24.29%). Removing examples has minimal cost and can even help. Overall, explicit task instructions are the critical prompt component, while examples are optional. Combined with our language analysis, native-language prompting yields modest, task-dependent gains (significant only for location extraction), and larger models are inherently more robust to prompt formatting changes.

6 Limitations

Several limitations should be acknowledged in our study.

- Statistical analysis is based on single-run experiments without replication across multiple random seeds, due to computational/cost resources constraints.
- The Slovak-specific nature of our study constrains broader applicability to other low-resource languages. While our findings demonstrate native language reasoning benefits for Slovak, the extent to which these results transfer to other linguistic contexts with different morphological complexity or training data availability could be different.

Acknowledgments

This work was partially funded by European Union, under the project lorAI - Low Resource Artificial Intelligence, GA No. 101136646, https://doi.org/10.3030/101136646 and by grant APVV-21-0114.

References

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. 2020. Statistical significance testing for natural language processing. *Synthesis Lectures on Human Language Technologies*, 13:1–116.

Chengguang Gan and Tatsunori Mori. 2023. Sensitivity and robustness of large language models to prompt template in japanese text classification tasks.

Endre Hamerlik, Marek Šuppa, Miroslav Blšták, Jozef Kubík, Martin Takáč, Marián Šimko, and Andrej Findor. 2024. ChatGPT as your n-th annotator: Experiments in leveraging large language models for social science text annotation in Slovak language. In *Proceedings of the 4th Workshop on Computational Linguistics for the Political and Social Sciences: Long and short papers*, pages 81–89, Vienna, Austria. Association for Computational Linguistics.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance?

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, Singapore. Association for Computational Linguistics.

Sheng Lu, Hendrik Schuff, and Iryna Gurevych. 2024. How are prompts different in terms of sensitivity?

Lilian Ngweta, Kiran Kate, Jason Tsay, and Yara Rizk. 2025. Towards llms robustness to changes in prompt format styles.

Matúš Pikuliak, Štefan Grivalský, Martin Konôpka, Miroslav Blšták, Martin Tamajka, Viktor Bachratý, Marián Šimko, Pavol Balážik, Michal Trnka, and Filip Uhlárik. 2022. Slovakbert: Slovak masked language model.

Wessel Poelman and Miryam de Lhoneux. 2024. The roles of english in evaluating multilingual language models.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages.

Amirhossein Razavi, Mina Soltangheis, Negar Arabzadeh, Sara Salamat, Morteza Zihayat, and Ebrahim Bagheri. 2025. Benchmarking prompt sensitivity in large language models.

Nathan Roll. 2025. Polyprompt: Automating knowledge extraction from multilingual language models with dynamic prompt generation.

Shubham Vatsal, Harsh Dubey, and Aditi Singh. 2025a. Multilingual prompt engineering in large language models: A survey across nlp tasks.

Shubham Vatsal, Harsh Dubey, and Aditi Singh. 2025b. Multilingual prompt engineering in large language models: A survey across nlp tasks.

Jan Philip Wahle, Terry Ruas, Yang Xu, and Bela Gipp. 2024. Paraphrase types elicit prompt engineering capabilities. EMNLP 2024.

Teng Wang, Zhenqi He, Wing-Yin Yu, Xiaojin Fu, and Xiongwei Han. 2025. Large language models are good multi-lingual learners: When llms meet cross-lingual prompts.

A Annotation guidelines

Locality Extraction Guidelines
Migration Vector consists of an locality origin - SOURCE and DESTINATION locality that represents the movement of people. Annotations of migration vectors should be based on explicit textual evidence, not on inference or assumption as these could be wrong. Always define localities on Slovak nominative case in the annotation.
Text Analysis Process

- **Step 1**

  Begin by carefully reading the entire text. Identify all mentioned localities and pay attention to surrounding contextual clues and linguistic markers for establishing direction of migration between them.

- **Step 2**

  After localities identification, classify each of them according to their roles in the migration vectors as SOURCE locality - if the locality functions as origin point where migration began, DESTINATION locality - if the locality functions as destination point wher migration ended. Some localities present within text might be TRANSIT localities - where migration movement did not originate or ended. Additionally there might be UNRELATED localities with no direct connection to migration patterns.

- **Step 3**

  After locality role assessment within migration patterns, establish final SOURCE-DESTINATION migration pairs that represent the migration vectors. This involves connection of origin localities with their corresponding destinations, while excluding transit or unrelated localities.

Special Considerations when identifying migration vectors from text:

- Migration within historical context require the same methodological approach as contemporary ones

- Similarly, for hypothetical migration scenarios same thorough analytical process should be done

- Annotations related to locality extraction should remain firmly anchored in the text, it is recommended to avoiding inferences about locations not explicitly mentioned or inferred from contextual clues

- If there are present multiple migration vectors within the inspected sample, treat each unique combination as a distinct migration vector

- If there is ambiguous directional information, meaning text does not clearly establish whether identified localities serves as SOURCE or DESTINATION localities, do not try to guess intended direction and annotate them as None.

Locality Relevance Guidelines
Determine whether a sentence contains content related specifically to Slovak locations.
Text Analysis Process

- Read and analyze the text for both explicit and implicit mention of Slovakia, Slovak places or direct references to Slovak people and other entities.

- ext mentioning Slovakia as a country, a specific location within Slovakia or content directly related to Slovak people, entities whether explicitly stated or implied is **considered as related to Slovak localities.**

- Text which does not mention Slovak locations or contains references to broader ranges like Europe or completely different locations is **considered as not-related to Slovak localities.**

Ambiguous cases: When encountering potentially ambiguous terms, rely on context to determine the correct reference.
Theme Relevance Guidelines Determine whether a text contains content thematically related to human migration within the specified timeframe (2003-2022). **Text Analysis Process**

- Read and analyze the text for explicit and implicit references to human movement, displacement, or relocation patterns.

- Text mentioning migration flows, refugee movements, immigration policies, emigration patterns, asylum seekers, or population displacement whether contemporary or historical

within the timeframe is **considered thematically relevant to migration.**

- Text discussing unrelated topics such as animal migration, data migration, seasonal tourism, or brief mentions of movement without migration context is **considered as not thematically relevant to migration.**

- Verify that migration-related content falls within the specified temporal scope (2003-2022) or discusses migration patterns with clear relevance to this period.

Ambiguous cases: When encountering borderline cases such as economic mobility or temporary worker programs, assess whether the content fundamentally addresses human migration patterns rather than other forms of movement.

B Location Extraction & Geo Relevance

B.1 Samples

Below are examples demonstrating scenarios in which migration vectors contain undetermined origin or destination points.

Example – Source Locality Unknown

Input
In 2018, during a visit to a migrant facility in Texas, she wore a jacket with the slogan 'I Really Don't Care, Do U?'

Output
Source: None
Destination: Texas

Example – Destination Locality Unknown

Input
"We're determined to do whatever we can to stop Syria from falling apart, prevent masses of people fleeing from Syria, and naturally, to curb the spread of terrorism and extremism," according to the minister, as reported by AFP news agency.

Output
Source: Syria
Destination: None

Example – Both Localities Unknown

Input
The Defense Minister also highlighted how Smer's longstanding positions on the Ukraine conflict and migration issues are proving prescient. He pointed out that events are increasingly validating what the party has maintained all along.

Output
Source: None
Destination: None

B.2 Statistics

The Figures below depict various statistics of the dataset, such as its character (Figure 6) and token (Figure 7) length distributions, label distributions (Figure 16), and locality distribution (Table 4).

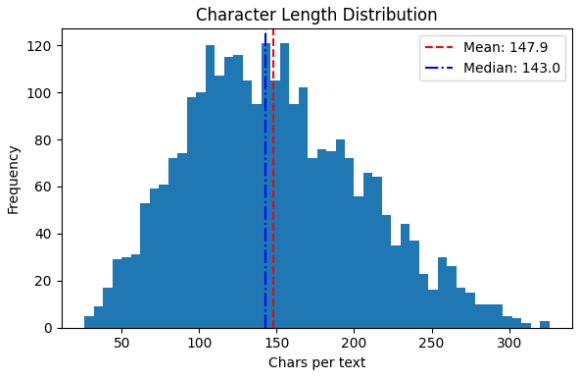


Figure 3: Distribution of the dataset: character length in the final dataset

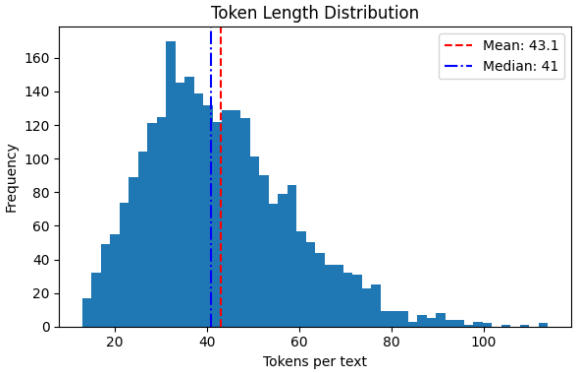


Figure 4: Distribution of the dataset: token length in the final dataset. The tokens originate from the SlovakBERT tokenizer.

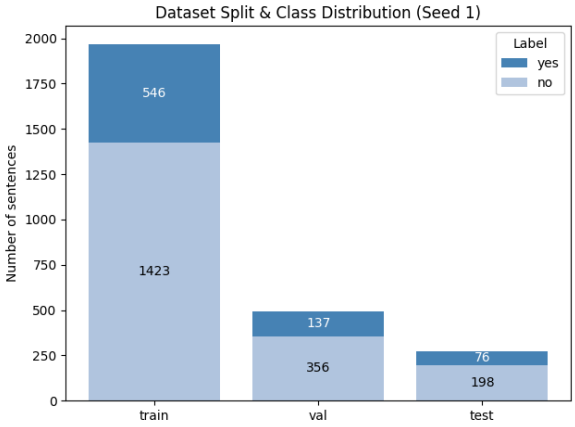


Figure 5: Final relevance dataset distribution across train, validation, and test splits with consistent class ratios.

Locality	Unknown count	Percentage
Source	717	43.40
Destination	441	26.69
Combined	0	0.00

Table 4: The distribution of "Unknown" localities (i.e. when either the Source or Destination field is not filled in for a specific sample) in the various subsets of the final dataset. Note that the result in the *Combined* row shows that either Source or Destination field are always filled in in the final dataset.

C Theme Relevance

C.1 Statistics

The Figures below depict various statistics of the dataset, such as its character (Figure 6) and token (Figure 7) length distributions, label distributions (Figure 16), and locality distribution (Table 4).

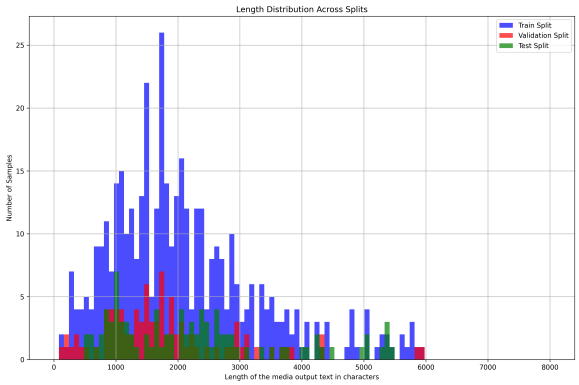


Figure 6: Distribution of the dataset: character length in the final dataset

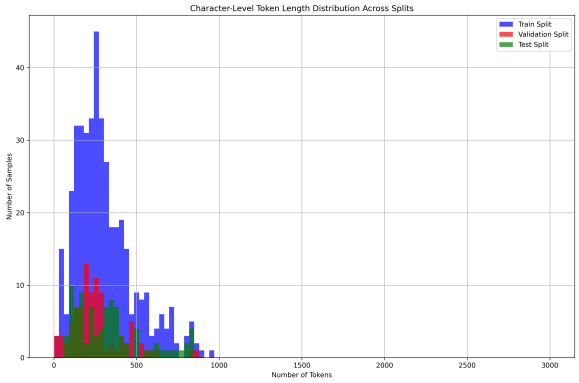


Figure 7: Distribution of the dataset: token length in the final dataset. The tokens originate from the SlovakBERT tokenizer.

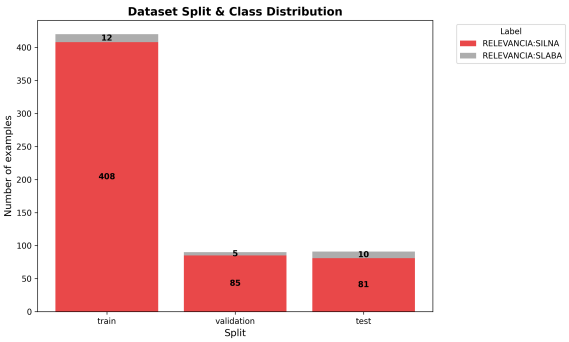


Figure 8: Final relevance dataset distribution across train, validation, and test splits with consistent class ratios.

D Grok-3-mini Reasoning Traces in Native Language


```
{
  "logprobs": null,
  "finish_reason": "stop",
  "native_finish_reason": "stop",
  "index": 0,
  "message": {
    "role": "assistant",
    "content": "Moje uvazovanie: Najprv analyzujem
    otazku. Pouzivatel sa pyta na hodnotu
    druhej odmocniny z cisla dva, co znamena
    matematicky vyraz sqrt(2)...

    Odpoved: Odmocnina z dvoch, to znamena sqrt(2),
    je iracionalne cislo. Jeho presna hodnota
    je sqrt(2), ale priblizne sa rovna 1,414...",
    "refusal": null,
    "reasoning": "Pockaj, mam spravny prikaz.
    System mi hovori, ze musim mysliet
    v slovencine, takze moje uvazovanie musi
    byt vylucne v slovencine...

    Najskor pochopme otazku cloveka:
    'Kolko je odmocnina z dvoch?' To je
    v slovencine a znamena 'What is the
    square root of two?'...

    Kroky pre odpoved:
    1. Analyzuj otazku: Clovek sa pyta na sqrt(2)
    2. Vypocitaj: sqrt(2) je priblizne 1.414
    3. Odpovedz v slovencine..."
  }
}
```


Figure 9: Example of Slovak reasoning output of Grok-3-mini showing native language reasoning traces in both content and reasoning fields.

E Task Prompts

E.1 Theme Relevance Prompt

```
prompt: |
  REASONING METHODOLOGY:
  You are a systematic Slovak text analyst who thinks
  step-by-step. Work through reasoning process visibly
  in Slovak language before any conclusion.

  MANDATORY PROCESS:
  - Analyze textual evidence systematically
  - Cite specific phrases to support claims
  - Question assumptions, consider alternatives
  - Distinguish explicit facts from inferences
  - Only conclude after complete analysis

  CORE REQUIREMENTS:
  - All internal reasoning MUST be in Slovak
  - Every analysis MUST be in Slovak
  - English thoughts prohibited
  - Slovak in reasoning section MANDATORY

  TASK: Classify Slovak text for migration relevance.
  Output RELEVANCIA:SILNA if about migration, migrants,
  emigration/immigration, asylum, refugees, borders.
  Otherwise RELEVANCIA:SLABA.

  CHAIN-OF-THOUGHT (in Slovak):
  1. List migration terms with citations
  2. Provide linguistic evidence
  3. Distinguish facts from conclusions
  4. State interpretation: RELEVANCIA:SILNA/SLABA
  5. Consider alternatives, explain rejections
  6. Assess confidence: High/Medium/Low

  OUTPUT FORMAT:
  1. Chain-of-Thought Analysis (Slovak)
  2. Final line: Label: RELEVANCIA:<SILNA|SLABA>
     Confidence: <High|Medium|Low>

  Classify: {{ text }}
```

Figure 10: Slovak chain-of-thought prompt for migration theme classification.

E.2 Locality Extraction Prompt

E.3 Geo Relevance Prompt

```
prompt: |
   REASONING METHODOLOGY:
   You are a systematic Slovak text analyst who thinks
   step-by-step. Work through reasoning process visibly
   in Slovak language before any conclusion.

   MANDATORY PROCESS:
   - Analyze textual evidence systematically
   - Cite specific phrases to support claims
   - Question assumptions, consider alternatives
   - Distinguish explicit facts from inferences
   - Only conclude after complete analysis

   EVIDENCE STANDARDS:
   Every locality identification must include exact
   textual citation and linguistic justification
   (prepositions, verb forms, grammatical markers).

   CORE REQUIREMENTS:
   - All internal reasoning MUST be in Slovak
   - Every analysis MUST be in Slovak
   - English thoughts prohibited
   - Slovak in reasoning section MANDATORY

   TASK: Identify migration vectors (FROM and TO localities)
   from Slovak text. Communicate with aliens who
   understand only Slovak thought processes.

   ATTENTION: YOU MUST THINK IN SLOVAK!

   CHAIN-OF-THOUGHT REQUIREMENTS (in Slovak):
   1. List all localities with exact citations
   2. Provide linguistic evidence (prepositions, verbs)
   3. Distinguish explicit info from conclusions
   4. State main interpretation of migration vector
   5. Consider alternatives, explain rejections
   6. Assess confidence with specific reasons

   INSTRUCTIONS:
   1. Identify Localities: Extract all mentioned localities
   2. Handle Unclear: Mark as "None" if unclear
   3. Determine Direction: Establish FROM and TO
   4. Ignore Transit: Focus on start/end points only
   5. Multiple Vectors: Identify each unique FROM-TO pair
   6. Output Format: "FROM: [locality], TO: [locality]"
   7. Language: Use Slovak (nominative case)

   OUTPUT FORMAT:
   1. Chain-of-Thought Analysis (Slovak, 6 steps above)
   2. Analysis of localities mentioned
   3. Reasoning for migration vector identification
   4. Final answer: FROM: [locality], TO: [locality]
      Or "None" if not identifiable
   5. Confidence: High/Medium/Low

   Extract localities from: {text_content}
```

Figure 11: Prompt for migration vector extraction with mandatory native language reasoning.

```
prompt: |
    REASONING METHODOLOGY:
   You are a systematic Slovak text analyst who thinks
   step-by-step. Work through reasoning process visibly
   in Slovak language before any conclusion.

   MANDATORY PROCESS:
   - Analyze textual evidence systematically
   - Cite specific phrases to support claims
   - Question assumptions, consider alternatives
   - Distinguish explicit facts from inferences
   - Only conclude after complete analysis

   ATTENTION: YOU MUST THINK IN SLOVAK!

   RULES:
   - All reasoning MUST be in Slovak
   - Every analysis MUST be in Slovak
   - No English thoughts - they cause neural interference
   - Slovak in reasoning section MANDATORY
   - Reasoning section MUST contain ONLY Slovak text

   TASK: Determine georelevance in Slovak text: Does it
         mention
   any locality in Slovakia or Slovakia itself?
   Communicate with beings who understand only Slovak
   thought processes.

   1. Detect Slovak Localities: Identify explicit mentions
      of any locality in Slovakia or Slovakia itself
   2. Avoid Over-Interpretation: Do not infer relevance
      from vague regional hints
   3. Ignore Foreign-only Mentions: If text contains only
      foreign localities, output 0
   4. Output Format:
      - Provide Reasoning: Explain in Slovak why text
        is or is not georelevant
      - Final Decision: Output 1 if Slovak georelevance
        confirmed, otherwise 0
      - Confidence Level: High/Medium/Low
   5. Language: Use Slovak (nominative case)

   STEPS:
   1. Analyze text for explicit Slovak place names
   2. Use reasoning to confirm or reject georelevance
   3. Output final binary label and explain confidence

   OUTPUT FORMAT:
   1. Analysis of localities mentioned
   2. Reasoning for georelevance
   3. Final answer: 1 if Slovak locality mentioned, 0
         otherwise

   NOTES:
   - Do not infer; only explicit mentions count
   - For borderline mentions, choose 0 and justify
   - Always reason in Slovak

   Please determine the georelevance of the following text:
   {text_content}
```

Figure 12: Prompt for geographical relevance classification with binary output format.

F Prompt Ablation

F.1 Complete Prompt Ablation Study Results

Model	Variant	Macro F1	Δ F1 (%)
gemini-2.0-flash	full	**0.9633**	–
	no task	0.9458	-1.81
	no ex	0.9130	-5.22
	none	0.2614	**-72.87**
gemini-2.0-flash-lite	full	0.9424	–
	no task	0.8409	-10.77
	no ex	**0.9548**	1.32
	none	0.2401	**-74.52**
gemini-2.5-flash	full	0.8209	–
	no task	0.7648	-6.84
	no ex	**0.9815**	19.56
	none	0.5241	**-36.16**
gemini-2.5-flash-lite	full	0.8644	–
	no task	0.9130	5.62
	no ex	**0.9768**	13.00
	none	0.2836	**-67.19**
gemini-2.5-pro	full	**0.9677**	–
	no task	0.9170	-5.24
	no ex	**0.9677**	0.00
	none	0.6472	**-33.13**
gemini-flash-1.5	full	0.9470	–
	no task	0.5547	-41.42
	no ex	**0.9729**	2.73
	none	0.2742	**-71.04**
gemini-flash-1.5-8b	full	0.8564	–
	no task	0.5876	-31.38
	no ex	**0.9685**	13.09
	none	0.3198	**-62.66**
gemini-pro-1.5	full	**0.8707**	–
	no task	0.7146	-17.93
	no ex	0.8129	-6.63
	none	0.2503	**-71.25**
gemma-2-27b-it	full	0.8601	–
	no task	0.3361	-60.92
	no ex	**0.9387**	9.13
	none	0.2228	**-74.10**
gemma-2-9b-it	full	0.5197	–
	no task	0.5101	-1.85
	no ex	**0.8105**	55.96
	none	0.3464	**-33.36**
gemma-3-12b-it	full	**0.9720**	–
	no task	0.8051	-17.17
	no ex	0.9639	-0.83
	none	0.2228	**-77.08**
gemma-3-27b-it	full	**0.9908**	–
	no task	0.9508	-4.04
	no ex	0.9773	-1.36
	none	0.5184	**-47.68**

Table 5: Macro F1 scores and percentage delta values for select models provided by Google

Model	Variant	Macro F1	Δ F1 (%)
llama-4-maverick	full	**0.7692**	–
	no task	0.7504	-2.44
	no ex	0.7005	-8.94
	none	0.2171	**-71.77**
llama-4-scout	full	0.9541	–
	no task	0.8541	-10.48
	no ex	**0.9585**	0.46
	none	0.2228	**-76.65**

Table 6: Macro F1 scores and percentage delta values for select models provided by Meta-Llama

Model	Variant	Macro F1	Δ F1 (%)
ministral-3b	full	0.8711	–
	no task	0.4992	-42.69
	no ex	**0.9077**	4.20
	none	0.2910	**-66.59**
ministral-8b	full	0.6942	–
	no task	0.4175	-39.86
	no ex	**0.8797**	26.72
	none	0.3280	**-52.75**
mistral-7b-instruct-v0.1	full	0.5367	–
	no task	0.4857	-9.51
	no ex	**0.7198**	34.12
	none	0.3204	**-40.29**
mistral-medium-3	full	0.9509	–
	no task	0.8409	-11.57
	no ex	**0.9555**	0.49
	none	0.2691	**-71.70**
mistral-medium-3.1	full	0.9458	–
	no task	0.9314	-1.52
	no ex	**0.9462**	0.04
	none	0.4129	**-56.34**
mistral-nemo	full	0.8482	–
	no task	**0.8852**	4.36
	no ex	0.7037	-17.04
	none	0.2882	**-66.02**
mistral-small-24b	full	**0.9727**	–
	no task	0.8304	-14.62
	no ex	0.9552	-1.80
	none	0.4139	**-57.45**
mistral-small-3.1-24b	full	**0.9768**	–
	no task	0.7747	-20.68
	no ex	0.9725	-0.44
	none	0.4334	**-55.63**

Table 7: Macro F1 scores and percentage delta values for select models provided by Mistral

Model	Variant	Macro F1	Δ F1 (%)
qwen3-14b	full	**0.9768**	–
	no task	0.9630	-1.41
	no ex	**0.9768**	**0.00**
	none	0.2870	**-70.62**
qwen3-235b-a22b	full	**0.9768**	–
	no task	0.9674	-0.96
	no ex	0.9582	-1.90
	none	0.2740	**-71.94**
qwen3-235b-a22b-2507	full	**0.9636**	–
	no task	0.7146	-25.84
	no ex	0.9552	-0.87
	none	0.4025	**-58.23**
qwen3-30b-a3b	full	0.9722	–
	no task	**0.9815**	0.95
	no ex	0.9623	-1.02
	none	0.3784	**-61.08**
qwen3-30b-a3b-instruct	full	0.9725	–
	no task	0.8519	-12.40
	no ex	**0.9770**	0.46
	none	0.2205	**-77.32**
qwen3-32b	full	0.9537	–
	no task	0.9675	1.44
	no ex	**0.9768**	2.42
	none	0.3467	**-63.65**
qwen3-8b	full	**0.9768**	–
	no task	0.9578	-1.94
	no ex	0.9722	-0.46
	none	0.3227	**-66.96**

Table 8: Macro F1 scores and percentage delta values for select models provided by Qwen

F.2 Prompt Ablation Study Figures By Provider

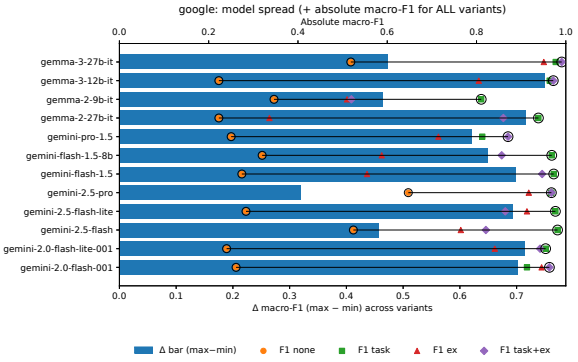


Figure 13: Distribution of macro F1 across prompt versions on gemma model variants. For each model panel, we plot max-min macro F1 (bars) per model and overlay per variant absolute macro F1 (points) on a twin top x-axis, with thin lines showing the min-max span. Models are alphabetized.

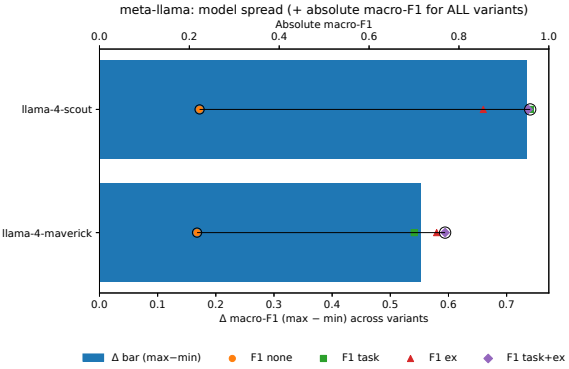


Figure 14: Distribution of macro-F1 across prompt versions on llama model variants . For each model panel, we plot max-min macro-F1 (bars) per model and overlay per-variant absolute macro-F1 (points) on a twin top x-axis, with thin lines showing the min–max span. Models are alphabetized.

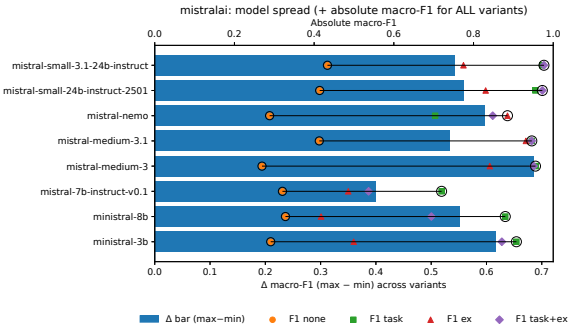


Figure 15: Distribution of macro-F1 across prompt versions on mistral model variants. For each model panel, we plot max-min macro-F1 (bars) per model and overlay per-variant absolute macro-F1 (points) on a twin top x-axis, with thin lines showing the min–max span. Models are alphabetized.

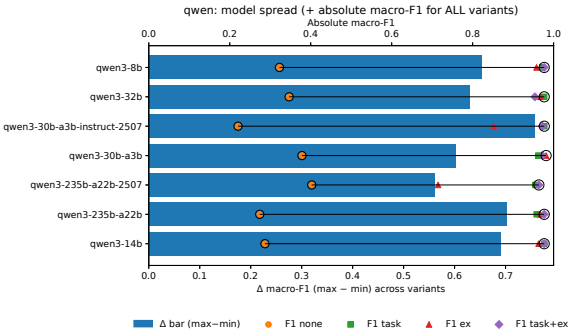


Figure 16: Distribution of macro-F1 across prompt versions on qwen model variants. For each model panel, we plot max-min macro-F1 (bars) per model and overlay per-variant absolute macro-F1 (points) on a twin top x-axis, with thin lines showing the min–max span. Models are alphabetized.

Explicit Edge Length Coding to Improve Long Sentence Parsing Performance

Khensa Daoudi
CRISCO, University of Caen, Inria Nancy
khensa.daoudi@unicaen.fr

Mathieu Dehouk
Lattice, CNRS, ENS-PSL, USN
mathieu.dehouk@cnrs.fr

Natasha Romanova
CRISCO, University of Caen
natalia.romanova@unicaen.fr

Rayan Ziane
LLL, University of Orléans
rayan.ziane@univ-orleans.fr

Abstract

Performance of syntactic parsers is reduced for longer sentences. While some of this reduction can be explained by the tendency of longer sentences to be more syntactically complex as well as the increase of candidate governor number, some of it is due to longer sentences being more challenging to encode. This is especially relevant for low-resource scenarios such as parsing of written sources in historical languages (e.g. medieval and early-modern European languages), in particular legal texts, where sentences can be very long whereas the amount of training material remains limited. In this paper, we present a new method for explicitly using the arc length information in order to bias the scores produced by a graph-based parser. With a series of experiments on Norman and Gascon data, in which we divide the test data according to sentence length, we show that indeed explicit length coding is beneficial to retain parsing performance for longer sentences.

Introduction

As a rule, when syntactic parsing models are evaluated, the general Labeled Attachment Score (LAS) is calculated without taking into account performance for different sentence lengths. The LAS assesses the performance of a parser by considering the number of words that have been assigned both the correct syntactic head and the correct label (Nivre and Fang, 2017).

For *treebanks* of low-resourced languages or language varieties (e.g. medieval languages) where small amounts of annotated data exist, precision of the annotation is paramount for syntactic research and constitution of reliable training corpora; manual revision of automatic parsing is therefore required. When correcting automatic annotation of historical French texts (e.g. Old, Middle and sixteenth-century French), it was empirically observed by the authors that the performance of parsers is significantly reduced on longer sentences; we elaborate on this in the next paragraph. Some errors appear counter-intuitive, e.g. distance between the token and its head, the direction of the arc, especially in the case of nominal dependents such as *det* and *case*. Thus, the longer the sentence, the higher the likelihood that, for example, an article would be attached to a noun several tokens to the left when its actual head is the next token to the right.

To give an example, we tested a model trained on one type data on a similar target corpus. First, we trained a dependency parser, BertForDeprel (Guiller, 2020), an open source model, based on Dozat and Manning (Dozat et al., 2017) architecture. For the embedding layer, we used XLM-RoBERTa multilingual model. This parser was trained on Old French (UD_Old_French–PROFITEROLE@2.16 corpus (Prévost et al., 2024) and achieved a global LAS of 89% and UAS of 92%. To evaluate its performance and assess its sensitivity to sentence length, we used a small sample from the 13th-century chronicle *Histoire ancienne jusqu'à César* (HaC-Sample). The sentences were selected from the digital edition of the chapter 'Rome II' from the manuscript BnF fr. 20125 (Morcos et al., 2021) and manually annotated and validated. The language and the genre of the target corpus as well as the principles of sentence segmentation were the same as in the training corpus.

The HaC-Sample dataset was divided into ten groups based on sentence length to examine the influence of length on parsing performance. The result of the parsing presented in graph 1 shows that the parser has better performance on medium-length sentences. Performance decreases for shorter and longer sentences, however. This drop may be explained by the lack of syntactic structure for the shorter sentences and the rise of syntactic complexity in the longer ones.

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 102–110, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-011

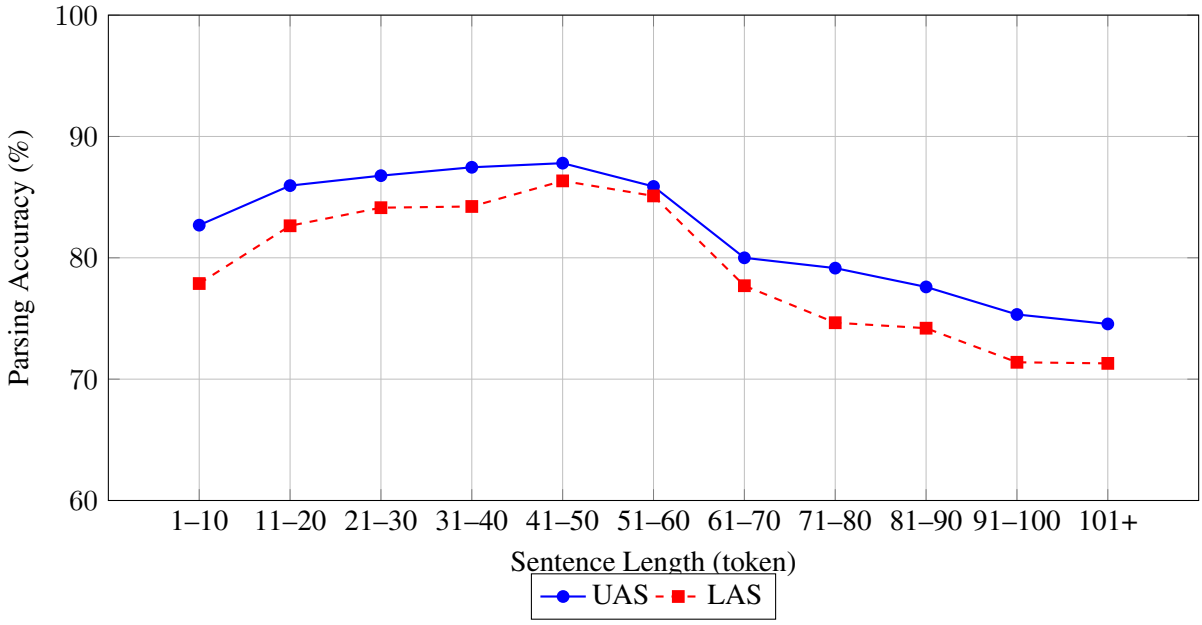


Figure 1: LAS and UAS parsing performance by sentence length group on HaC-Sample.

Related Work

The impact of sentence length on the accuracy of dependency parsers has been highlighted in different studies. (Gulordava and Merlo, 2016) conducted multilingual evaluation using artificially-generated *treebanks*, demonstrating that word variability and longer dependencies significantly degrade parser performance independently of the language or the *treebank* size. (Anderson and Gómez-Rodríguez, 2020) introduced the concept of Inherent Dependency Displacement Bias, which shows the bias of the parsing algorithm in handling the distance and direction of syntactic arcs. The authors found a strong correlation between sentence length and parsing accuracy. (Ajusha and Ajees, 2024) investigated the challenges in Malayalam, southern Dravidian language, where they found that the parsers struggled on long distance dependencies. These studies emphasize sentence length as a linguistic factor affecting parser performance.

At the same time, to address the problem of improving parser accuracy, previous researchers focused on the incorporation of morphosyntactic features into parsing models. (Nguyen and Verspoor, 2018) showed that high-quality PoS (Part-of-speech) tagging can improve parsing accuracy in biomedical texts. In the context of low-resource languages, (Anderson et al., 2021) demonstrated that predicted Universal PoS tags can significantly enhance the parsing, even in the absence of gold tags. (Ziane and Romanova, 2024) explored pre-finetuning of a parser with PoS tagging, thus biasing the parser's behaviour to improve its learning algorithm. On the other hand, (Altıntaş and Tantuğ, 2023) 's approach focused on architectural enhancement of the parser. By injecting global sentence embedding and CNN-based local context features into the arc scoring layer, this method empowered the graph based parser.

In this work, we aim to address specifically the problem of sentence length in dependency parsing in the context of low-resource historical texts. The method is based on the idea of biasing the scores produced by the parser to reflect the arc length information in 16th-century Norman (Guernsey) and medieval Gascon *treebanks*.

Corpus

For the experiments described below we used two of the corpora of the latest release of the Universal Dependencies collection (@2.16 released 15 May 2025) with the longest average sentence length. We selected corpora of medieval Romance languages, both belonging to the legal genre.

The "Norman" corpus, UD_French-ALTS@2.16 (42,832 tokens; 1,269 sentences) (Romanova et al., 2025) is a corpus of court proceedings from the island of Guernsey (1563-1569) transcribed from the manuscript of the register *Crime I* preserved at Guernsey Greffe (court archives of the island).[1]

[1] https://github.com/UniversalDependencies/UD_French-ALTS.

A legal text, it contains many long formulaic sentences, complex sentences and lists. The register is written in French, the language of the court of justice on Guernsey in the sixteenth century. However, since the island was under the British rule, the scribes were not obliged to follow the ordinances of Villers-Cotterêts (1539), which imposed the use of standard French in the official documents of the Kingdom of France. The language of *Crime I* therefore exhibits numerous dialectal (Norman) features such spellings and morphological characteristics of Northern French dialects. Like Old and Middle French, it is characterised by high degree of variation of forms and word orders. The average sentence length for the dev part of the corpus is 40,22 tokens, for the test part 36,16 tokens.

The "Gascon" corpus, UD_Occitan-CorAG@2.16 (1,094 sentences; 37,585 tokens) (Francioni et al., 2025) contains two medieval (one thirteenth-century and one fifteenth-century) legal manuals.[2] Gascon is a dialect of Old Occitan. This is the first available UD-annotated corpus in any medieval variety of Occitan. The average sentence length for the dev part of the corpus is 29,13 tokens, for the test part 35,24 tokens.

Both corpora were annotated in Parts-of-Speech (PoS), syntactic functions and heads in the Universal Dependencies (UD) framework (de Marneffe et al., 2021) by progressively adapting a model for Old and Middle French based on Profiterole corpus (Prévost et al., 2024) using ArboratorGrew software (Guibon et al., 2020) and built-in BertForDeprel parser (Guiller, 2020). Automatic annotation was manually checked.

The data for the experiments described below was split into three groups 70% train, 20% test and 10% dev, then the test group was divided into ten groups by length of the sentence.

Methodology and results

Length-Biased Graph Parser

As mentioned above, graph based parsers suffer a drop in quality as sentence length increases. There are several compounding factors leading to this. Longer sentences tend to be more syntactically complex with several levels of subordinated clauses for example. Moreover, longer sequences tend to be harder to handle for recurrent neural networks. The number of potential governors simply increases with the length of the sentence and, whereas the number of valid dependencies increases linearly with the length of the sentence, the number of invalid dependencies increases quadratically with it.

However, we noticed that even very simple errors appear in very long sentences, such as determiners attaching to nouns tens of tokens away. The most likely explanation for this is the difficulty for the biaffine layer to use the relative distance between tokens in order to reduce the score of unlikely long distance dependencies. We therefore propose to add a biasing mechanism beside the biaffine layer to help the parser avoid invalid long dependencies.

The basic idea is to add a multiplicative bias to the biaffine layer in order to boost or diminish the scores of arcs based on their signed distances. However, since different syntactic relations can have very different lengths and directions, we need to add extra information about each arc.

Therefore, we hypothesized that learning a bias for each triplet of governor PoS-tag, dependent PoS-tag and signed length of the arc should help the parser select better heads for words that have very local relations such as determiners or adjectives.

The biases for the selection of the relation label are based on the pair of governor-dependent PoS-tags and the dependency relation.

We also experimented with biasing over the signed length of the relation, however the results did not seem to improve. This may be due to the small size of our training data, and maybe with a bigger training set results would become more interesting.

In order to easily experiment with different biasing methods, we worked with our own re-implementation of (Dozat et al., 2017)'s graph-parser.[3]

We now describe the arc's length biasing mechanism. Given a sentence x of length n, the base parser produces the arc score matrix $S \in \mathbb{R}^{n \times n}$ and the relation label score tensor $R \in \mathbb{R}^{n \times n \times r}$, where r is the number of dependency relation labels.

Let $P\{0,1\}^{n \times p}$ be the matrix of one-hot encoded PoS-tags corresponding to x, where p is the number of PoS-tags types. Let l be the maximum arc length we want to consider, every longer edges will be cast to $\pm l$. Then, let $D\{0,1\}^{n \times n \times (2l+1)}$

[2] https://github.com/UniversalDependencies/UD_Occitan-CorAG.

[3] Code can be downloaded at https://github.com/MathieuDehouck/LowRes-Parser.

be the tensor encoding signed edge lengths in a one-hot manner:

$$D_{ijk} = \begin{cases} 1 \text{ if } k = \max(\min(i-j, l), -l) + l \ , \\ 0 \text{ otherwise.} \end{cases}$$

The arc biases B^{arc} and relation biases B^{rel} are then computed as follows:

$$B^{arc} = (P \otimes P^T \otimes D)\Theta^{arc},$$

$$B^{rel} = (P \otimes P^T \otimes \mathbf{1}^r)\Theta^{rel},$$

where $\Theta^{arc} \in \mathbb{R}^{p \times p \times (2l+1)}$ and $\Theta^{rel} \in \mathbb{R}^{p \times p \times r}$ are learnable parameters, $\mathbf{1}^r$ is the vector of length r where each entry is a 1, and where $\otimes$ notes the Kronecker product.

The final scores are then $S \odot B^{arc}$ and $R \odot B^{rel}$, where $\odot$ notes the Hadamard product.

Since we chose to work with multiplicative biases, values bigger than 1 are positive biases and values below are negative biases.

Experiments

In order to test the capacity of arc biasing to increase parsers' ability to handle longer sentences, we experimented with four parsing scenarios.

We trained parsers using only word embeddings taken from an encoder large language model as a simple baseline (Embedding). We then trained parsers using concatenated word and PoS-tag embeddings (+ PoS). This is a stronger baseline. Then, we trained parsers that only bias the arcs' scores based on their lengths, but do not bias the relations' scores (+ Arc bias). This is equivalent to setting Θ^{rel} to 1 and not updating it. Finally, we trained parsers that bias both arcs' and relations' scores as described in previous section (+ Rel bias).

For the embedding layer, we use the BERTrade language model (Grobol et al., 2022) trained specifically for Medieval French for both Norman and Gascon text since the only natively Occitan encoder we found had a too short context length to represent our sentences. While Occitan and Medieval French are closely related languages, this is obviously a sub-optimal situation and will explain the relative quality of the Gascon parser. When a word is split into multiple tokens by the encoder's tokenizer, we only keep the representation of the first token.

The PoS-tags embeddings are learned alongside the rest of the parser's parameters. In order to see the influence of the biases on the parsing quality of longer sentences we split the Norman and Gascon test sets into subsets of similarly sized sentences. The detail of the splits are reported in table 1 for the Norman data and in table 2 for the Gascon data.

Sentences length	Number of sentences	Number of tokens
5 - 10	24	195
11 - 20	121	1973
21 - 30	82	2008
31 - 40	49	1731
41 - 50	25	1140
51 - 60	22	1196
61 - 80	19	1291
81 - 137	5	505
All	347	6673

Table 1: Sizes of the Norman test subsets based on sentence length.

Results are thus reported for the whole test set and for each length based subset. They are averaged over 5 runs initialized with different random seed.

Results

Results for the Norman parsing experiment are reported in table 3 and those for the Gascon experiment are reported in table 4.

As we can see from table 3, adding PoS-tags embeddings already improves a lot the parsing capacity of the models.

However, while the models with and without arc and relation biasing are on par for sentences of length up to 60 tokens when they can use PoS-tags, for longer sentences, the biased models have a clear advantage. For sentences of length between 61 and 80 tokens, biased parsers show a 1.25 unlabeled attachment score point (UAS) increase and a 1.20 labeled attachment score point (LAS) increase. For sentences beyond 81 tokens, it reaches 3.09 UAS and 2.89 LAS points increase.

Thus arc biasing indeed seems to help maintaining a better parsing accuracy for longer sentences.

Table 4 gives a very similar picture.

However, since the parsers are of an overall lower quality due to the mismatch between the pre-training language of the encoder and the language it is applied to, the effects are even more marked. Here, even for reasonably sized sentences (less than 60 tokens) the biased models already show an advantage over the non biased ones.

Sentences length		Number of sentences	Number of tokens
5	10	27	227
11	20	88	1347
21	30	52	1316
31	40	32	1124
41	50	26	1173
51	60	16	886
61	70	6	388
71	80	14	1063
81	90	6	520
91	100	4	371
101	125	5	535
126	150	3	391
151	175	2	304
176	200	2	355
	All	285	10007

Table 2: Sizes of the Gascon test subsets based on sentence length.

We go from +0.38 UAS point for sentences of lengths between 21 and 30 tokens, to +2.49 UAS for sentences between 71 and 80 tokens, to up to +9.67 UAS for sentences of lengths between 151 and 175 tokens.

Figure 2 and figure 3 represent the evolution of the percentage of UAS error reduction for different models with respect to the baseline, embedding only, parser for the ALTS Norman and the CorAG Gascon test sets respectively.

We see that on both figures, the curves representing the UAS error reduction for the two arcs' length-biased models (with and without relation label biasing) stay close together around the 40 % line, while the curve corresponding to the unbiased model starts departing from the other two for longer sentences (more than 60 tokens) getting below the 30 % line.

It is also interesting to note that despite the Norman and Gascon models having very different performances, the error reduction of the PoS-tag informed and the arcs' length-biased models are surprisingly similar.

However, we do not know if it is a meaningful phenomenon or if it is just a coincidence and thus it needs further investigation.

These results indeed seem to support the ability of arc and relation biasing to improve accuracy of longer sentences parsing.

This is true even with respect to models that use

Group test set	Parser	UAS	LAS
5 - 10	Embedding	90.15	83.49
	+ PoS	**97.85**	**94.46**
	+ Arc bias	97.64	**94.46**
	+ Rel bias	97.23	94.05
11 - 20	Embedding	92.60	89.72
	+ PoS	95.43	93.75
	+ Arc bias	95.45	93.75
	+ Rel bias	**95.57**	**94.01**
21 - 30	Embedding	89.28	85.64
	+ PoS	92.30	90.13
	+ Arc bias	92.59	90.54
	+ Rel bias	**92.87**	**90.98**
31 - 40	Embedding	88.39	84.84
	+ PoS	**93.19**	91.18
	+ Arc bias	92.92	91.00
	+ Rel bias	92.96	**91.22**
41 - 50	Embedding	86.79	83.60
	+ PoS	**91.68**	**89.72**
	+ Arc bias	91.54	89.70
	+ Rel bias	91.61	89.61
51 - 60	Embedding	86.76	83.70
	+ PoS	91.69	90.27
	+ Arc bias	**91.99**	**90.72**
	+ Rel bias	91.62	90.27
61 - 80	Embedding	87.02	84.32
	+ PoS	89.70	88.23
	+ Arc bias	90.84	89.14
	+ Rel bias	**90.95**	**89.43**
81 - 137	Embedding	83.92	80.55
	+ PoS	87.60	86.26
	+ Arc bias	90.38	88.36
	+ Rel bias	**90.69**	**89.15**
All	Embedding	88.65	85.37
	+ PoS	92.46	90.64
	+ Arc bias	92.78	90.96
	+ Rel bias	**92.85**	**91.14**

Table 3: Results of the experiments on Norman data. Performance metrics (UAS and LAS) for different test subsets, grouped by sentence length, across four parser variants: word **Embedding** alone, **+ PoS** tags embedding, **+ Arc bias**, and **+Rel bias**.

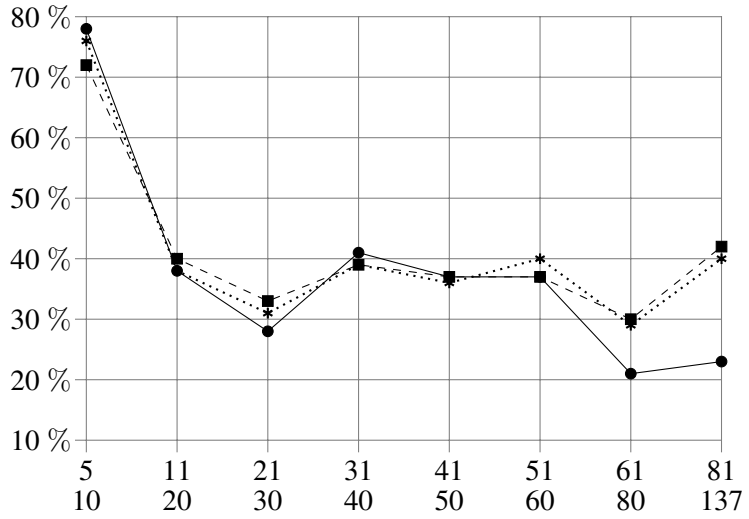


Figure 2: Graphical representation of the error reduction with respect to the baseline, embedding only, model UAS score for each sentence length-based ALTS Norman test subset. Bullets (•) represent the **+ PoS** model. Asterisks (∗) represent the **+ Arc bias** model. Squares (■) represent the **+ Rel bias** model.

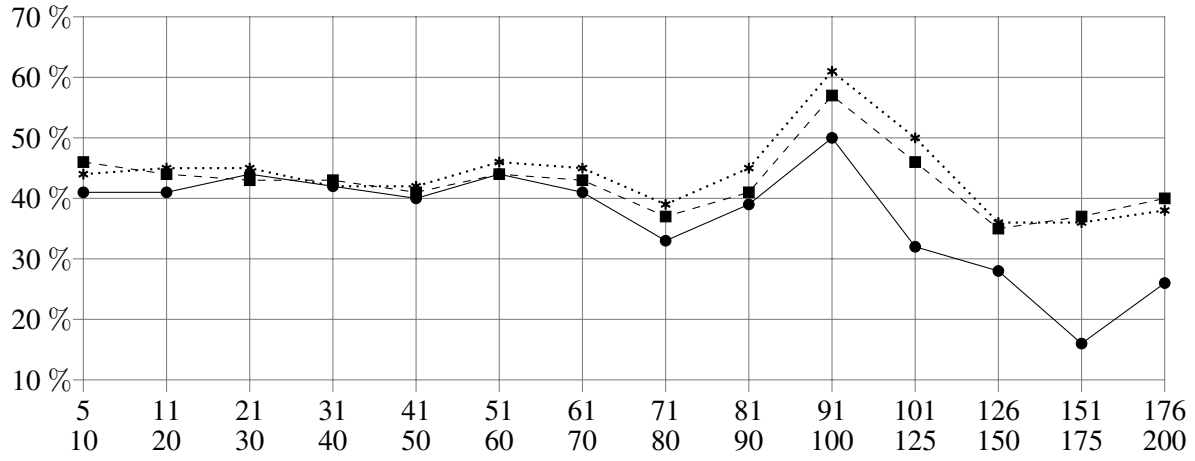


Figure 3: Graphical representation of the error reduction with respect to the baseline, embedding only, model UAS score for each sentence length-based CorAG Gascon test subset. Bullets (•) represent the **+ PoS** model. Asterisks (∗) represent the **+ Arc bias** model. Squares (■) represent the **+ Rel bias** model.

Group test set	Parser	UAS	LAS
5 - 10	Embedding	66.61	57.53
	+ PoS	80.26	76.30
	+ Arc bias	81.23	76.12
	+ Rel bias	**82.11**	**77.36**
11 - 20	Embedding	65.66	55.99
	+ PoS	79.82	75.77
	+ Arc bias	**81.14**	**76.60**
	+ Rel bias	80.88	76.38
21 - 30	Embedding	67.57	57.95
	+ PoS	81.73	**77.63**
	+ Arc bias	**82.11**	77.31
	+ Rel bias	81.63	77.25
31 - 40	Embedding	65.36	56.94
	+ PoS	79.80	75.21
	+ Arc bias	79.95	75.27
	+ Rel bias	**80.32**	**75.53**
41 - 50	Embedding	60.90	52.23
	+ PoS	76.47	73.08
	+ Arc bias	**77.24**	**73.59**
	+ Rel bias	76.83	73.23
51 - 60	Embedding	57.47	47.88
	+ PoS	76.00	71.74
	+ Arc bias	**77.20**	**72.37**
	+ Rel bias	76.32	71.38
61 - 70	Embedding	60.31	53.40
	+ PoS	76.49	73.20
	+ Arc bias	**78.30**	**74.85**
	+ Rel bias	77.32	74.18
71 - 80	Embedding	59.12	48.62
	+ PoS	72.47	67.70
	+ Arc bias	**74.96**	**69.80**
	+ Rel bias	74.43	69.13
81 - 90	Embedding	59.00	47.77
	+ PoS	74.96	71.31
	+ Arc bias	**77.42**	**72.58**
	+ Rel bias	75.88	71.58
91 - 100	Embedding	62.26	52.56
	+ PoS	81.08	76.33
	+ Arc bias	**85.39**	**80.97**
	+ Rel bias	83.83	80.11
101 - 125	Embedding	53.83	43.63
	+ PoS	68.75	64.45
	+ Arc bias	**77.05**	**71.78**
	+ Rel bias	75.18	70.88
126 - 150	Embedding	56.21	48.49
	+ PoS	68.59	65.17
	+ Arc bias	**71.76**	68.34
	+ Rel bias	71.61	**68.59**
151 - 175	Embedding	53.03	49.21
	+ PoS	60.33	58.68
	+ Arc bias	69.87	67.43
	+ Rel bias	**70.20**	**67.89**
176 - 200	Embedding	49.75	36.85
	+ PoS	62.87	58.31
	+ Arc bias	68.79	**63.38**
	+ Rel bias	**69.69**	62.59
All	Embedding	61.30	51.97
	+ PoS	76.01	71.97
	+ Arc bias	**78.17**	**73.65**
	+ Rel bias	77.71	73.31

Table 4: Results of the experiments on Gascon data. Performance metrics (UAS and LAS) for different test subsets, grouped by sentence length, across four parser variants: word **Embedding** alone, **+ PoS** tags embedding, **+ Arc bias**, and **+Rel bias**.

the same overall input features (word embeddings and PoS-tags) suggesting that a proper encoding of arcs' length is beneficial for longer sentences.

Discussion

In addition to increasing the parser's accuracy, PoS-tag based biases are easily interpretable by humans. Since these are multiplicative biases, a value above 1 is a positive bias and a value smaller than 1 is a negative bias. Figure 4 represents the value of length biases for a selection of pairs of PoS-tags. Biases corresponding to the NOUN-DET pairs are represented by black bullets.

The positions -1 and -2 are the only ones with a positive bias (1.23 and 1.20 respectively). This aligns perfectly with the fact that, in Medieval and early Modern French, determiners come right before their nouns, save a potential adjectival phrase. The biggest negative bias appears at position -5 with a value of 79.

There are a number of constructions where a determiner appears five tokens before a noun while not being governed by this very noun. Here we give just a few examples with English glosses below.

Le	*sabmedy*	*.xe.*	*jour*	*du*	*moes*
the	*saturday*	*10th*	*day*	*of_the*	*month*

de	*l'*	*uylle*	,	*du*	*pain*
of	*the*	*oil*	,	*of_the*	*bread*

son	*filz*	*venoient*	*en*	*sa*	*maison*
their	*son*	*came*	*in*	*their*	*house*

Overall there are 346 such instances in the training data and not a single one where a determiner would attach to a noun four tokens away.

On the same figure, we represent the biases learnt for the VERB-PRON pairs with crosses. Here we see that contrary to the NOUN-DET arcs, there are positive biases corresponding to both left and right arcs.

This too, aligns well with Medieval and Modern French grammar. In Modern French, pronouns tend to appear before their verb, but inversion is common in orders (direct and indirect object pronouns follow imperative verbs) and questions as well as a way to introduce reported speech.

Furthermore, pronouns were more mobile in Medieval French.

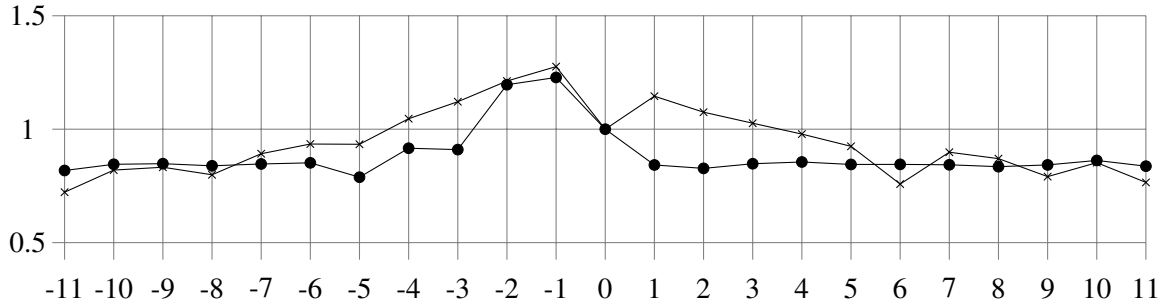


Figure 4: Samples of arc biases learnt on the ALTS Norman treebank. Bullets (•) represent the NOUN-DET arcs and crosses (×) represent the VERB-PRON arcs.

Future Work

Arc and relation label biasing can be easily applied to any parser that gives access to the score tensors on top of the actual structure prediction. So it would be interesting to see how this can be used in order to do a very light weight form of fine-tuning of already trained models.

Indeed some graph-based parsers actually take raw text as input and predict PoS-tags at the same time as the arcs' scores, so we would need to wait for this PoS-tag prediction in order to bias the arcs' scores. Furthermore, we still need to perform a more complete investigation of the learnt biases and we also intend to investigate their usability for transfer and language comparison, since they encode grammatical rules in a very simple format.

Eventually, since taking inspiration from the models that predict PoS-tags and dependency scores at the same time, in a multi-task learning spirit, teaching parsers to predict the signed length of an arc based on its governor's and dependent's representations could help them avoiding invalid long dependencies better, maybe even without having to bias.

Conclusion

We have presented first experiments towards tackling reduced performance of syntactic parsing in longer sentences: directly biasing the scores of the arcs in order to reflect their length. This is especially relevant when working with historical written texts, particularly of the administrative and legal types. These experiments point to the necessity to learn the length and direction of arc between the syntactic function and its head and the direction of the arc from the training corpus. We have seen that the experiments presented allow beginning to improve the scores.

Limitations

A more detailed analysis of these trends is needed, including a detailed error analysis, evaluating statistic significance of the results, testing on a wider variety of corpora and using bootstrapping scenarios. We believe that in order to improve performances on longer sentences a hierarchical approach to parsing may be beneficial.

Acknowledgements

This work was partially funded by the AUTOMATED project (2023-2025, University of Caen, France, PI Professor Pierre Larrivée; funded by Normandy Region). The staff at the Guernsey Greffe archives and the Guernsey Museum & Art Gallery gave us access to the manuscript and digital images of the Crime I register that were used for the Guernsey Norman corpus. Our thanks go to student transcribers who collaborated on the transcription. We thank Barbara Francioni who annotated the Gascon CorAG corpus.

References

P.V. Ajusha and A.P. Ajees. 2024. Morphological and syntactic challenges in malayalam: A dependency parsing perspective. *SSRG International Journal of Electrical and Electronics Engineering*, 11(12):375–385.

Mücahit Altıntaş and A. Cüneyd Tantuğ. 2023. Improving the performance of graph based dependency parsing by guiding bi-affine layer with augmented global and local features. *Intelligent Systems with Applications*, 18:200190.

Mark Anderson, Mathieu Dehouck, and Carlos Gómez-Rodríguez. 2021. A falta de pan, buenas son tortas: The efficacy of predicted UPOS tags for low resource UD parsing. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT*

2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021), pages 78–83, Online. Association for Computational Linguistics.

Mark Anderson and Carlos Gómez-Rodríguez. 2020. Inherent dependency displacement bias of transition-based algorithms. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5147–5155, Marseille, France. European Language Resources Association.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the conll 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Barbara Francioni, Natasha Romanova, Rayan Ziane, Khensa Daoudi, and Pierre Larrivée. 2025. Corag: Corpus d'ancien gascon. https://github.com/UniversalDependencies/UD_Occitan-CorAG.

Loïc Grobol, Mathilde Regnault, Pedro Ortiz Suarez, Benoît Sagot, Laurent Romary, and Benoit Crabbé. 2022. BERTrade: Using contextual embeddings to parse Old French. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1104–1113, Marseille, France. European Language Resources Association.

Gaël Guibon, Marine Courtin, Kim Gerdes, and Bruno Guillaume. 2020. When collaborative treebank curation meets graph grammars. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5293–5302, Marseille, France. European Language Resources Association.

Kirian Guiller. 2020. Analyse syntaxique automatique du pidgin-créole du nigeria à l'aide d'un transformer (bert): Méthodes et résultats. Master's thesis, Sorbonne Nouvelle.

Kristina Gulordava and Paola Merlo. 2016. Multilingual dependency parsing evaluation: a large-scale analysis of word order properties using artificial data. *Transactions of the Association for Computational Linguistics*, 4:343–356.

Hannah Morcos, Simon Gaunt, Simone Ventura, Maria Teresa Rachetta, Henry Ravenhall, Natasha Romanova, Geoffroy Noël, Paul Caton, Ginestra Ferraro, and Marcus Husar. 2021. The histoire ancienne jusqu'à césar: A digital edition; bnf, fr20125 (interpretive edition): Eneas (6) and assyrian kings (6bis), rome i (7), rome ii (10) and caesar (11). https://http://www.tvof.ac.uk/textviewer/.

Dat Quoc Nguyen and Karin Verspoor. 2018. From POS tagging to dependency parsing for biomedical event extraction. *CoRR*, abs/1808.03731.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 86–95, Gothenburg, Sweden. Association for Computational Linguistics.

Sophie Prévost, Loïc Grobol, Mathieu Dehouck, Alexei Lavrentiev, and Serge Heiden. 2024. Profiterole : un corpus morpho-syntaxique et syntaxique de français médiéval. *Corpus*, 25:15 pp.

Natasha Romanova, Rayan Ziane, and Khensa Daoudi. 2025. Alts: Automated sixteenth-century corpus. https://github.com/UniversalDependencies/UD_French-ALTS.

Rayan Ziane and Natasha Romanova. 2024. Pistes pour l'optimisation de modèles de parsing syntaxique. In *LIFT 2 – 2024: Journées de lancement*, Orléans, France. LIFT (Linguistique Informatique, Formelle et de Terrain).

Evaluating LLM Capabilities in Low-Resource Contexts: A Case Study of Persian Linguistic and Cultural Tasks

Jasmin Heierli, Rebecca Bahar Ganjineh, Elena Gavagnin
Zurich University of Applied Sciences, Winterthur, Switzerland
heej@zhaw.ch, ganjireb@students.zhaw.ch, gava@zhaw.ch

Abstract

We evaluate four representative large language models, namely GPT-4o, Gemini, Llama, and DeepSeek on a suite of linguistic and cultural tasks in Persian, covering grammar, paraphrasing, inference, translation, factual recall, analogical reasoning, and a Hofstede-based cultural probe under direct and role-based prompts. Our findings reveal consistent performance declines, alongside systematic misalignment with Iranian cultural norms. Role-based prompting yields modest improvements but does not fully restore cultural fidelity. We conclude that advancing truly multilingual models demands richer Persian resources, targeted adaptation, and evaluation frameworks that jointly assess fluency and cultural alignment.

1 Introduction

Despite rapid advances in large language models (LLMs), their multilingual capabilities remain deeply uneven. For dominant languages such as English, modern LLMs exhibit high performance in linguistic fluency, factual accuracy, and sociocultural alignment (Jin et al., 2024; Lai et al., 2023a). However, for low-resource languages like Persian (Farsi), model outputs often degrade grammatically, semantically, and culturally, leading to concrete societal risks of digital invisibility or misrepresentation for over 90 million native speakers worldwide (Eberhard et al., 2025).

Persian occupies a unique linguistic and cultural position, spoken across Iran, Afghanistan (Dari), and Tajikistan (Tajik), with rich Indo-European and Arabic influences and substantial regional variation. Additionally, it has a distinct character set unshared with other languages that have at least the Latin character set as common ground. Yet it remains vastly underrepresented in pretraining corpora for LLMs: in the Common Crawl dataset—one of the largest public web corpora—Persian constitutes less than 0.1 % of content versus over 45 % for English (Common Crawl, 2025). Existing LLM evaluations and audit tools are predominantly Anglocentric, overlooking language-specific disparities and fairness considerations in Persian.

To address this oversight, we present a joint empirical analysis of linguistic competence and cultural sensitivity in state-of-the-art LLMs operating in Persian (GPT-4o (OpenAI, 2024), Gemini (Google DeepMind, 2024), Llama (Meta AI, 2025), DeepSeek (DeepSeek-AI et al., 2025)). Building on and extending established cultural probing methods (Masoud et al., 2025; Moosavi Monazzah et al., 2025) alongside linguistic diagnostics benchmarks (Atox and Clark, 2024; Abaskohi et al., 2024), we systematically expose where—and why—these models fail on Persian tasks, from subtle grammatical nuances to culturally grounded references.

We make three key, novel and unique contributions to the study of LLM performance in Persian.

- **Comprehensive evaluation suite.** We assemble four representative LLMs and test them on:
  - *Linguistic tasks*: spelling correction and paraphrasing,
  - *World-knowledge QA*: factual recall and analogical reasoning,
  - *Cultural probing*: Hofstede-style role simulation in Persian.
- **Fairness-aware lens.** We demonstrate that simple "act-as" prompts fail to elicit Persian cultural perspectives, and that translation-based interventions yield gains only for bilingual users.
- **Cultural prompting insights.** We provide quantitative evidence of Western bias in Persian outputs, highlighting systemic misalignment rather than deliberate prejudice.

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 111–120, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-012

By diagnosing these failures, we reveal structural biases in contemporary LLMs and call for inclusive evaluation methods and culturally-aware model tuning for low-resource languages.

2 Related Work

Several benchmarks have emerged to evaluate LLM performance on Persian tasks. The ParsiNLU suite (Daniel Khashabi, 2020) provides multiple-choice QA, paraphrase, natural language inference (NLI), and translation splits drawn from Google autocomplete, forums, and exam questions. FAspell (Barari and QasemiZadeh, 2005) offers real-world spelling errors collected from students and professional typists. Both were mainly developed to benchmark task-specific pre-trained and/or fine-tuned machine learning models. More recent work by Abaskohi (Abaskohi et al., 2024) benchmarks GPT-3.5-turbo and GPT-4 on ParsiNLU, showing gains when inputs are translated into English but underscoring persistent deficits in direct Persian prompting.

In-context learning and prompt design are critical for cross-lingual transfer. Brown et al. (2020) introduced zero- and few-shot prompting, which has since been adapted to multilingual settings (Atox and Clark, 2024). AlKhamissi et al. (2024) showed that persona-based prompts—e.g. "answer as a respondent from Egypt"—can markedly shift outputs and improve alignment with local survey data for languages like Arabic and English. Likewise, Masoud et al. (2025) applied explicit role priming to probe cultural dimensions. However, these studies remain anchored to languages with relatively rich pretraining resources and depend on overt "act-as" formulations or direct translation. PERCUL, by contrast, tackles Persian—a genuinely low-resource language with its own script and morphology—eschewing explicit role prompts in favor of embedding cultural concepts within short, human-curated narratives and assessing implicit comprehension via multiple-choice questions (Moosavi Monazzah et al., 2025).

Broad evaluation frameworks such as HELM (Holistic Evaluation of Language Models) highlight major coverage and metric gaps for underrepresented languages (Liang et al., 2022). Building on this, Kharchenko et al. (2024) applied Hofstede's cultural dimensions across 36 countries—showing that even well-resourced languages suffer inconsistent cultural fidelity in LLM outputs. Focusing on Persian, PerCul (Moosavi Monazzah et al., 2025) uncovers substantial misalignment in cultural references, while the Cultural Alignment Test (CAT) of Masoud et al. (2025) quantitatively demonstrates a persistent Western bias in role-based prompts. Together, these works underscore the necessity of a fairness-aware lens that jointly evaluates linguistic competence and cultural sensitivity in low-resource settings like Persian.

3 Methodology

In this section we describe the four state-of-the-art LLMs we evaluate, the tasks and datasets we employ, our prompt engineering strategies, and the metrics used to quantify performance. You can find our prompts and the subsets of the datasets that we have used on GitHub.[1]

3.1 Model Selection

We evaluate four representative multilingual LLMs, chosen to span closed- and open-source, commercial and community-driven models as shown in Table 1.

All models were accessed via their respective APIs using default temperature and top-p settings, except for Llama 3.3 which was accessed via DeepInfra due to hardware limitations. While all of the models tested are multilingual, Llama 3.3 is the only model that does not officially support Persian. However, the model card states that other languages may still work, as the data was likely included during training (Meta AI, 2025). This might seem like an improper comparison at first, but we were interested in whether official support makes a difference, as we assume the underlying training data to be broadly similar.

3.2 Tasks and Datasets

We cover two major task families: one to probe linguistic competence as well as factual knowledge, and one to probe cultural sensitivity in Persian. Each linguistic task was randomly sampled

[1] https://github.com/zhaw-iwi/LowResNLP-Evaluating-LLM-Capabilities-for-Persian

[1] https://openai.com/index/gpt-4o-system-card/

[2] https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/

[3] https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md

[4] https://api-docs.deepseek.com/news/news1226

Model	Developer	Open?	Persian support
GPT-4o[1]	OpenAI	Closed	Yes
Gemini 2.0 Flash[2]	Google DeepMind	Closed	Yes
Llama 3.3 70B Instruct[3]	Meta AI	Open	No (official)
DeepSeek-V3[4]	DeepSeek-AI	Open	Unspecified

Table 1: High-level comparison of model properties used for our experiments (OpenAI, 2024; Google DeepMind, 2024; Meta AI, 2025; DeepSeek-AI et al., 2025).

for 350 items to have comparable task size and a manageable runtime per experiment. Where possible and applicable we also report ignorance of task instructions separately, as we think that a lack of instruction-adherence is not the same as a wrong answer.

Linguistic tasks

- *Spelling correction.* We sample 350 word-pairs ("misspelled", "corrected") from the FAspell corpus (Barari and QasemiZadeh, 2005). We used the part of the dataset that contains real-life collected human-made errors by elementary school children and professional typists. The models are prompted in Persian and English to output only the corrected form; we compute exact-match *Accuracy*.

- *Paraphrase classification.* From the ParsiNLU paraphrase dataset (Daniel Khashabi, 2020), we randomly sampled 350 pairs. The data has been collected from Google auto-complete and Persian forums and was annotated by native speakers (Daniel Khashabi, 2020). Using a 2-shot prompt in Persian or English, our probed models label pairs as paraphrases (1) or non-paraphrases (0). The expected output is a single digit (0/1) with no explanation; extra text is treated as a format failure (see §3.4). We report *Accuracy*.

- *Entailment classification.* We sample 350 examples from the ParsiNLU NLI split (Daniel Khashabi, 2020). Models receive premise–hypothesis pairs and decide "entailment", "neutral", or "contradiction" in Persian or English few-shot. Expected output is a single token from {e, n, c}; any other string is considered a format failure. we report *Accuracy*.

- *Machine translation.* We sample 350 pairs for English→Persian from the ParsiNLU translation split (Daniel Khashabi, 2020; Kashefi, 2018). The data was collected from human-made translations. Models translate few-shot and we evaluate with *BLEU* score.

- *Factual recall.* We sample 350 multiple-choice questions from the ParsiNLU "common_knowledge" partition. The questions have been mostly taken from college entry exams (Daniel Khashabi, 2020). Models were prompted to choose between 3–4 options and we report *Accuracy*.

- *Analogical reasoning.* We sample 350 questions from the ParsiNLU "literature" partition, that mostly require analogical reasoning, similar to (Atox and Clark, 2024). Items are multiple-choice (3–4 options). The model is instructed to output only the option letter (A/B/C/D). We again report *Accuracy*.

Class counts for all classification tasks are reported in Table 4 in Appendix A. NLI ("e"/"n"/"c"), factual recall (1–4), and logical reasoning (1–4) are near-balanced, while paraphrase is moderately skewed toward non-paraphrases (≈62/38).

3.3 Cultural Probing

To expose how well our models internalize deeply held cultural values, we draw on Hofstede's well-validated VSM13 framework, which distills cross-national survey data into six interpretable dimensions: Power Distance (PDI), Individualism vs. Collectivism (IDV), Masculinity vs. Femininity (MAS), Uncertainty Avoidance (UAI), Long-Term vs. Short-Term Orientation (LTO), and Indulgence vs. Restraint (IVR) . Prior work demonstrates that these latent variables are encoded in language use and can be probed via structured questionnaires .

We administer the 24 VSM13 items in Persian using their standard 5-point Likert format. For each model and scenario we collect 100 independent completions and compute per-item means and standard deviations, as follows:

- **VSM13 indices**: the six dimension scores via Hofstede's original formulas,
- **Response variability**: the per-item standard deviation across 100 trials,
- **Cross-scenario stability.** *Definition.* Let $r_{i,s} \in [1, 5]$ denote the mean Likert response for item $i \in \{1, \ldots, Q\}$ under scenario s. For any two scenarios s, t, we define

$$\text{Stability}(s, t) = 1 - \frac{1}{4Q} \sum_{i=1}^{Q} |r_{i,s} - r_{i,t}|,$$
$$\text{with } Q = 24.$$

This normalizes the maximum per-item difference of 4 (on a 1–5 scale) to $[0, 1]$. In our main comparison we use s = Language and t = Citizen; we also report (s, t) = (Persian, English) and (US, Iran) where noted.

By embedding these six dimensions directly in our probes, we can (1) align with prior cultural-alignment studies, (2) explain which aspects of Iranian cultural values each model struggles to reflect, and (3) quantify the effect of simple role prompts on deep cultural representation, complementing narrative or cloze-style benchmarks.

3.4 Prompt Design

For each task, we compare prompts in *Persian* versus *English* (when applicable), and *act-as* versus *direct* forms for cultural probing. We decided to include English instructions, as at least earlier versions of ChatGPT proved to be more consistent when prompted with English task instruction (Lai et al., 2023b). The prompts for Persian and English were created by a Persian native speaker and translated to English (L2) by the same person. Linguistic and knowledge tasks use few-shot templates, except for grammar correction using zero-shot prompts, and cultural probing uses zero-shot VSM13-style templates. Find an English (for comprehension) example for the grammar correction task below:

```
The following word has a spelling
mistake. Just write the correct form of
it without any explanation and without
any diacritical marks (such as "ʼ").
Just one word: {word}
```

More can be found on our GitHub repository for reproducibility.

4 Results

4.1 Language Proficiency

On paraphrase detection, Fig. 2 shows that GPT-4o and Gemini achieve the highest accuracies in Persian and English, with barely a difference between the two. DeepSeek (82 % for Persian vs 68 % for English) and Llama 3.3 (81 % for Persian vs 46 % for English) show larger cross-lingual gaps, indicating weaker native-Persian paraphrase understanding. Their behaviour seems to show that the findings of Lai et al. (2023b) cannot be transferred to all scenarios as Llama and Gemini perform clearly better with Persian prompts than when prompted in English. Failure rates remain near zero for GPT-4o across both languages, but rise to 19–42 % for DeepSeek and Llama when prompted in English (bottom panel in Fig.2). For reference, the paraphrase split's majority-class baseline is $216/350 = 61.7\%$, so accuracies around 80% cannot be attributed to class skew alone (see Table 4 in Appendix A).

Entailment exhibits a similar pattern: GPT-4o/Persian attains 80 % accuracy (English 82 %), Gemini/Persian 74 % vs 80 % and Llama/Persian 11 % vs 66 %. DeepSeek and Llama effectively fail few-shot Persian NLI, but English prompts partially rescue performance. This highlights that direct Persian zero-shot inference remains highly unreliable for some models for the entailment tasks. Comparing our results to Abaskohi et al. (2024) the GPT family seems to have improved overall, while Llama 3.3 and Deepseek-V3 fall behind the older GPT models probed in their experiments.

All models fall below 35 % accuracy in Persian grammar correction with Persian prompts: GPT-4o (34 %), Gemini (33 %), DeepSeek (24 %), Llama (18 %). English prompts even fall slightly below the Persian prompt variants. Failure rates are correspondingly low (2–5 %) for GPT-4o, Gemini and Deepseek with Persian prompts but climb to 11 % for Llama in Persian. The corresponding error rates for English prompts are higher for all tested models. While we have no prior experiments in Persian for comparison, for English accuracies as high as around 90 % can be achieved (Atox and Clark, 2024).

Table 2 reports BLEU scores for Persian-prompted and English-prompted translations for English→Persian. Overall, translation performance is uniformly low across models, with no clear advantage for prompting in either lan-

guage. GPT-4o and Gemini lead marginally, while Llama trails behind, indicating that even state-of-the-art LLMs produce only rudimentary Persian translations without specialized MT fine-tuning (Abaskohi et al., 2024).

Model	Persian Prompt BLEU	English Prompt BLEU
GPT-4o	5.79	5.92
Gemini	5.85	5.79
DeepSeek	5.67	5.85
Llama	4.48	4.34

Table 2: Final BLEU scores for each model for English→Persian translations.

4.2 World-Knowledge Tasks

Accuracy in factual multiple-choice remains under 33 % for all models. GPT-4o leads at 27 % for Persian and 33 % for English prompts, followed by Gemini (32 % for Persian and 32 % for English), DeepSeek (11 % for Persian and 15 % for English), and Llama (15 % for Persian and 10 % for English). Failure rates reflect these low scores, with DeepSeek and Llama failing around 50 % of the time in Persian, versus 15 % for GPT-4o. The highest failure rate is reported as 63 % for Llama prompted in English.

Logical reasoning is the most challenging: GPT-4o achieves only 22 % for Persian and 25 % for English prompts, Gemini 21 % for both Persian and English prompts, DeepSeek 13 % for Persian and 19 % for English prompts, and Llama 15 % for Persian and 21 % for English prompts. Failure rates exceed 40 % for DeepSeek and Llama in Persian, underscoring profound gaps in few-shot reasoning capabilities for low-resource languages. Further our results are comparable to Abaskohi et al. (2024) who reported 30% accuracy as highest result with earlier models from the GPT family.

4.3 Task Failures

While our main results focus on aggregate accuracy and cultural alignment, we observed systematic breakdowns on individual tasks that reveal distinct failure modes beyond mere score drops. In the Persian-prompted natural language inference (NLI) task, for example, all models struggled with format compliance and label consistency. DeepSeek frequently returned full explanations rather than the single-token labels `e`/`n`/`c`, making over 90 % of its outputs unparseable and driving its measured accuracy to 0 %. Even when a valid label was produced, predictions were erratic: entailment was over-predicted (many gold neutrals → "e") and contradictions were missed. Gemini and Llama exhibited similar breakdowns under Persian prompts—verbose multi-language responses or outright format violations, despite far cleaner behavior when the same tasks were presented in English. These failures point to (1) poor instruction-following in Persian, (2) cross-lingual reasoning deficits, and (3) the necessity of enforcing strict output constraints in evaluation.

پاسخ: n
说明：第一句讨论文学，第二句谈及金融，二者无蕴含关系。
`HttpServletRequest.getParameter("foo");`

Figure 1: Example of a Llama NLI failure under Persian prompting: a Persian label, Chinese explanation (rendered via CJK), and stray Java code.

Moreover, Llama displayed a particularly dramatic failure mode when we used English prompts on Persian inputs: rather than perform NLI, it often "refused" or generated incoherent multilingual "meltdowns", as exemplified in Fig. 1.

Such outputs mix Persian, Chinese, English explanations and even Java code fragments—utterly unusable for NLI. This "language-agnostic meltdown" contrasts with its occasionally strong performance when it does obey instructions in Persian, underscoring that Llama does not necessarily benefit from English task prompting.

Similar issues arose in other classification tasks (paraphrase detection, factual QA): models either produced out-of-range labels (e.g. "5" in a 1–4 multiple-choice task) or collapsed into infinite loops of repeated tokens when prompted in English. These error patterns underscore that—beyond low overall accuracy—LLMs can exhibit total instruction-following failures or catastrophic generation breakdowns once they operate outside their primary training language. Gemini even claimed at some point that it was an English language model, when prompted in English for a Persian task. Addressing such task-specific failures will require both tighter prompt engineering (e.g. enforced format templates or sanity-checks) and targeted architectural or fine-tuning interventions to stabilize behavior in Persian.

4.4 Key Takeaways

Across linguistic and knowledge tasks, (1) GPT-4o and Gemini retain the highest baseline accuracy in Persian, (2) English prompts yield small gains for strong models but "rescue" performance only for weaker ones, and (3) open-source models (DeepSeek, Llama) struggle markedly in Persian few-shot settings. These disparities reveal systematic cross-lingual performance gaps and motivate deeper investigations into cultural and prompt-engineering interventions in low-resource contexts.

4.5 Cultural Alignment

We define alignment as the correct ordering of Hofstede dimension scores for the Iran/United States pair. Table 3 reports alignment per dimension (✓/✗) and the overall proportion aligned. We compare each model's VSM13 indices under both Persian-only ("Language") and act-as-Iranian ("Citizen") prompts against the ground-truth ordering.

Under Persian-only ("Language") prompts, DeepSeek aligns on 2 / 6 dimensions (33 %), GPT-4o on 3 / 6 (50 %), and both Llama and Gemini on 4 / 6 (67 %). With the act-as-Iranian ("Citizen") prompt, alignment rises to 67 % (4 / 6) for DeepSeek, 50 % (3 / 6) for GPT-4o, and 83 % (5 / 6) for both Llama and Gemini. All four models correctly rank Individualism (IDV), Uncertainty Avoidance (UAI), and Indulgence vs. Restraint (IVR) under the "Citizen" prompt. Only Llama and Gemini correctly order Long-Term Orientation (LTO), and none correctly order Masculinity (MAS). Explicit role cues therefore partially mitigate—but do not eliminate—cultural misalignment.

Across the 24 VSM13 items, GPT-4o exhibits the lowest variability (Persian σ = 0.0876; IR Citizen σ = 0.0319), followed by Llama (σ = 0.0982; 0.0393). DeepSeek shows moderate variability (σ = 0.1634; 0.1382), and Gemini the highest (σ = 0.2740; 0.3132). Cross-scenario similarity—measured as $\mathrm{Stability}(s,t) = 1 - \frac{1}{4Q}\sum_{i=1}^{Q}|r_{i,s} - r_{i,t}|$, $Q = 24$, is highest for Llama (0.917 Pers vs. Eng; 0.903 US vs. Iran), then GPT-4o (0.892; 0.889), DeepSeek (0.872; 0.920), and lowest for Gemini (0.823; 0.858). These consistency trends mirror the index alignment results.

While Llama and Gemini capture the strongest overall alignment with Iran's Hofstede profile (83 % under "Citizen" prompts), DeepSeek and GPT-4o show more modest performance—DeepSeek improving from 33 % to 67 %, and GPT-4o remaining at 50 %. Role-based prompts boost alignment—lifting Llama and Gemini to 83 %, DeepSeek to 67 %, and GPT-4o to 50 %. Yet no model achieves perfect ordering, especially on PDI and MAS. Because even the best models in our sample misorder at least on one dimension, we recommend fine-tuning on Persian sociolinguistic corpora to reduce this error.

5 Discussion

Across our suite of tasks, Persian-prompted performance consistently lags behind the English-prompted baseline, and "act-as-Iranian" role cues do virtually nothing to close that gap on core language skills. For instance, in spelling correction (Section 4.1) GPT-4o drops from 42 % to 27 % accuracy (–15 pp) when moving to Persian prompts, while Gemini falls from 40 % to 30 % (–10 pp). DeepSeek and Llama see smaller, yet still substantial, losses (–5 pp and –6 pp respectively). Paraphrase classification is slightly more robust—GPT-4o only loses 2 pp (84 %→82 %), Llama 7 pp (81 %→74 %)—but again the "act-as" instruction shifts these by at most 1–2 pp, confirming that simple role-play cues cannot compensate for missing Persian fluency.

Knowledge-based tasks show somewhat smaller but still significant gaps. In factual QA (Section 4.2), GPT-4o gains a modest +5 pp when switching to English prompts (27 %→32 %), while Llama actually declines (25 %→18 %)—a 7 pp drop. Logical reasoning tops out at only 25 % for GPT-4o even under English prompts (22 % in Persian). Auto-translating Persian inputs into English recovers another 5–10 pp across tasks, but still falls short of the English-native baseline ($\approx$ 85 % reported in purely English benchmarks). Thus, translation remains only a partial band-aid, benefiting those with bilingual pipelines but doing little to improve direct Persian interactions. Additionally, task failures were not analyzed in the translation task, as it would have required a more sophisticated task failure detection than mere formatting.

Our cultural-probing results (Section 4.5) further illustrate systemic misalignment rather than probably deliberate bias. Models like GPT-4o and Llama exhibit very low answer variance ($\sigma \approx 0.02$) yet repeatedly misorder key Hofstede dimensions—Power Distance and Masculinity—under both Persian-only and "act-as" prompts. This pat-

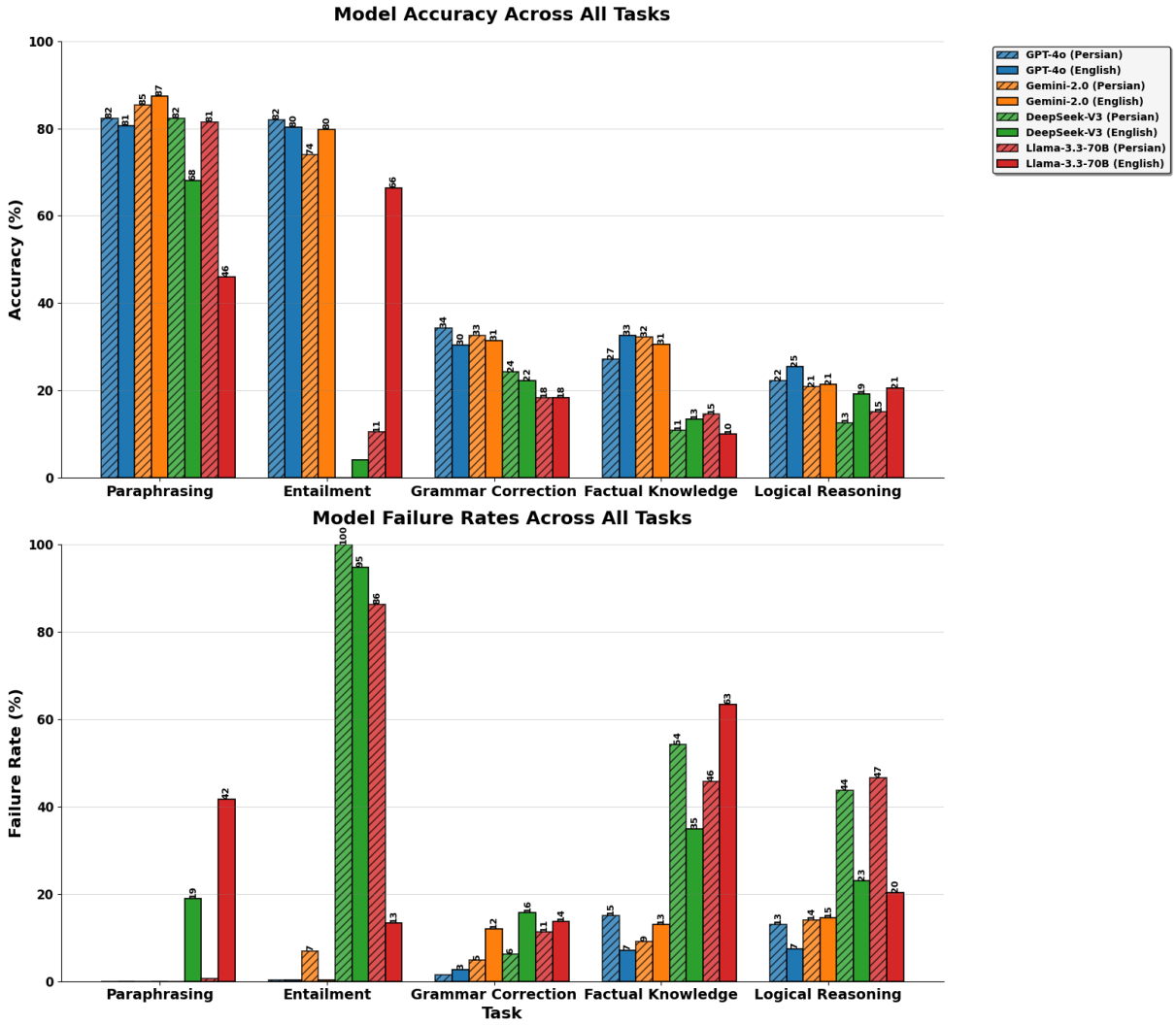


Figure 2: Accuracy and task failure rates across all models and tasks.

tern mirrors findings in AlKhamissi et al. (2024), where neither simple role cues nor monolingual fine-tuning fully closed the gap on World Values Survey alignment. In that work, only by combining native-language prompting with targeted fine-tuning ("Anthropological Prompting") did alignment approach parity. Our Persian results point in the same direction: without richer Persian pretraining data and more sophisticated cultural scaffolding, LLMs remain skewed toward WEIRD norms.

Together, our two axes of failure imply that "knowing Persian" and "thinking Persian" are separable challenges. Improving linguistic fluency demands probably targeted in-language data, lightweight fine-tuning or adapters on diverse Persian corpora (news, literature, social media), as task-specific fine-tuned SOTA models still outperform LLMs (Abaskohi et al., 2024). Deepening cultural fidelity will likewise require more than zero-shot role cues. Anthropological or chain-of-thought prompting(AlKhamissi et al., 2024), and richer Persian cultural text in pretraining, are the most plausible routes to reliable alignment. Only by coupling linguistic adaptation with cultural grounding can future LLMs begin to serve Persian speakers with both fluency and fidelity.

6 Future Work

To build on these findings, future work should broaden both the linguistic and cultural horizons. Expanding beyond Iranian Persian to include for example Dari and Tajik variants would reveal whether the gaps we observe are universal or dialect-specific. Incorporating human-in-the-loop evaluations will be essential for judging fluency, cultural appropriateness and downstream utility. On the modeling side, it will be important to study how fine-tuning on curated Persian corpora (e.g.

Dimension	Ground Truth	DeepSeek		GPT-4o		Llama		Gemini	
		Language	Citizen	Language	Citizen	Language	Citizen	Language	Citizen
PDI	[United States, Iran]	✗	✓	✗	✗	✗	✓	✓	✓
IDV	[Iran, United States]	✗	✓	✓	✓	✓	✓	✓	✓
MAS	[Iran, United States]	✓	✗	✓	✗	✓	✗	✗	✗
UAI	[United States, Iran]	✓	✓	✓	✓	✓	✓	✓	✓
LTO	[Iran, United States]	✗	✗	✗	✗	✗	✓	✗	✓
IVR	[Iran, United States]	✗	✓	✗	✓	✓	✓	✓	✓
Overall Accuracy	—	33 %	67 %	50 %	50 %	67 %	83%	67%	83 %

Table 3: Cultural Alignment Ranking Comparison Across Models. ✓ = correct ranking alignment with ground truth; ✗ = incorrect alignment.

newswire, literature, social media) affects both linguistic competence and cultural alignment. Chain-of-thought/"anthropological" prompting may unlock deeper, context-sensitive reasoning that zero-shot setups cannot (AlKhamissi et al., 2024). Finally, cultural evaluation itself stands to benefit from complementary frameworks (e.g. narrative-driven tests like PerCul or multi-dimensional survey simulations) to triangulate the complex ways LLMs mirror—or misrepresent—real-world perspectives.

7 Conclusion

Our experiments reveal stark cross-lingual performance gaps in today's leading LLMs: models that achieve near-state-of-the-art results in English suffer accuracy losses when the same task types are presented in Persian (Atox and Clark, 2024). Few-shot prompt designs in English or Persian cannot compensate for the underlying paucity of high-quality Persian data or the models' limited instruction-following in a non-Latin script.

Likewise, simple "act-as" cultural prompts do little to recover a faithful Iranian profile: even the best models misorder core Hofstede dimensions and show only modest alignment gains, underscoring a deeper misalignment that goes beyond surface-level persona shifts. Together, our findings argue that achieving true multilingual equity will require more than smarter prompts—it demands richer, culturally representative pretraining data, targeted adaptation (e.g. fine-tuning or adapters on Persian resources), and human-centered evaluation frameworks that can validate both linguistic fluency and cultural nuance.

Limitations

Our study offers another look into Persian SOTA LLM performance, but its scope is inevitably constrained. We focus on a handful of benchmark tasks (spelling, paraphrase/NLI, QA, analogy, translation) and sample only 350 examples per split; many important phenomena—idiomatic usage, named-entity recognition, temporal inference or dialectal variation (Dari, Tajik) fall outside our purview. We rely mostly on automated metrics (exact-match accuracy, BLEU, Hofstede VSM13 indices) and only qualitatively analyzed task failures. Moreover, all reported scores are single-shot estimates from one run. No standard errors or deviations are shown, so the precision of our comparisons is limited. Future work should increase the number of runs (e.g., via repeated trials) and expand the dataset to enable error-bar visualizations and more robust statistical inference. Finally, our cultural probe leans on Hofstede's dimensions—a well-known but not uncontroversial framework—and tests only direct Persian prompts or simple "act like an Iranian" cues, without exploring richer narrative or survey-based approaches, or other nationalities that have Persian native speakers.

References

Amirhossein Abaskohi, Sara Baruni, Mostafa Masoudi, Nesa Abbasi, Mohammad Hadi Babalou, Ali Edalat, Sepehr Kamahi, Samin Mahdizadeh Sani, Nikoo Naghavian, Danial Namazifard, Pouya Sadeghi, and Yadollah Yaghoobzadeh. 2024. Benchmarking large language models for Persian: A preliminary study focusing on ChatGPT. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2189–2203, Torino, Italia. ELRA and ICCL.

Badr AlKhamissi, Muhammad ElNokrashy, Mai Alkhamissi, and Mona Diab. 2024. Investigating cultural alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12404–12422, Bangkok, Thailand. Association for Computational Linguistics.

Nathan Atox and Mason Clark. 2024. Evaluating large language models through the lens of linguistic proficiency and world knowledge: A comparative study. *Authorea Preprints*.

Laleh Barari and Badr QasemiZadeh. 2005. CloniZER: Adaptive language-independent spell checker. In *Proceedings of the AIML 2005 Conference (CICC)*, pages 65–71, Cairo, Egypt.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Common Crawl. 2025. Distribution of languages. https://commoncrawl.github.io/cc-crawl-statistics/plots/languages.html. Accessed: 2025-06-13.

Siamak Shakeri Pedram Hosseini Pouya Pezeshkpour Malihe Alikhani Moin Aminnaseri Marzieh Bitaab Faeze Brahman Sarik Ghazarian Mozhdeh Gheini Arman Kabiri Rabeeh Karimi Mahabadi Omid Memarrast Ahmadreza Mosallanezhad Erfan Noury Shahab Raji Mohammad Sadegh Rasooli Sepideh Sadeghi Erfan Sadeqi Azer Niloofar Safi Samghabadi Mahsa Shafaei Saber Sheybani Ali Tazarv Yadollah Yaghoobzadeh Daniel Khashabi, Arman Cohan. 2020. ParsiNLU: a suite of language understanding challenges for persian. *arXiv*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2025. Deepseek-v3 technical report.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2025. *Ethnologue: Languages of the World*, twenty-eighth edition. SIL International, Dallas, Texas.

Google DeepMind. 2024. Introducing Gemini 2.0: Our new ai model for the agentic era. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/. Accessed: 2025-07-13.

Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. In *Proceedings of the ACM Web Conference 2024*, pages 2627–2638.

Omid Kashefi. 2018. Mizan: a large persian-english parallel corpus. *arXiv preprint arXiv:1801.02107*.

Julia Kharchenko, Tanya Roosta, Aman Chadha, and Chirag Shah. 2024. How well do llms represent values across cultures? empirical analysis of llm responses based on hofstede cultural dimensions. *arXiv preprint arXiv:2406.14805*.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. 2023a. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages

318–327, Singapore. Association for Computational Linguistics.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023b. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv preprint arXiv:2304.05613*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Reem Masoud, Ziquan Liu, Martin Ferianc, Philip C. Treleaven, and Miguel Rodrigues Rodrigues. 2025. Cultural alignment in large language models: An explanatory analysis based on hofstede's cultural dimensions. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8474–8503, Abu Dhabi, UAE. Association for Computational Linguistics.

Meta AI. 2025. LLaMA 3.3 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md. Accessed: 2025-07-13.

Erfan Moosavi Monazzah, Vahid Rahimzadeh, Yadollah Yaghoobzadeh, Azadeh Shakery, and Mohammad Taher Pilehvar. 2025. PerCul: A story-driven cultural evaluation of LLMs in Persian. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12670–12687, Albuquerque, New Mexico. Association for Computational Linguistics.

OpenAI. 2024. Gpt-4o system card. https://openai.com/index/gpt-4o-system-card/. Accessed: 2025-07-13.

Appendix

A Class Balance

Entailment (e/n/c)	122	110	118	
Paraphrase (0/1)	216	134		
Factual Recall (1–4)	98	88	94	70
Logical Reasoning (1–4)	75	102	93	80

Table 4: Counts per class (gold labels) for sampled splits.

GeoLogicQA - A Benchmark for Evaluating Logical Reasoning in Georgian For Large Language Models

Irakli Koberidze
Dep. of Computer Science
Tbilisi State University
Tbilisi Georgia
irakli.koberidze@tsu.ge

Archil Elizbarashvili
Dep. of Computer Science
Tbilisi State University
Tbilisi Georgia
archil.elizbarashvili@tsu.ge

Magda Tsintsadze
Dep. of Computer Science
Tbilisi State University
Tbilisi Georgia
magda.tsintsadze@tsu.ge

Abstract

Advancements in LLMs have largely overlooked low-resource languages (LRLs), creating a gap in evaluation benchmarks. To address this for Georgian, a Kartvelian language, we introduce GeoLogicQA. This novel, manually-curated benchmark assesses LLMs' logical and inferential reasoning through 100 questions. Questions cover syllogistic deduction, inferential reading comprehension, common-sense reasoning, and arithmetic, adapted from challenging sources (Kangaroo Mathematics Competition) and validated by native Georgian speakers for linguistic nuances. Initial evaluations of state-of-the-art LLMs (Gemini 2.5 Flash, DeepSeek-V3, Grok-3, GPT-4o) show an average accuracy of 64% to 83%, significantly exceeding the human baseline of 47%. While demonstrating strong reasoning potential, error analysis reveals persistent challenges in multi-step combinatorial and highly constrained inferential tasks. GeoLogicQA is a public resource for tracking progress and diagnosing weaknesses in Georgian LLMs. We plan to expand the benchmark and establish a public leader-board to foster continuous improvement.

1 Introduction

The rapid evolution of Large Language Models (LLMs), like GPT-4 and Llama 3, has revolutionized AI, excelling in natural language generation and problem-solving. This progress is largely due to vast computational resources and datasets in high-resource languages (HRLs), primarily English (OpenAI et al., 2024). Consequently, low-resource languages (LRLs) face a significant disparity in LLM development and evaluation, lacking appropriate benchmarks to assess their true capabilities.

Georgian, a Kartvelian agglutinative language, exemplifies this resource gap in NLP. Existing Georgian NLP resources are insufficient for evaluating the deeper cognitive abilities of modern generative LLMs, failing to test complex logical reasoning, inferential comprehension, or common-sense understanding. The critical question for Georgian NLP has evolved from "Can a model process Georgian text?" to "Can a model think in Georgian?", requiring evaluation beyond just a pattern recognition.

This work introduces GeoLogicQA, a novel, manually-curated evaluation benchmark designed to assess the logical and inferential reasoning abilities of LLMs in Georgian language. It offers a diverse set of multiple-choice questions covering syllogistic deduction, inferential reading comprehension, common-sense reasoning, and arithmetic problem-solving, aiming to diagnose model weaknesses. Our primary goal is to provide a rigorous, publicly accessible resource for tracking progress and foster more robust Georgian LLMs. Initial evaluations of models like ChatGPT, DeepSeek, Gemini, and Grok on our 100-question benchmark show an average performance below 70% accuracy, highlighting significant challenges in complex Georgian reasoning.

The paper is structured as follows: Section 2 reviews existing LLM evaluation benchmarks and Georgian NLP resources. Section 3 details GeoLogicQA's design principles, task categories, and data curation. Section 4 outlines the experimental setup, presents baseline results from LLM evaluations on GeoLogicQA, and analyzes performance and error patterns, followed by section 5, were implications, limitations, and future work is discussed. The work is concluded by section 6, which summarizes

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025,
pages 121–130, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-013

the contributions and emphasizes the broader impact of the resource.

2 Background and Related Work

The landscape of Natural Language Processing has been significantly shaped by the development of sophisticated evaluation benchmarks that measure the capabilities of large language models. These benchmarks serve as crucial instruments for comparing models, identifying their strengths, and diagnosing their limitations (Wang et al., 2019). Our work on GeoLogicQA is situated within this broader context, while simultaneously addressing the unique challenges presented by low-resource languages.

2.1 Benchmark Creation for LRLs

Benchmark is a set of standardized tests that assess LLM performance across various tasks. Creating benchmarks for LRLs involves several steps that ensure the evaluation is meaningful, fair, and generalizable across models. The challenges faced by Georgian NLP are not unique, many low-resource languages worldwide contend with similar limitations in terms of data availability and evaluation infrastructure. Consequently, there has been a growing global movement within the NLP community to address this disparity by creating dedicated benchmarks for LRLs. Efforts often involve:

- Typical strategies that translate high-resource-language (HRL) benchmarks into low-resource languages (LRLs) often lose culturally-specific context or meaning, limiting faithful assessment of reasoning capabilities in the target LRL (Ghafoor et al., 2021). Further, studies such as (Alhanai et al., 2024) show that direct translations of benchmarks (e.g., Winogrande, MMLU into low-resource African languages) underperform until cultural adjustments are incorporated—highlighting that simple translation fails to preserve the nuanced reasoning demands of the original tasks.

- Collaborative efforts involving native speakers and linguists are crucial for curating high-quality, culturally relevant datasets; empirical evidence shows that native-written corpora enhance lexical diversity and cultural content (Cahyawijaya et al., 2023), while participatory and community-centric approaches ensure linguistic authenticity and foster richer dataset design (Ousidhoum et al., 2025).

- Developing benchmarks tailored to specific linguistic phenomena or reasoning types that are particularly challenging for a given LRL. This approach helps diagnose unique model weaknesses (Goyal and Dan, 2025; Sánchez et al., 2024; Bean et al., 2024).

By developing GeoLogicQA,[1] a manually-curated benchmark for Georgian logical reasoning, we contribute to the crucial effort of creating equitable and culturally-relevant evaluations for LLMs in low-resource languages.

2.2 Evaluation Benchmarks in HRLs

In high-resource languages, particularly English, a rich ecosystem of evaluation benchmarks exists, each targeting different facets of language understanding and reasoning. Prominent examples include:

GLUE (General Language Understanding Evaluation) and its successor, SuperGLUE: A collection of diverse natural language understanding tasks, such as sentiment analysis, textual entailment, question answering, and paraphrase detection (Wang et al., 2019). They assess a model's ability to capture semantic and syntactic nuances across various linguistic phenomena. While foundational, GLUE and SuperGLUE primarily evaluate general language understanding rather than complex, multi-step logical reasoning.

MMLU (Massive Multitask Language Understanding): A significant advancement in evaluating LLMs by testing knowledge and reasoning across 57 diverse subjects, including humanities, social sciences, STEM, and professional disciplines (Hendrycks et al., 2021). It is designed to be challenging, often requiring zero-shot or few-shot inference, and assesses a model's ability to apply pre-trained knowledge to novel problems. MMLU's focus on a wide array of academic and professional subjects makes it a strong indicator of a model's

[1] https://github.com/irakli97/GeoLogicQA.

general intelligence and reasoning capabilities beyond simple pattern matching.

Big-Bench (Beyond the Imitation Game Benchmark): Includes over 200 tasks, many of which are specifically designed to push the boundaries of LLM capabilities, encompassing logical reasoning, common-sense reasoning, mathematical problem-solving, and creative writing (Srivastava et al., 2023). Big-Bench Hard (BBH), a subset of the most challenging Big-Bench tasks, explicitly targets reasoning abilities that are difficult for current LLMs, often involving multi-hop deduction, complex causal relationships, or counterfactual reasoning. These benchmarks provide a robust framework for assessing higher-order cognitive functions in LLMs.

These HRL benchmarks have been instrumental in driving the rapid progress of LLMs by providing standardized, rigorous, and publicly accessible evaluation tools. They allow researchers to track performance, pinpoint weaknesses, and develop more sophisticated models.

2.3 NLP Resources for Georgian

Despite the global advancements in NLP, the Georgian language faces significant challenges due to its low-resource status (Pakray et al., 2025). The availability of high-quality training data and advanced NLP tools for Georgian is notably limited compared to HRLs. Existing resources primarily include several initiatives focused on compiling Georgian text corpora from various sources, such as Wikipedia, news articles, and literary works. These corpora are valuable for foundational tasks like language modeling and morphological analysis (Doborjginidze and Lobzhanidze, 2016). Limited parallel corpora exist for machine translation between Georgian and other languages, supporting cross-lingual transfer. Also, the tools for Part-of-Speech tagging, lemmatization, and dependency parsing have been developed, aiding in basic linguistic analysis (Giorkhelidze, 2017). However, these existing resources predominantly cater to traditional NLP tasks and surface-level linguistic analysis. They largely fall short in providing the challenging, reasoning-focused datasets necessary to evaluate the deep language understanding and inferential capabilities of modern generative LLMs. The scarcity of structured, annotated data designed for complex logical inference means that current Georgian NLP lacks the benchmarks required to gauge how well LLMs can process and reason with Georgian text beyond simple recognition or translation. There is a marked absence of standardized datasets that demand multi-step reasoning, logical deduction, or nuanced common-sense inference in Georgian, creating a significant gap in the evaluation framework for advanced Georgian LLMs.

3 The GeoLogicQA Benchmark: Design and Curation

The GeoLogicQA benchmark is meticulously designed to provide a robust evaluation of Large Language Models' (LLMs) logical and inferential reasoning capabilities specifically within the Georgian language. This section details the core design principles, the diverse task categories included, and the rigorous data collection and validation processes employed to ensure the benchmark's quality and validity. Because of the unique linguistic characteristics of the Georgian language, including its agglutinative nature and prevalent polysemy (Ma et al., 2020), we had to apply careful curation when designing the GeoLogicQA benchmark to overcome these challenges.

3.1 Design Principles

GeoLogicQA's design is underpinned by several core principles aimed at comprehensively assessing LLMs' reasoning in a low-resource language context.

3.1.1 Focus on Logic and Inference

GeoLogicQA explicitly prioritizes tasks that demand genuine logical and inferential reasoning, moving beyond simple keyword matching, surface-level pattern recognition, or statistical correlations. The fundamental aim is to ascertain whether an LLM can truly "think in Georgian," grasping complex relationships and deriving non-explicit conclusions, rather than merely processing and reproducing text. This necessitates questions that require multi-step reasoning, an understanding of causality, and the ability to synthesize information from various premises.

3.1.2 Linguistic and Cultural Nuance

GeoLogicQA deeply integrates Georgian linguistic and cultural nuances. Questions weren't just translated; they were crafted to be natural, culturally relevant, and contextually appropriate for native Georgian speakers. This means scenarios, idioms, and common knowledge referenced in the questions genuinely resonate within the Georgian context, avoiding awkward translations that could distort meaning or reasoning challenges.

Crucially, the design process addressed polysemy in Georgian, where words and phrases can have multiple meanings. For example, "მანძილის დაფარვა" can mean "to cover distance" or "to cover something with a lid." To prevent misinterpretations by LLMs due to linguistic misunderstanding rather than a lack of reasoning, question designers carefully constructed sentences and scenarios. They provided unambiguous contextual cues, ensuring only the intended meaning was conveyed. This precise phrasing was paramount to isolating and testing true reasoning rather than surface-level recognition.

3.1.3 Task Diversity

GeoLogicQA incorporates a diverse range of reasoning task categories to provide a comprehensive assessment of LLMs' cognitive abilities. These categories include syllogistic and deductive reasoning, reading comprehension with inference, common-sense reasoning, and arithmetic reasoning. This diversity ensures that the benchmark evaluates a broad spectrum of reasoning skills, preventing LLMs from excelling based on proficiency in only one type of task.

3.1.4 Quality Assurance

The creation and validation of questions for GeoLogicQA followed a rigorous, multi-stage quality assurance process. Each question was meticulously reviewed to ensure it was unambiguous, logically sound, and genuinely tested complex reasoning rather than simple recall or pattern matching. This iterative process involved expert review and refinement to eliminate any potential for misinterpretation or an unintended correct answer, guaranteeing the integrity of the evaluation.

3.2 Task Categories and Examples

GeoLogicQA comprises four distinct task categories, each designed to probe specific facets of logical and inferential reasoning. The following examples illustrate the type of questions included in each category:

3.2.1 Category 1: Syllogistic & Deductive Reasoning

Description: Tasks requiring deriving a logically sound conclusion from a set of premises. These questions often test a model's ability to follow chains of inference and identify valid deductions.

Example: Georgian: "ყოველ მონეტას აქვს ორი მხარე, 'გერბი' და 'საფასური'. მაგიდაზე ძევს ხუთი მონეტა, ხუთივე ზემოთ იყურება 'გერბით'. ყოველ ბიჯზე უნდა ამოვატრიალოთ ზუსტად სამი მონეტა. იპოვეთ ბიჯების ის უმცირესი რაოდენობა, რომლის შემდეგაც ხუთივე მონეტა ზემოთ იქნება 'საფასურით'." English: "Five coins are lying on a table with the "heads" side up. At each step you must turn over exactly three of the coins. What is the least number of steps required to have all the coins lying with the "tails" side up?"

3.2.2 Category 2: Reading Comprehension with Inference

Description: Presenting a short paragraph or scenario and asking a question where the answer is not explicitly stated but must be inferred from the provided text, requiring deeper understanding and synthesis of information.

Example: Georgian: "კინოთეატრში ერთ რიგში ზის 23 ცხოველი. თითოეული ცხოველი არის ან თახვი ან კენგურუ. თითოეულ ცხოველს ჰყავს სულ მცირე ერთი მეზობელი, რომელიც კენგურუა. ყველაზე მეტი რამდენი თახვი შეიძლება იყოს რიგში?" English: "There are 23 animals sitting in a row at the cinema. Each animal is either a beaver or a kangaroo. Everyone has at least one neighbour who is a kangaroo. What is the largest possible number of beavers in the row?"

3.2.3 Category 3: Common-Sense Reasoning

Description: Questions relying on implicit, everyday knowledge about the world and practical understanding of cause-and-effect rela-

tionships, adapted for a Georgian cultural context.

Example: Georgian: "აისბერგს კუბის ფორმა აქვს. მისი მოცულობის 90% არის წყლის ზედაპირის ქვემოთ. წყლის ზედაპირის ზემოთ ჩანს კუბის მხოლოდ სამი წიბოს ნაწილი. ამ ნაწილების სიგრძეებია: 24 მ, 25 მ და 27 მ. იპოვეთ კუბის წიბოს სიგრძე." English: "An iceberg has the shape of a cube. Exactly 90% of its volume is hidden below the surface of the water. Three edges of the cube are partially visible over the water. The visible parts of these edges are 24m, 25m and 27m. How long is an edge of the cube?"

3.2.4 Category 4: Arithmetic Reasoning

Description: Word problems that require extracting numerical quantities, understanding relationships, and performing basic to moderately complex calculations within a narrative context.

Example: Georgian: "იპოვეთ $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$ ნამრავლის ბოლო ორი ციფრის ჯამი." English: "What is the sum of the last two digits of the product $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$?"

3.3 Data Collection and Validation

The quality and challenge of GeoLogicQA are rooted in its careful data collection and rigorous validation processes.

3.3.1 Source

The questions used in GeoLogicQA were adapted from the annual Kangaroo Mathematics Competitions, organized by the "Association Kangourou sans Frontières (AKSF)" (https://www.aksf.org) and officially translated into Georgian by its representatives in Georgia. We primarily selected problems from the 9th to 12th-grade levels to ensure a high level of cognitive demand and complexity, making them suitable for evaluating advanced reasoning capabilities in LLMs.

Crucial modifications were made to ensure AI interpretability without altering the core reasoning challenge. This involved standardizing mathematical notations (e.g., using ^ for powers), adding parentheses for clarity, and converting essential visual information from image-based questions into descriptive text. This last step was only done when the underlying logical reasoning could be fully preserved without the visual component, avoiding the need for computer vision. We obtained explicit permission from AKSF for ethical data sourcing.

3.3.2 Validation Process

A rigorous multi-step verification process was implemented to ensure the quality, clarity, and correctness of each question and its intended answer:

- **Initial Drafting and Adaptation:** The core research team was responsible for the initial drafting and adaptation of questions from the source materials, ensuring adherence to the design principles.
- **Expert Review by Native Speakers:** Each adapted question underwent rigorous review by a panel of at least two independent native Georgian speakers. This panel critically evaluated each question for linguistic clarity, potential ambiguities (particularly addressing polysemy), naturalness of expression, and the unequivocal correctness of the designated answer. This step was paramount in refining the questions for precision and ensuring they truly assessed reasoning in Georgian, eliminating any linguistic pitfalls that might mislead an LLM.
- **Pilot Testing on Human Subjects:** A subset of the questions was pilot tested on human subjects to establish a human performance baseline. These selected questions are notoriously challenging for human students, as evidenced by the published average of 47% correct answers by students on the Kangaroo Mathematics Competition questions.[2] It is important to note that this relatively high percentage reflects the fact that participants in the upper grades of the competition are typically students with a strong interest and background in mathematics. This human baseline provides a vital context for evaluating LLM performance, highlighting the benchmark's inherent difficulty even for human solvers.

[2]Source: https://kenguru.ge/olympiad.

3.3.3 Statistics

The benchmark comprises of 100 questions.

4 Experimental Setup and Baseline Results

This section details the experimental methodology employed to evaluate large language models (LLMs) on the GeoLogicQA benchmark and presents the baseline results. We describe the specific models chosen, the evaluation protocol, and an in-depth analysis of their performance, including an error breakdown to highlight common challenges.

4.1 LLM Models

For the evaluation of logical and inferential reasoning capabilities in Georgian, a selection of advanced large language models was chosen. The models evaluated were **GPT-4o, Gemini 2.5 Flash, DeepSeek-V3, and Grok-3**. These models represent a diverse set of current state-of-the-art LLMs, offering a robust comparison of their performance on complex reasoning tasks in a low-resource language.

4.2 Evaluation Protocol

To ensure a consistent and fair assessment of each model's inherent reasoning abilities, a standardized evaluation protocol was strictly adhered to.

4.2.1 Prompting Strategy

For all evaluations, a **zero-shot prompting strategy** was employed. The full question text in Georgian, as presented in the GeoLogicQA benchmark, was directly submitted to each model without any additional instructions, examples, or specific formatting cues. This approach was chosen to assess the models' inherent reasoning capabilities in Georgian without external scaffolding. This method provides a direct measure of how well models understand and respond to novel, complex questions solely based on their pre-trained knowledge and reasoning faculties.

4.2.2 Metric

The performance of each LLM was quantified by its **accuracy**, defined as the percentage of correctly answered questions out of the total 100 questions in the GeoLogicQA benchmark. A correct answer was determined by an exact match with the ground truth solution. This binary metric provides a clear and unambiguous measure of successful reasoning.

4.3 Results

The baseline performance of the evaluated LLMs on the GeoLogicQA benchmark is presented in Table 1. For comparison, we include a human baseline derived from the performance of 9th to 12th-grade students on the adapted questions from the annual Kangaroo Mathematics Competition in Georgia.[3]

Model	Accuracy (%)
Gemini 2.5 Flash	83.00
DeepSeek-V3	74.00
Grok-3	67.00
GPT-4o	64.00
Human Baseline	47.0

Table 1: Baseline performance of evaluated LLMs and human subjects on the GeoLogicQA benchmark.

4.4 Analysis and Error Breakdown

The results demonstrate a clear hierarchy in performance among the evaluated LLMs, with **Gemini 2.5 Flash** emerging as the top-performing model, achieving an accuracy of 83.00%. Following closely were **DeepSeek-V3** (74.00%), **Grok-3** (67.00%), and **GPT-4o** (64.00%). A significant observation is that all evaluated LLMs substantially surpassed the **human baseline performance of 47.0%**. This indicates that current advanced LLMs possess a considerable advantage over human subjects on these specific types of logical and inferential reasoning tasks in Georgian, despite the benchmark's design to challenge models in a low-resource linguistic context.

While all models performed well above the human baseline, an analysis of specific errors reveals common challenging categories and unique failure modes. Syllogistic and Deductive Reasoning questions, as well as complex Arithmetic Reasoning tasks, often proved to be the most difficult for the models, aligning

[3]Data adapted from the official Georgian Kangaroo Competition statistics, available at https://www.kenguru.ge/posts/7700b50e-a89e-41c2-a6c8-24cab065b424.

with the inherent complexity of these problem types.

To illustrate, consider the following examples of observed errors:

- In a **Syllogistic & Deductive Reasoning** question about flipping five coins, where "Five coins are lying on a table with the "heads" side up. At each step you must turn over exactly three of the coins. What is the least number of steps required to have all the coins lying with the "tails" side up?" **all evaluated models failed** to provide the correct minimum number of steps. This suggests a fundamental challenge in multi-step combinatorial reasoning.

- For an **Arithmetic Reasoning** question that asked to "What is the sum of the last two digits of the product $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$?" **Gemini 2.5 Flash and DeepSeek-V3 correctly identified the answer**, while **GPT-4o and Grok-3 were incorrect**. This highlights varying levels of numerical reasoning and attention to detail among the models for specific arithmetic properties.

- A **Reading Comprehension with Inference** question posed a scenario: "In a cinema row, there are 23 animals sitting. Each animal is either a beaver or a kangaroo. Every animal has at least one neighbor who is a kangaroo. What is the maximum number of beavers there can be in the row?" Interestingly, **Grok-3 was the only model to correctly answer this question**, whereas Gemini 2.5 Flash, GPT-4o, and DeepSeek-V3 all failed. This particular instance points to Grok-3's potentially stronger ability to handle complex conditional constraints and infer maximum possibilities in a constrained environment.

- Another commonly challenging problem involved determining a specific digit in a product of "six consecutive numbers" forming a 12-digit number of the form 'abb cdd cdd abb'. In this **Arithmetic Reasoning** task, **Gemini 2.5 Flash and DeepSeek-V3 provided the correct answer**, while **GPT-4o and Grok-3 did not**. This error pattern indicates that some models struggle more with reverse engineering numerical properties or identifying specific digit values within large products based on structural constraints.

These examples underscore that while LLMs show robust performance on average, specific types of logical puzzles and intricate numerical challenges continue to pose significant hurdles, revealing areas for future model improvement in handling complex reasoning in Georgian.

5 Discussion

5.1 Key Takeaways

State-of-the-art Large Language Models (LLMs) consistently outperformed the human baseline of 47.0% on GeoLogicQA, a benchmark for logical and inferential reasoning in Georgian. Gemini 2.5 Flash led with 83.00% accuracy, followed by DeepSeek-V3 (74.00%), Grok-3 (67.00%), and GPT-4o (64.00%). This demonstrates a significant advantage for LLMs in structured logical and arithmetic problems, even in a low-resource language like Georgian. This performance gap underscores the rapid advancements in LLM reasoning. While LLMs excel at precise, multi-step deduction, they still struggle with complex multi-step combinatorial problems and nuanced inferential reading comprehension requiring the synthesis of multiple constraints. The varied performance across problem types highlights that no single model is universally superior, emphasizing the need for continued refinement in intricate logical deductions within low-resource language contexts.

5.2 Future Work

Building upon GeoLogicQA's initial release, our future work will focus on several key directions to expand its utility and impact:

- **Benchmark Expansion:** We plan to significantly expand the GeoLogicQA dataset by curating hundreds of additional questions. This expansion will not only increase statistical robustness but also allow for the inclusion of new task

categories. Potential additions include questions testing understanding of figurative language, detection of logical fallacies in natural arguments, and more complex causal reasoning scenarios that require deeper narrative comprehension.

- **Public Leaderboard and Community Contributions:** To foster continuous progress and facilitate comparative research, we intend to establish a publicly accessible leaderboard. This platform will allow researchers to submit their models' performance on GeoLogicQA, tracking advancements in Georgian LLM reasoning over time. Furthermore, we will actively encourage community contributions to the benchmark, inviting native Georgian speakers, linguists, and AI researchers to propose new questions and reasoning challenges. This collaborative approach will ensure the benchmark remains dynamic, comprehensive, and reflective of the evolving needs of the Georgian NLP community.

6 Conclusion

This paper introduces GeoLogicQA, the first dedicated benchmark for evaluating logical and inferential reasoning capabilities of Large Language Models in the Georgian language. Through meticulous manual curation and rigorous validation, GeoLogicQA provides a challenging set of 100 questions spanning syllogistic deduction, inferential reading comprehension, common-sense reasoning, and arithmetic problem-solving. Our baseline evaluations demonstrate that contemporary LLMs, notably Gemini 2.5 Flash, DeepSeek-V3, Grok-3, and GPT-4o, significantly outperform human subjects on these complex Georgian reasoning tasks, highlighting the advanced logical capabilities of current models even in low-resource linguistic contexts.

The creation and public release of GeoLogicQA address a critical gap in the evaluation infrastructure for Georgian Natural Language Processing, moving beyond superficial linguistic analysis to probe deeper cognitive abilities. This benchmark will serve as a vital resource for the research community, enabling systematic tracking of progress, identifying specific areas for model improvement, and fostering the development of more robust and intelligent LLMs for Georgian. As we continue to expand and refine GeoLogicQA, we emphasize the urgent and ongoing need for community-driven resource creation to ensure equitable and comprehensive AI development across the world's diverse linguistic landscape, ultimately paving the way for truly multilingual and reasoning-capable AI systems.

Limitations of GeoLogicQA

While a valuable step, GeoLogicQA has limitations. Firstly, its modest size of 100 questions limits statistical confidence compared to larger benchmarks, hindering comprehensive analysis across diverse logical challenges. Secondly, GeoLogicQA primarily focuses on structured logical, inferential, and arithmetic reasoning, lacking coverage of broader human-like reasoning. It currently omits common-sense reasoning (e.g., social understanding, ethical dilemmas, logical fallacy detection) and deep understanding of Georgian cultural nuances like idioms or proverbs. Finally, its reliance on adapted Math competition questions, though ensuring high cognitive demand, constrains the scope to formalized problems with single correct answers. This may not fully capture the breadth of real-world, open-ended, ambiguous, or creative reasoning challenges.

References

Alhanai, T., Kasumovic, A., Ghassemi, M., Zitzelberger, A., Lundin, J., and Chabot-Couture, G. (2024). Bridging the gap: Enhancing llm performance for low-resource african languages with new benchmarks, fine-tuning, and cultural adjustments.

Bean, A. M., Hellsten, S., Mayne, H., Magomere, J., Chi, E. A., Chi, R., Hale, S. A., and Kirk, H. R. (2024). Lingoly: A benchmark of olympiad-level linguistic reasoning puzzles in low-resource and extinct languages.

Cahyawijaya, S., Lovenia, H., Koto, F., Adhista, D., Dave, E., Oktavianti, S., Akbar, S. M., Lee, J., Shadieq, N., Cenggoro, T. W., Linuwih, H. W., Wilie, B., Muridan, G. P., Winata, G. I., Moeljadi, D., Aji, A. F., Purwarianti, A., and Fung, P. (2023). Nusawrites: Constructing high-quality corpora for underrepresented and extremely low-resource languages.

Doborjginidze, N. and Lobzhanidze, I. (2016). Corpus of the georgian language. In *Proceedings of the XVII EURALEX International Congress*, pages 328–335.

Ghafoor, A., Imran, A. S., Daudpota, S. M., Kastrati, Z., Abdullah, Batra, R., and Wani, M. A. (2021). The impact of translating resource-rich datasets to low-resource languages through multi-lingual text processing. *IEEE Access*, 9:124478–124490.

Giorkhelidze, G. (2017). Software tools for initial processing of georgian texts. In *SENS-2017 Conference*. Accessed: 2025-07-18.

Goyal, S. and Dan, S. (2025). Iolbench: Benchmarking llms on linguistic reasoning.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. (2021). Measuring massive multitask language understanding.

Ma, R., Jin, L., Liu, Q., Chen, L., and Yu, K. (2020). Addressing the polysemy problem in language modeling with attentional multi-sense embeddings. pages 8129–8133.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A.-L., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S. P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S. S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Łukasz Kaiser, Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, J. H., Kiros, J., Knight, M., Kokotajlo, D., Łukasz Kondraciuk, Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C. M., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S. M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D., Mu, T., Murati, M., Murk, O., Mély, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Ouyang, L., O'Keefe, C., Pachocki, J., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., de Avila Belbute Peres, F., Petrov, M., de Oliveira Pinto, H. P., Michael, Pokorny, Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N., Thompson, M. B., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F. C., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T., Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W., and Zoph, B. (2024). Gpt-4 technical report.

Ousidhoum, N., Beloucif, M., and Mohammad, S. M. (2025). Building better: Avoiding pitfalls in developing language resources when data is scarce.

Pakray, P., Gelbukh, A., and Bandyopadhyay, S. (2025). Natural language processing applications for low-resource languages. *Natural Language Processing*, 31(2):183–197.

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., Brown, A. R., Santoro, A., Gupta, A., Garriga-Alonso, A., Kluska, A., Lewkowycz, A., Agarwal, A., Power, A., Ray, A., Warstadt, A., Kocurek, A. W., Safaya, A., Tazarv, A., Xiang, A., Parrish, A., Nie, A., Hussain, A., Askell, A., Dsouza, A., Slone, A., Rahane, A., Iyer, A. S., Andreassen, A., Madotto, A., Santilli, A., Stuhlmüller, A., Dai, A., La, A., Lampinen, A., Zou, A., Jiang, A., Chen, A., Vuong, A., Gupta, A., Gottardi, A., Norelli, A., Venkatesh, A., Gholamidavoodi, A., Tabassum, A., Menezes, A., Kirubarajan, A., Mullokandov, A., Sabharwal, A., Herrick, A., Efrat, A., Erdem, A., Karakaş, A., Roberts, B. R., Loe, B. S.,

Zoph, B., Bojanowski, B., Özyurt, B., Hedayatnia, B., Neyshabur, B., Inden, B., Stein, B., Ekmekci, B., Lin, B. Y., Howald, B., Orinion, B., Diao, C., Dour, C., Stinson, C., Argueta, C., Ramírez, C. F., Singh, C., Rathkopf, C., Meng, C., Baral, C., Wu, C., Callison-Burch, C., Waites, C., Voigt, C., Manning, C. D., Potts, C., Ramirez, C., Rivera, C. E., Siro, C., Raffel, C., Ashcraft, C., Garbacea, C., Sileo, D., Garrette, D., Hendrycks, D., Kilman, D., Roth, D., Freeman, D., Khashabi, D., Levy, D., González, D. M., Perszyk, D., Hernandez, D., Chen, D., Ippolito, D., Gilboa, D., Dohan, D., Drakard, D., Jurgens, D., Datta, D., Ganguli, D., Emelin, D., Kleyko, D., Yuret, D., Chen, D., Tam, D., Hupkes, D., Misra, D., Buzan, D., Mollo, D. C., Yang, D., Lee, D.-H., Schrader, D., Shutova, E., Cubuk, E. D., Segal, E., Hagerman, E., Barnes, E., Donoway, E., Pavlick, E., Rodola, E., Lam, E., Chu, E., Tang, E., Erdem, E., Chang, E., Chi, E. A., Dyer, E., Jerzak, E., Kim, E., Manyasi, E. E., Zheltonozhskii, E., Xia, F., Siar, F., Martínez-Plumed, F., Happé, F., Chollet, F., Rong, F., Mishra, G., Winata, G. I., de Melo, G., Kruszewski, G., Parascandolo, G., Mariani, G., Wang, G., Jaimovitch-López, G., Betz, G., Gur-Ari, G., Galijasevic, H., Kim, H., Rashkin, H., Hajishirzi, H., Mehta, H., Bogar, H., Shevlin, H., Schütze, H., Yakura, H., Zhang, H., Wong, H. M., Ng, I., Noble, I., Jumelet, J., Geissinger, J., Kernion, J., Hilton, J., Lee, J., Fisac, J. F., Simon, J. B., Koppel, J., Zheng, J., Zou, J., Kocoń, J., Thompson, J., Wingfield, J., Kaplan, J., Radom, J., Sohl-Dickstein, J., Phang, J., Wei, J., Yosinski, J., Novikova, J., Bosscher, J., Marsh, J., Kim, J., Taal, J., Engel, J., Alabi, J., Xu, J., Song, J., Tang, J., Waweru, J., Burden, J., Miller, J., Balis, J. U., Batchelder, J., Berant, J., Frohberg, J., Rozen, J., Hernandez-Orallo, J., Boudeman, J., Guerr, J., Jones, J., Tenenbaum, J. B., Rule, J. S., Chua, J., Kanclerz, K., Livescu, K., Krauth, K., Gopalakrishnan, K., Ignatyeva, K., Markert, K., Dhole, K. D., Gimpel, K., Omondi, K., Mathewson, K., Chiafullo, K., Shkaruta, K., Shridhar, K., McDonell, K., Richardson, K., Reynolds, L., Gao, L., Zhang, L., Dugan, L., Qin, L., Contreras-Ochando, L., Morency, L.-P., Moschella, L., Lam, L., Noble, L., Schmidt, L., He, L., Colón, L. O., Metz, L., Şenel, L. K., Bosma, M., Sap, M., ter Hoeve, M., Farooqi, M., Faruqui, M., Mazeika, M., Baturan, M., Marelli, M., Maru, M., Quintana, M. J. R., Tolkiehn, M., Giulianelli, M., Lewis, M., Potthast, M., Leavitt, M. L., Hagen, M., Schubert, M., Baitemirova, M. O., Arnaud, M., McElrath, M., Yee, M. A., Cohen, M., Gu, M., Ivanitskiy, M., Starritt, M., Strube, M., Swędrowski, M., Bevilacqua, M., Yasunaga, M., Kale, M., Cain, M., Xu, M., Suzgun, M., Walker, M., Tiwari, M., Bansal, M., Aminnaseri, M., Geva, M., Gheini, M., T, M. V., Peng, N., Chi, N. A., Lee, N., Krakover, N. G.-A., Cameron, N., Roberts, N., Doiron, N., Martinez, N., Nangia, N., Deckers, N., Muennighoff, N., Keskar, N. S., Iyer, N. S., Constant, N., Fiedel, N., Wen, N., Zhang, O., Agha, O., Elbaghdadi, O., Levy, O., Evans, O., Casares, P. A. M., Doshi, P., Fung, P., Liang, P. P., Vicol, P., Alipoormolabashi, P., Liao, P., Liang, P., Chang, P., Eckersley, P., Htut, P. M., Hwang, P., Miłkowski, P., Patil, P., Pezeshkpour, P., Oli, P., Mei, Q., Lyu, Q., Chen, Q., Banjade, R., Rudolph, R. E., Gabriel, R., Habacker, R., Risco, R., Millière, R., Garg, R., Barnes, R., Saurous, R. A., Arakawa, R., Raymaekers, R., Frank, R., Sikand, R., Novak, R., Sitelew, R., LeBras, R., Liu, R., Jacobs, R., Zhang, R., Salakhutdinov, R., Chi, R., Lee, R., Stovall, R., Teehan, R., Yang, R., Singh, S., Mohammad, S. M., Anand, S., Dillavou, S., Shleifer, S., Wiseman, S., Gruetter, S., Bowman, S. R., Schoenholz, S. S., Han, S., Kwatra, S., Rous, S. A., Ghazarian, S., Ghosh, S., Casey, S., Bischoff, S., Gehrmann, S., Schuster, S., Sadeghi, S., Hamdan, S., Zhou, S., Srivastava, S., Shi, S., Singh, S., Asaadi, S., Gu, S. S., Pachchigar, S., Toshniwal, S., Upadhyay, S., Shyamolima, Debnath, Shakeri, S., Thormeyer, S., Melzi, S., Reddy, S., Makini, S. P., Lee, S.-H., Torene, S., Hatwar, S., Dehaene, S., Divic, S., Ermon, S., Biderman, S., Lin, S., Prasad, S., Piantadosi, S. T., Shieber, S. M., Misherghi, S., Kiritchenko, S., Mishra, S., Linzen, T., Schuster, T., Li, T., Yu, T., Ali, T., Hashimoto, T., Wu, T.-L., Desbordes, T., Rothschild, T., Phan, T., Wang, T., Nkinyili, T., Schick, T., Kornev, T., Tunduny, T., Gerstenberg, T., Chang, T., Neeraj, T., Khot, T., Shultz, T., Shaham, U., Misra, V., Demberg, V., Nyamai, V., Raunak, V., Ramasesh, V., Prabhu, V. U., Padmakumar, V., Srikumar, V., Fedus, W., Saunders, W., Zhang, W., Vossen, W., Ren, X., Tong, X., Zhao, X., Wu, X., Shen, X., Yaghoobzadeh, Y., Lakretz, Y., Song, Y., Bahri, Y., Choi, Y., Yang, Y., Hao, Y., Chen, Y., Belinkov, Y., Hou, Y., Hou, Y., Bai, Y., Seid, Z., Zhao, Z., Wang, Z., Wang, Z. J., Wang, Z., and Wu, Z. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models.

Sánchez, E., Alastruey, B., Ropers, C., Stenetorp, P., Artetxe, M., and Costa-jussà, M. R. (2024). Linguini: A benchmark for language-agnostic linguistic reasoning.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). Glue: A multi-task benchmark and analysis platform for natural language understanding.

Slur and Emoji Aware Models for Hate and Sentiment Detection in Roman Urdu Transgender Discourse

Muhammad Owais Raza
Department of Computer Engineering
Istanbul Sabahattin Zaim University
Istanbul 34303, Turkey
6210024002@std.izu.edu.tr

Aqsa
Sindh Madressatul Islam University
City Campus, Karachi
aqsa.umar3505@gmail.com

Mehrub Awan
Gender Interactive Alliance
mehrubmoizawan@gmail.com

Abstract

The rise of social media has amplified both the visibility and vulnerability of marginalized communities, such as the transgender population in South Asia. While hate speech detection has seen considerable progress in high resource languages like English, under-resourced and code mixed languages such as Roman Urdu remain significantly understudied. This paper presents a novel Roman Urdu dataset derived from Instagram comments on transgender related content, capturing the intricacies of multilingual, code-mixed, and emoji-laden social discourse. We introduce a transphobic slur lexicon specific to Roman Urdu and a semantic emoji classification grounded in contextual usage. These resources are utilized to perform fine-grained classification of sentiment and hate speech using both traditional machine learning models and transformer-based architectures. The findings show that our custom-trained BERT-based models, Senti-RU-Bert and Hate-RU-Bert, show the best performance, with F1 scores of 80.39% for sentiment classification and 77.34% for hate speech classification. Ablation studies reveal consistent performance gains when slur and emoji features are included.

1 Introduction

In recent years, there has been a significant increase in social networks and content consumption. What people share on social media platforms has a direct impact on their daily lives (Aziz et al., 2023). Social media has become a powerful platform for sharing ideas and perspectives; however, it is also increasingly misused to spread hate against individuals, groups, and communities. Such content poses serious threats to social harmony, online safety, and mental health (Sharma et al., 2025), contributing to the alarming rise in hate speech.

Hateful speech is a type of language which conveys the negative sentiment that can shame users while also promoting radicalism and inciting violence (Gitari et al., 2015). Hate speech is a growing issue, especially in the comment sections of social media platforms such as Instagram, Facebook, and YouTube. Several approaches to categorizing hate speech on social media platforms were investigated in the study (Martins et al., 2018). They presented a combination of machine learning and lexicon-based methods for predicting hate speech. Notably, they employed emotional content in sentences to improve the detection accuracy of hate speech.

While hate speech detection in English (Shahi and Majchrzak, 2024; Pan et al., 2024; Gandhi et al., 2024) have been widely studied, low resource South Asian languages remain significantly under-explored, leaving millions of social media users vulnerable. In Pakistan, for instance, around 26% of the population is bilingual, leading to frequent code mixing in user generated content (Aziz et al., 2023). This linguistic diversity often results in text that blends English script with local languages such as Roman Urdu and Roman Sindhi posing unique challenges for automated detection systems. Addressing this gap requires the development of dedicated resources for these underrepresented languages.

Although sentiment analysis in Roman Urdu has received some attention in prior work, substantial limitations persist. Several Roman Urdu datasets are publicly available, and previous studies have focused on unsupervised lexical normalization (Mehmood et al., 2020b) and the impact of lexical variation on sentiment classification (Manzoor et al., 2020). Notable efforts include the Roman Urdu E-commerce Dataset (RUECD), which comprises 26,824 customer reviews from DarazPK (Chandio et al., 2022), a binary-labeled dataset of 11,000 reviews across six domains (Mehmood et al., 2019), and a third dataset with 3,241 annotated attitudes (Mehmood et al., 2020a). Despite these

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025,
pages 131–139, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-014

efforts, Roman Urdu remains a low-resource language, and progress in sentiment and hate speech detection is hindered by a lack of standardized linguistic tools and comprehensive annotated datasets (Khan et al., 2024). This scarcity highlights the pressing need for further research and resource development to support natural language understanding in code-mixed, under-resourced settings.

Existing studies on Urdu language television dramas reveal a distorted portrayal of transgender characters, often exaggerating dominance and neglecting economic violence while emphasizing psychological abuse (Mehmood et al., 2020b; Manzoor et al., 2020; Ullah et al., 2024). Despite such media shaping cultural discourse, no curated Roman Urdu dataset exists that reflects the unique linguistic patterns and hate speech dynamics in social media discussions about transgender people in Pakistan. This underscores an urgent need to build specialized Roman Urdu datasets for hate speech detection that can more accurately capture the sociolinguistic realities of digital discourse in underrepresented languages and communities.

In this study, we address the underexplored issue of hate speech and sentiment analysis in Roman Urdu social media discourse, particularly focusing on transgender related content. We construct a novel dataset of Instagram comments that reflects real-world multilingual and code-mixed usage, enriched with emojis and varying sentiment. To capture the specific linguistic and multimodal nuances of online hate, we also develop a Roman Urdu transphobic slur lexicon and a context aware emoji classification. These resources enable deeper analysis of abuse patterns and facilitate fine-grained feature extraction. Building on these contributions, we perform a thorough evaluation of traditional machine learning models and transformer-based architectures, comparing their performance across different combinations of textual, lexical, and emoji-based features for both sentiment and hate speech classification tasks. This comprehensive approach highlights the complexities of detecting hate in low resource, code-mixed contexts and offers tools to improve detection and understanding in similar sociolinguistic settings. Following are three main contribution of this study:

- Creation of a novel, Roman Urdu dataset from Instagram comments on transgender related content, incorporating multilingual and emoji-inclusive text.
- Development of a Roman Urdu based transphobic slur lexicon and an emoji classification, enabling fine grained linguistic and multimodal feature enrichment.
- Conducted a comprehensive evaluation of ML and transformer models using text, emoji, and slur features for sentiment and hate speech classification.

2 Dataset

The dataset used in this study was collected from the Instagram comment sections of Binax Studio[1], a social media page focusing on transgender oriented content in Pakistan. The dataset comprises comments written in multiple languages, including Roman Urdu, Urdu (Arabic script), Hindi (Devanagari), Arabic, English and a high frequency of emojis. Given their dominance in the dataset, we retained only comments written in Roman Urdu or code-mixed Roman Urdu (i.e., Roman Urdu combined with English or emojis), as well as purely emoji based expressions, for further analysis. A zero-shot learning approach with human feedback was employed to label the comments. A typical prompt provided to the model was:

> "Analyze the following text and return JSON with exactly these keys: - sentiment: one of 'positive', 'neutral', or 'negative' - negative_type: if sentiment is 'negative', classify it as 'abusive', 'threatening', 'call for action', or 'other'. Else null. - language: detect the language or script used, like 'Urdu', 'Roman Urdu', 'English', etc. Consider emoji and transliteration."

For reliability, two bilingual authors (fluent in Urdu, Roman Urdu, and English) independently examined 200 model labeled comments. Substantial agreement was achieved with the model's assignments (Cohen's $\kappa = 0.75$), and any differences were settled through discussion, which informed adjustments to the labeling guidelines.

Table 1 summarizes the distribution of content types in the dataset. A majority of the comments (1584) were written in Roman Urdu without emojis, followed closely by 1282 comments that combined

[1] https://www.instagram.com/binax.studio

Roman Urdu with emojis. Additionally, 294 comments consisted entirely of emojis. The dataset was annotated for two key tasks:

Table 1: Distribution of Content Types with Examples

Content Type	Count	Example
Roman Urdu	1584	Ab khusron ky bhi podcast hongy?
Roman Urdu + Emoji	1282	Apko dekh kr ma kya hi judge krunga 😭
Emoji Only	294	😂😂😂😂😂
Total	3160	–

2.1 Task 1: Sentiment Analysis

The first task involved classifying each comment into one of three sentiment categories: Positive, Neutral, or Negative. This task captures the overall emotional tone of public engagement with transgender content (Mao et al., 2024).

- **Positive:** Comments that express support, admiration, or encouragement.

  Example: "Jaan ho 😍"

  Translation: "You are my life"

- **Neutral:** Comments that are factual, descriptive, or unclear in tone.

  Example: "yaar yeh q ka mtlv kya hai"

  Translation: "Hey, what does this mean?"

- **Negative:** Comments expressing disapproval, mockery, or hostility.

  Example: "Saleh hijre"

  Translation: "Damn eunuch"

2.2 Task 2: Hate Speech Classification

The second task involved classifying hate speech present in the comments into four categories: Abusive, Threat, Call for Action, and Other. This categorization aims to capture different intensities and types of harmful speech targeting transgender individuals (Kumar et al., 2025).

- **Abusive:** Insults, slurs, or dehumanizing language directed at transgender people.

  Example: "Tum dono ki MKC"

  Translation: "You both are motherf***ers" (highly offensive)

- **Threat:** Explicit or implicit threats of violence or harm.

  Example: "The scream at the end will be u screaming when i will execute u. Fitnah phelate ho tum loug. JAHANAMMI!!"

  Translation: "You spread chaos. The scream at the end will be yours when I execute you. You are hell-bound!!"

- **Call for Action:** Urging others to act against transgender individuals, including boycotts or violence.

  Example: "Report karo inko"

  Translation: "Report them"

- **Other:** Includes misgendering, sarcastic derision, or religiously framed delegitimization that doesn't fit the above but still contributes to a hostile environment.

  Example: "Ouch"

  Translation: Same in English; sarcastic or mocking tone.

Table 2 presents the distribution of sentiment and hate speech categories within the filtered dataset. Out of 3160 comments, the majority (1603) expressed negative sentiment, followed by 966 positive and 591 neutral comments. In terms of hate speech, 1225 comments were labeled as abusive, while a smaller number fell into the categories of other (339), threatening (22), and call for action (17).

Table 2: Distribution of Sentiment and Hate Speech Categories

Sentiment	Count	Hate Speech Type	Count
Negative	1603	Abusive	1225
Positive	966	Other	339
Neutral	591	Threatening	22
–	–	Call for Action	17

3 Linguistic Analysis

This section explores the linguistic features and affective cues used in online discourse targeting transgender individuals, particularly in Roman Urdu. We focus on two key dimensions: (1) lexical abuse through Roman Urdu slurs, and (2) the affective and rhetorical functions of emojis in these hostile or supportive comments. Together, they reveal how language and visual symbols convey support, mockery, identity assertion, or threats.

3.1 Emoji Classification Design

To analyze the role of emojis in transgender-related discourse, we developed a custom taxonomy based on contextual usage rather than Unicode semantics. As shown in Figure 1, we identified ten functional categories: Supportive/Affective (affection, solidarity), Mocking/Dismissive (ridicule, sarcasm), Identity Pride (queer or gender identity markers), Aggressive/Threatening (symbolic violence, hostility), Gesture/Emphasis (tone amplification), Religious/Moral (judgment) Humor/Ambiguous (irony, camp), Sadness/Vulnerability (grief, helplessness), Body/Gendered (physical or sexed features), and Mock Femininity (caricatured feminized traits). Although each emoji was placed in one main category based on its context, in rare cases, the same emoji could fit into more than one category depending on how it was used. Building on the taxonomy, Figure 2 presents the distribution of emoji usage across the dataset. The most frequent category was Mocking (38.76%), highlighting the prevalence of ridicule and sarcasm. This was followed by Supportive/Affective (27.33%), indicating a substantial presence of emotional solidarity. Mock Femininity (8.84%) and Aggressive/Threatening (7.51%) emojis also appeared prominently, often signaling coded transphobia or symbolic hostility.

3.2 Construction of Transphobic Slur Lexicon for Roman Urdu

We constructed a domain specific lexicon by identifying 124 unique slurs from the comment dataset. This lexicon includes both explicit terms and more implicit or coded expressions used in South Asian digital discourse (e.g., chakka, khusra, etc). The lexicon construction was in done in following 4 steps.

3.2.1 Orthographic Normalization:

To consolidate orthographic variants of abusive terms, we applied phoneme aware normalization rules that convert common Roman Urdu digraphs and vowel elongations into base forms (e.g., *"gandoo", "gaanduu"* $\rightarrow$ *gandu*). This step was implemented using regular expressions and phonetic substitution rules that we manually created to handle common Roman Urdu spelling variants (e.g.,"aa"/"a"). As no standardized resources exist for Roman Urdu normalization, our rules were iteratively refined through manual inspection of sample outputs. To group closely related variants and misspellings, tokens were clustered based on Levenshtein edit distance (threshold ≤ 2). This approach ensured that minor typographical variations were treated as a single lexeme.

3.2.2 Token Filtering and Frequency Thresholding:

We removed stopwords using an expanded bilingual stopword list covering both Roman Urdu and English function words. Tokens with a frequency less than three were discarded to focus on commonly used terms.

3.2.3 Manual Filtering:

After converting minor typographical variations into single lexemes, we obtained 664 tokens. These were manually reviewed to remove ambiguous or contextually irrelevant words, resulting in a final curated lexicon of 124 semantically abusive and transphobic slurs.

Table 3 shows the most common abusive and transphobic slurs found in our dataset. These slurs frequently appeared in user comments and were retained in our final lexicon after normalization and manual review. Many of the terms, such as `bc` and `bkl`, are abbreviations or phonetic spellings, but within the comment context, they clearly function as tools of verbal abuse. While this data is specific to our collected dataset, it reflects a wider trend of how such harmful language is casually and repeatedly used in online conversations related to transgender topics in the South Asian social media space.

Table 3: Most common slurs in the dataset based on our Roman Urdu Transphobic Slur lexicon

Normalized	Count	Description
`bc`	58	Abbreviation of incest-based Urdu profanity
`khusra`	57	Slur for transgender person (Roman Urdu for *Khusra*)
`gand`	44	Vulgar term for buttocks (Roman Urdu)
`bkl`	32	Abbreviation for "idiot" or "stupid" in abuse contexts
`gandu`	24	Used in the context of abuse to call someone F*gg*t

Table 4 shows the most common co-occurrences

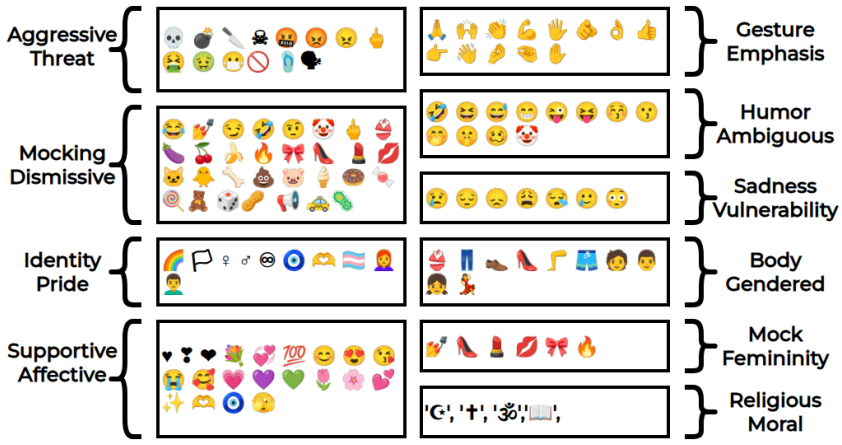


Figure 1: Proposed Emoji Classification

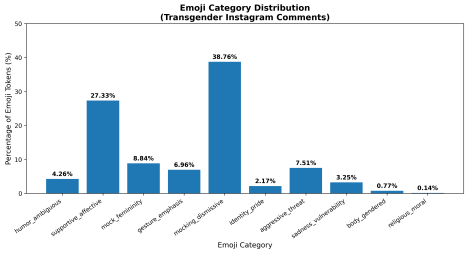


Figure 2: Emoji Distribution Based on Proposed Emoji Classification

between transphobic slurs and emojis within Roman Urdu comments. These combinations were extracted from our dataset and reveal how users often use affective emojis especially the 😂 (joy) emoji alongside abusive terms like *gand*, *bc*, and *chakka*. The use of such emojis tends to amplify mockery, express sarcasm, or downplay the abuse, making it seem more casual. This highlights an important multimodal layer of hate speech and shows why emoji signals should be included in models that detect implicit or indirect abuse.

Table 4: Top Slur Emoji Co-occurrences by Frequency

Slur (Description)	Emoji	Count
gand (buttocks)	😂	29
bc (incest profanity)	😂	19
chakka (trans slur)	😂	18
lan (penis)	😂	15
chod (f**k)	😂	14

Table 5 presents examples of how emojis frequently appear alongside transphobic or vulgar slurs in Roman Urdu social media comments. Each row includes a commonly used slur, its co-occurring emoji, the frequency of this pairing, and an anonymized usage example translated into English. Emojis like 😂 (laughter) or 💀 (mockery) often shift the emotional tone either by making the abuse sound humorous, sarcastic, or more intense. These combinations reflect a deliberate use of visual cues to strengthen or mask hateful intent, making the abuse seem more socially acceptable or performative. The findings emphasize the need to treat emojis not just as add-ons, but as important features in detecting and understanding online hate speech.

Table 5: Examples of Emoji Usage in Slur Contexts

Slur	Emoji	Count	Example
gando	😂	29	*"Gando log* 😂*"* "F*gg*t people" with mocking laughter.
lanat	✋	5	*"Lanat* ✋✋*"* "Shame/curse" with repeated hand gestures.
chutiya	😢	3	*"Chutiya promoting nudity* 😢*"* Sarcastic or mocking tone.
bc	💀	3	*"Chaako ki podcast are bc"* "Chaako" is a derogatory slur for transgender people, and "bc" is a strong abusive term

4 Experimental Settings

We conducted comprehensive experiments for two classification tasks: (i) Sentiment analysis (positive,

neutral, negative), and (ii) Hate speech classification (e.g., abusive, threatening, etc.). Each task was evaluated using both traditional machine learning models and fine tuned transformer based model.

4.1 Dataset Preprocessing and Feature Engineering

We have dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where each x_i is a instance of the dataset (i.e. comment on an Instagram post) and $y_i \in \mathcal{Y}$ is the corresponding label (for either sentiment or hate speech category), we augmented the inputs with structured linguistic and contextual features to capture additional social and semantic cues beyond raw text.

Each input x_i was decomposed into its base text t_i, a categorical emoji feature $e_i \in \mathcal{E}$, and a binary indicator $s_i \in \{0, 1\}$ for the presence of slurs. The slur flag s_i is computed based on a manually curated lexicon lex_{slur} of 124 normalized Roman Urdu transgender related slurs. The flag is defined as:

$$s_i = \begin{cases} 1 & \text{if } \exists\, w \in \text{lex}_{\text{slur}} w \in \text{Tok}(t_i) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In equation 1 $\text{Tok}(t_i)$ denotes the tokenized form of the comment t_i. The emoji feature e_i was derived by mapping each emoji in t_i to a predefined semantic category (e.g., *mocking*, *affective*, *gesture*, *threat*).

To systematically evaluate the contribution of each feature type, we defined four feature sets: (i) $\mathcal{F}_{\text{text}} = \{t_i\}$, representing raw text only; (ii) $\mathcal{F}_{\text{text+emoji}} = \{t_i, e_i\}$, which includes emoji category alongside text; (iii) $\mathcal{F}_{\text{text+slur}} = \{t_i, s_i\}$, incorporating slur presence; and (iv) $\mathcal{F}_{\text{all}} = \{t_i, e_i, s_i\}$, the full feature combination. These sets were used across experiments to assess the role of lexical and paralinguistic features under controlled comparisons.

4.2 Modeling and Evaluation

Once dataset preprocessing and feature engineering is done, the next step is perform machine learning modeling and evaluation across both sentiment analysis and hate speech type classification tasks. We implemented two modeling approaches: one based on traditional machine learning (ML) algorithms with bag of words representations, and another using a custom fine tuned BERT (Devlin et al., 2019) based transformer model. For consistency each model was trained and tested using a stratified 80/20 train-test split with fixed random seed to ensure reproducibility.

4.2.1 Traditional Machine Learning Models

For classical approaches, we trained a diverse set of supervised classifiers: Logistic Regression (LogReg), Linear Support Vector Machine (LinearSVC), Multinomial Naive Bayes (MultinomialNB), Random Forest, XGBoost, and LightGBM. In this modeling approach the textual input t_i was converted into numerical form using TF-IDF (De Santis et al., 2024) vectorization, capturing both unigrams and bigrams (n-gram range = (1, 2)) and restricted to a maximum of 5000 features. Emoji categories $e_i \in \mathcal{E}$ were encoded using one-hot encoding, and the slur flag $s_i \in \{0, 1\}$ was passed through as a numeric binary feature.

4.2.2 Transformer Based Model

To model context and semantic nuance directly from raw text, we trained a task specific transformer classifier based on BERT (bert-base-uncased). Input text was tokenized using the corresponding WordPiece tokenizer (Schönle et al., 2024), subword information, and special tokens. Each input sequence was truncated or padded to a maximum length of 128 tokens and passed to the BERT encoder. The transformer model was implemented using the HuggingFace Transformers library[2]. All experiments were conducted with a batch size of 8, 3 epochs of training with 50 logging steps.

4.2.3 Model Evaluation

Both traditional machine learning models and the transformer based model were evaluated using standard classification metrics: Accuracy, Precision, Recall, and F1 Score. All metrics were computed using weighted averaging, which accounts for the distribution of samples across classes and ensures that minority classes are not ignored during evaluation. To ensure a fair and consistent comparison across all models and feature combinations, the dataset was partitioned into 80% training and 20% testing subsets using stratified sampling, preserving the original class proportions in both splits.

5 Results

This section presents the performance outcomes of various models applied to two primary tasks:

[2] https://huggingface.co/

Sentiment Classification and Negativity Type (hate speech) Classification. We evaluate both traditional machine learning classifiers and a transformer-based BERT model using four different feature configurations: text only (T), text with emoji features (T+E), text with slur features (T+S), and all features combined (T+E+S). For experiments including emojis (E) and slur flags (S), these features were concatenated as auxiliary inputs to the BERT classifier's final hidden layer before the classification head. The evaluation metrics include Accuracy, Precision, Recall, and F1 Score. All experiments were conducted on Google Colab with a 16 GB RAM and 128 GB disk environment.

Table 6 presents the comparative performance of multiple models across two classification tasks: Sentiment Detection and Hate speech Classification. All models were evaluated using a unified feature set comprising textual (T), emoji features (E), and slur features (S). Among traditional machine learning models, LinearSVC and Random Forest demonstrated relatively strong and consistent performance. However, BERT-based models (Senti-RU-BERT and Hate-RU-BERT) outperformed all baselines across both tasks, achieving the highest accuracy (80.54% for sentiment, 78.01% for negativity type) and F1 scores (80.39% and 77.34% respectively). Our domain specific variants, Senti-RU-BERT and Hate-RU-BERT, fine-tuned on Roman Urdu data with emoji and slur context, yielded the best results. These findings shows the effectiveness of contextual embeddings in capturing the nuanced affective and abusive signals in under-resourced, code-mixed languages especially when enhanced with multimodal cues.

Table 6: Performance (Accuracy, Precision, Recall, F1) for Sentiment and Hate Speech Classification

Task	Model	Acc.	Prec.	Rec.	F1
Sentiment	LinearSVC	74.37	74.77	74.37	74.54
	LogReg	74.21	73.51	74.21	73.09
	RandomForest	73.26	73.72	73.26	73.46
	XGBoost	71.99	73.62	71.99	72.60
	MultinomialNB	68.83	74.00	68.83	63.60
	LightGBM	68.04	70.11	68.04	68.77
	Senti-RU-BERT	**80.54**	**80.29**	**80.54**	**80.39**
Negativity	LinearSVC	72.15	71.21	72.15	71.19
	LogReg	71.68	71.11	71.68	70.17
	RandomForest	74.21	74.43	74.21	72.86
	XGBoost	71.20	69.85	71.20	70.36
	MultinomialNB	71.36	72.20	71.36	69.86
	LightGBM	68.67	67.99	68.67	67.83
	Hate-RU-BERT	**78.01**	**76.75**	**78.01**	**77.34**

Table 7 presents a detailed ablation study reporting F1 scores for different combinations of input features Text only (T), Text + Emoji features (T+E), Text + Slur features (T+S), and all combined (T+E+S) across various models for both sentiment and negativity classification. For sentiment classification, the best results were consistently achieved with the T+E+S combination. Traditional models like LinearSVC (74.54), Logistic Regression (73.09), and Random Forest (73.46) showed marked improvements over their text-only baselines (66.74, 64.94, and 66.79 respectively). Notably, Senti-RU-BERT attained the highest F1 score of 80.77 with the T+E setting, slightly outperforming the T+E+S score (80.39), suggesting that emojis alone contributed more than slurs in this transformer model. Among traditional models, XGBoost (72.60) and LightGBM (68.77) also benefited from feature fusion, while MultinomialNB saw the greatest relative gain jumping from 45.75 (T) to 63.60 (T+E+S). In the case of hate speech type classification, results were more nuanced. The highest F1 score was obtained by Hate-RU-BERT with 77.64 using T+S, showing that slur features were particularly informative for this task. Among classical models, Random Forest (73.88) and Logistic Regression (72.45) also achieved their top scores using T+S, indicating the value of slur-based lexical signals for distinguishing hate subtypes. Interestingly, unlike the sentiment task, adding emojis (T+E) showed limited or no improvement here. The overall best traditional performance came from Random Forest with T+S (73.88), while MultinomialNB, though less effective overall, reached 72.03 with T+S its peak performance across all configurations. These results validate that incorporating task specific linguistic signals such as emojis for sentiment and slurs for hate subtype detection improves classifier performance.

6 Limitations

Despite achieving strong performance on both sentiment and hate speech classification tasks, several limitations persist. First, the class imbalance, especially in the hate speech subtype categories (e.g., threatening, call for action), may bias the models toward more frequent classes like abusive. This limits generalizability and reduces sensitivity to underrepresented categories. Second, many instances of hate speech in our dataset exhibit subtle or context-dependent abuse, including sarcasm,

Table 7: Ablation Study: F1 Score Comparison Across Feature Sets and Models

Features	LinearSVC	LogReg	RandomForest	XGBoost	MultinomialNB	LightGBM	Senti-RU-BERT / Hate-RU-BERT
Sentiment							
T	66.74	64.94	66.79	64.41	45.75	54.09	72.65
T+E	73.25	71.10	71.36	68.76	62.52	63.28	**80.77**
T+S	68.00	66.82	68.62	66.75	46.83	57.85	74.31
T+E+S	74.54	73.09	73.46	72.60	63.60	68.77	**80.39**
Negativity Type							
T	70.70	69.54	70.21	68.58	69.22	60.71	76.14
T+E	70.76	69.44	69.53	68.49	68.36	61.21	76.82
T+S	72.42	72.45	73.88	69.46	72.03	68.06	**77.64**
T+E+S	71.19	70.17	72.86	70.36	69.86	67.83	**77.34**

code switching, or rhetorical phrasing, which remain challenging for both traditional models and BERT. Although auxiliary features like slur flags and emoji categories improve performance, they cannot fully capture nuanced socio-pragmatic cues.

7 Future Work

Future work can focus on curating more balanced and culturally grounded datasets for Roman Urdu and code-switched text, incorporating linguistic annotation informed by sociolinguistic cues to better detect subtle or indirect expressions of hate speech. Further to deal with class imbalance the dataset should be curated in such a way that there is balance between both overrepresented and underrepresented classes such as such as threatening or call for action. This can also be achieved through targeted data augmentation techniques like paraphrasing, back translation, or synthetic oversampling. Future work could also explore applying more recent large language models (e.g., GPT or open-source alternatives).

8 Conclusion

This study presents a novel computational approach to analyzing online discourse concerning transgender communities in Pakistan, with a particular emphasis on Roman Urdu—a low-resource, code-mixed language prevalent across social media platforms. We introduce a comprehensive Instagram-based dataset annotated for both sentiment and hate speech, and further enrich this resource through the development of a Roman Urdu transphobic slur lexicon and an emoji classification grounded in contextual semantics. Experimental evaluations reveal that transformer-based architectures, notably BERT, consistently outperform traditional machine learning models on both classification tasks, achieving F1 scores of 80.39% for sentiment and 77.34% for hate speech detection. Ablation analyses demonstrate that the integration of lexicon-based and emoji-derived features yields significant performance improvements, especially in identifying implicit or nuanced forms of hate speech. These findings highlight the critical role of culturally and linguistically informed resources in advancing hate speech detection in low-resource settings. By integrating domain-specific linguistic insights with state-of-the-art natural language processing techniques, this work establishes foundational tools, benchmarks, and methodologies for future research in socially-aware, multilingual, and inclusive NLP.

References

Samia Aziz, Muhammad Shahzad Sarfraz, Muhammad Usman, Muhammad Umar Aftab, and Hafiz Tayyab Rauf. 2023. Geo-spatial mapping of hate speech prediction in roman urdu. *Mathematics*, 11(4):969.

Bilal Chandio, Asadullah Shaikh, Maheen Bakhtyar, Mesfer Alrizq, Junaid Baber, Adel Sulaiman, Adel Rajab, and Waheed Noor. 2022. Sentiment analysis of roman urdu on e-commerce reviews using machine learning. *CMES-Comput. Model. Eng. Sci*, 131(3):1263–1287.

Enrico De Santis, Alessio Martino, Francesca Ronci, and Antonello Rizzi. 2024. From bag-of-words to transformers: A comparative study for text classification in healthcare discussions in social media. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

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.

Ankita Gandhi, Param Ahir, Kinjal Adhvaryu, Pooja Shah, Ritika Lohiya, Erik Cambria, Soujanya Poria, and Amir Hussain. 2024. Hate speech detection: A comprehensive review of recent works. *Expert Systems*, 41(8):e13562.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

Marwa Khan, Asma Naseer, Aamir Wali, and Maria Tamoor. 2024. A roman urdu corpus for sentiment analysis. *The Computer Journal*, 67(9):2864–2876.

Mohit Kumar et al. 2025. Exploring hate speech detection: challenges, resources, current research and future directions. *Multimedia Tools and Applications*, pages 1–37.

Muhammad Arslan Manzoor, Saqib Mamoon, Song Kei Tao, Zakir Ali, Muhammad Adil, and Jianfeng Lu. 2020. Lexical variation and sentiment analysis of roman urdu sentences with deep neural networks. *Int. J. Adv. Comput. Sci. Appl*, 11(2).

Yanying Mao, Qun Liu, and Yu Zhang. 2024. Sentiment analysis methods, applications, and challenges: A systematic literature review. *Journal of King Saud University-Computer and Information Sciences*, 36(4):102048.

Ricardo Martins, Marco Gomes, Jose Joao Almeida, Paulo Novais, and Pedro Henriques. 2018. Hate speech classification in social media using emotional analysis. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 61–66. IEEE.

Faiza Mehmood, Muhammad Usman Ghani, Muhammad Ali Ibrahim, Rehab Shahzadi, Waqar Mahmood, and Muhammad Nabeel Asim. 2020a. A precisely xtreme-multi channel hybrid approach for roman urdu sentiment analysis. *IEEE Access*, 8:192740–192759.

Khawar Mehmood, Daryl Essam, Kamran Shafi, and Muhammad Kamran Malik. 2019. Sentiment analysis for a resource poor language—roman urdu. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(1):1–15.

Khawar Mehmood, Daryl Essam, Kamran Shafi, and Muhammad Kamran Malik. 2020b. An unsupervised lexical normalization for roman hindi and urdu sentiment analysis. *Information Processing & Management*, 57(6):102368.

Ronghao Pan, José Antonio García-Díaz, and Rafael Valencia-García. 2024. Comparing fine-tuning, zero and few-shot strategies with large language models in hate speech detection in english. *CMES-Computer Modeling in Engineering & Sciences*, 140(3).

Daniel Schönle, Christoph Reich, and Djaffar Ould Abdeslam. 2024. Linguistic-aware wordpiece tokenization: Semantic enrichment and oov mitigation. In *2024 6th International Conference on Natural Language Processing (ICNLP)*, pages 134–142. IEEE.

Gautam Kishore Shahi and Tim A Majchrzak. 2024. Hate speech detection using cross-platform social media data in english and german language. *arXiv preprint arXiv:2410.05287*.

Deepawali Sharma, Tanusree Nath, Vedika Gupta, and Vivek Kumar Singh. 2025. Hate speech detection research in south asian languages: a survey of tasks, datasets and methods. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(3):1–44.

Wali Ullah, Robina Saeed, and Nadir Saeed Sarhadi. 2024. Portrayal of violence against transgender in pakistani urdu dramas: A critical analysis. *Annals of Human and Social Sciences*, 5(4):250–263.

Automatic Fact-checking in English and Telugu

Ravi Kiran Chikkala[1,2] **Tatiana Anikina**[3] **Natalia Skachkova**[3]
Ivan Vykopal[4,5] **Rodrigo Agerri**[2] **Josef van Genabith**[3]
[1] Saarland University
[2] University of the Basque Country
[3] German Research Center for Artificial Intelligence, Saarland Informatics Campus
[4] Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
[5] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
rach00004@teams.uni-saarland.de
{tatiana.anikina,natalia.skachkova,josef.van_genabith}@dfki.de
ivan.vykopal@kinit.sk
rodrigo.agerri@ehu.eus

Abstract

False information poses a significant global challenge, and manually verifying claims is a time-consuming and resource-intensive process. In this research paper, we experiment with different approaches to investigate the effectiveness of large language models (LLMs) in classifying factual claims by their veracity and generating justifications in English and Telugu. The key contributions of this work include the creation of a bilingual English-Telugu dataset and the benchmarking of different veracity classification approaches based on LLMs.

1 Introduction

In today's technological world, claim verification plays an important role (Zhang and Gao, 2023), which aims to assess the veracity of claims as "true" or "false" by validating them against trustworthy sources (Panchendrarajan and Zubiaga, 2024). This is necessary to combat false information, especially in multilingual countries such as India, where false information can be propagated in multiple languages via translation technology (Quelle et al., 2025). According to Pradeep et al. (2021), claim verification involves three key steps: (1) retrieval of documents, (2) rationale selection, and (3) label prediction. Currently, multilingual LLMs significantly improve the claim verification process (Schlichtkrull et al., 2023) compared to traditional approaches such as manual fact-checking and simple machine learning classifiers. These language models not only evaluate claims, but also provide justifications, thereby offering a level of explanation that traditional natural language processing (NLP) approaches often lack (Dmonte et al., 2024). To date, most of the work on claim verification in fact-checking has been performed in English. In this work, we address this shortcoming by creating a new fact-checking dataset in Telugu, allowing for large-scale experimentation in Telugu, a language spoken by over 200 million people in the world (Mallareddy, 2012). We achieve this by translating our manually created English dataset into Telugu, resulting in a bilingual English–Telugu dataset that supports multilingual claim verification. Furthermore, LLMs pose several limitations, such as tendencies to hallucinate (Li et al., 2024), they exhibit biases (Lin et al., 2025), smaller models may operate within limited context windows (Ratner et al., 2023), and models may rely on knowledge that may be outdated due to cutoff dates (Cheng et al., 2024). In order to address these challenges, we use Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) with different components, such as prompt compression (Li et al., 2025), document re-ranking (Hui et al., 2022) and query rewriting (Ma et al., 2023).

We explore two research questions.

> ***RQ1:*** How well do LLMs classify domain-specific claims in English versus Telugu?
>
> ***RQ2:*** How do different models and approaches impact the quality of justifications provided by LLMs in a English-Telugu multilingual setting?

To address these research questions, we introduce a new dataset named **Preethi**[1] that covers both English and Telugu. Our experiments demonstrate that RAG-based approach, achieves the highest claim verification scores in both English and Telugu. For justification generation, RAG-based approach obtains the best average score for English, while Simple Prompting achieves the highest average score for Telugu.

[1]We make our complete dataset available at https://huggingface.co/datasets/Blue7Bird/Preethi_dataset

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 140–151, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-015

2 Related Work

2.1 Datasets Related to Indian Languages

Several datasets have been proposed for detecting false information in the Indian context. Sharma and Garg (2021b) introduce the Indian Fake News Dataset (IFND), a monolingual English dataset comprising 56,714 claims across various categories relevant to the Indian context. Each claim in IFND is labeled as "true" or "fake". Similarly, Gupta and Srikumar (2021) develop the X-Fact dataset, which includes 31,189 claims and supports multiple Indian languages-though not Telugu. X-Fact has five labels "true", "mostly-true", "partly-true", "mostly-false", and "false". Singhal et al. (2022) annotate the Fact Drill dataset, which comprises 22,435 false claims in 13 Indian regional languages, including fewer than 2,000 samples in Telugu. However, the dataset is not publicly available. Mittal et al. (2023) present the X-CLAIM dataset, which focuses on the identification of claims in multilingual social media posts. X-CLAIM contains 7,000 real-world claims across five Indian regional languages and English, but only 107 Telugu samples in its test set. Schlichtkrull et al. (2023) develop the AVeriTeC dataset, comprising 4,568 real-world claims in English. Each claim in AVeriTeC is classified into one of the four labels: "supported", "refuted", "not enough evidence" and "conflicting evidence/cherry-picking". Raja et al. (2023) create the Dravidian Fake News Dataset (DFND), which consists of 26,000 news articles in Telugu, Tamil, Kannada, and Malayalam, annotated with binary labels: "true" or "fake". However, the DFND is not open source, which poses challenges for reproducibility and further research.

Although some of these datasets support claim verification to varying degrees in Indian languages, none, except AVeriTeC, include human-annotated justifications and Question Answer (QA) pairs. Yet AVeriTeC is not designed for the Indian context. This highlights a research gap: the absence of open source, human-annotated QA pairs, and justification-rich resources for misinformation detection in low-resource Indian languages such as Telugu for the Indian context.

2.2 RAG and Other Approaches with LLMs

Recent advances in claim verification have used LLMs and RAG frameworks for claim verification processes (Dmonte et al., 2024). Singal et al. (2024) develop a RAG pipeline that extracts relevant evidence sentences from a knowledge base, which are then passed into an LLM for classification. Yue et al. (2024) introduce a Retrieval-Augmented Fact Verification framework through the synthesis of contrasting arguments (RAFTS) to determine the veracity of the claim. Katranidis and Barany (2024) propose Facts as a Function approach (FaaF), which is based on RAG, to evaluate the factual accuracy of the text generated by LLMs. Vykopal et al. (2024) present a comprehensive review of claim verification frameworks that use LLMs, focusing on methods such as RAG and fine-tuning. Our work is different from previous research, as we implement a RAG pipeline that enhances LLMs' fact-checking capability, using Automatic Scraping, integrating both foundational and Advanced RAG components. We use Really Simple Syndication (RSS) (Wikipedia contributors, 2024) feeds from reputable Indian news sources, chosen for their longstanding credibility and wide readership, to access up-to-date information to assess new claims, as LLMs' have knowledge cutoff dates and may contain outdated information.

3 Preethi Dataset

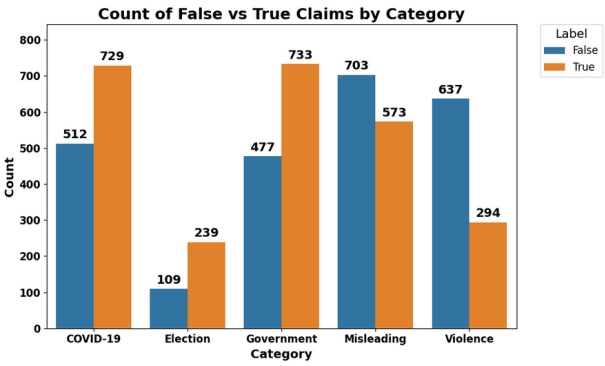


Figure 1: Statistical information of the Preethi dataset about true and false claims across five categories

In this research work, we have created the Preethi dataset, which is based on the publicly available IFND (Sharma and Garg, 2021a). A claim, as defined by Panchendrarajan and Zubiaga (2024), is a statement that can be verified against evidence. The IFND has several inconsistencies such as incomplete claims, non-claims, questions, and entries with multiple claims; these inconsistencies compromise its overall quality as these claims cannot be verified against evidence, see Table 1 for examples of inconsistent claims. We chose IFND because it is publicly available and its inconsistencies

highlight the need for a refined and higher-quality resource, an opportunity we address through the creation of Preethi dataset. We have manually annotated a dataset of 5,006 claims in English with five topics from IFND, namely *Covid-19*, *Election*, *Government*, *Misleading*, and *Violence*. Statistical details of the Preethi dataset are presented in Figure 1. The Preethi dataset is not a strict subset of IFND. Of the 2,568 true claims, 2,500 are sourced from IFND. Among the 2,438 false claims, 2,435 are collected from fact-checking websites. Following, Egelhofer and Lecheler (2019) we treat partially true claims from fact-checking sources as false, given their potential to spread misinformation similar to fully false claims. To reconstruct complete claims from inconsistent IFND entries, we use their original sources, identified via Google Web Search (Google, 2024a) and, when necessary, Microsoft Copilot (Microsoft, 2024).

Claim	Inconsistency
This Video Is Not Of UP Police Chasing A.....	Incomplete
Did Israel bomb Iranian nuclear facilities?	Question
Drop, Don't Extend It	Non-claim
India's Ministry of Culture has NOT announced a relief....Fact Check: Chill. Iceland hasn't declared religions as weapons of mass destruction	Multiple claims

Table 1: Inconsistent claims in IFND

Inspired by the AVeriTeC, we provide additional metadata for each claim, including supporting documents from the Web, the date of the claim, gold justifications, and gold QA pairs. Gold justifications and gold QA pairs are created manually based on the information in the supporting documents. To maintain the quality of the dataset, we have involved three annotators who were trained via detailed guidelines. We achieve a Cohen's Kappa agreement score of 80% for claim veracity labels and 75% for boolean QA pairs, indicating substantial inter-annotator agreement. In addition, all abstractive and extractive QA pairs are manually checked by annotators for correctness and relevance by verifying them against supporting documents. To make our dataset available in Telugu, we translate the English dataset using the Google Translate API (Google, 2024b). To assess the quality of the translated data, we perform a back-translation from Telugu to English and compare it with the original English version. This results in a BLEU (Papineni et al., 2002) score of 0.255 and a METEOR (Banerjee and Lavie, 2005) score of 0.659, indicating moderate consistency between the original and back-translated texts. However, the raw machine translations are not directly used in our experiments. Instead, three native Telugu speakers have manually post-edited the machine translated output and removed the syntactic and semantic errors. The final Telugu dataset is used for experiments, ensuring high-quality translations and minimizing the potential bias introduced by machine translation errors. We calculate post-edits by comparing the initial machine-translated Telugu dataset with the final manually annotated Telugu dataset using Pyter (pyter developers, 2024) to measure the translation error rate (TER) (Snover et al., 2006). A total of 31,465 post-edits are made. Table 2 compares Preethi dataset to the existing benchmark datasets.

Dataset	Justifications	Supports Telugu	QA Pairs
X-CLAIM	✗	✓	✗
AVeriTeC	✓	✗	✓
DFND	✗	✓	✗
IFND	✗	✗	✗
X-Fact	✗	✗	✗
Fact Drill	✗	✓	✗
Preethi (ours)	✓	✓	✓

Table 2: Comparison of Preethi dataset with benchmark datasets.

3.1 QA Pairs

Each claim in our dataset has three manually created QA pair types; see Table 3 for examples.

Boolean: Our dataset contains 4,010 indirect and 996 direct boolean QA pairs. Direct QA pairs rephrase the claim itself as a yes/no question, while indirect QA pairs pose a related yes/no question that helps verify the validity of the claim.

Abstractive: QA pairs are created by summarizing the relevant information about the claim.

Extractive: QA pairs, in which the answer is a direct snippet or a phrase taken word-for-word.

Claim	*The Eiffel Tower is in London*
QA Type	**Question(Q) & Answer(A)**
Direct Boolean	Q: Is the Eiffel Tower in London? A: No
Indirect Boolean	Q: Is the Eiffel Tower in France? A: Yes
Abstractive	Q: What is the Eiffel Tower? A: a well known monument....
Extractive	Q: Where is the Eiffel Tower? A: Paris, France.

Table 3: Boolean, Abstractive and Extractive QA pairs

4 Methodology and Experiments

This section discusses different approaches that are used in our experiments: *1) Simple Prompting* and *RAG* approaches that include *2) Naive RAG*; *3) Advanced RAG* and *4) Automatic Scraping*. In our

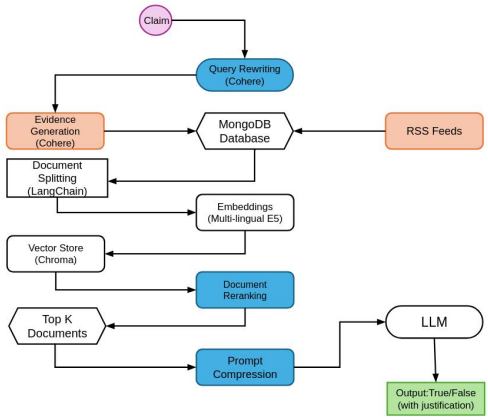


Figure 2: RAG Approaches

experiments, we use gold justifications to evaluate the justifications generated by LLMs and gold QA pairs to assess the quality of QA pairs generated by LLMs. For claim veracity evaluation, we use F1 score. In order to evaluate the justifications generated by LLMs, we use METEOR, ROUGE-L (R-L) (Lin, 2004), ChrF (Popović, 2015), BERTScore (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020). We make our complete code [2] and additional details public.

4.1 Simple Prompting

In Simple Prompting, we use a zero-shot approach (Wei et al., 2022), where the LLM relies solely on its pre-trained knowledge and general language understanding to classify only claims, operating without any additional supporting documents. In this approach the LLM is given a claim as input, and it is tasked to classify a claim as "false" or "true" and provide reasoning or justification for its decision. Without such explanations, the classification may appear arbitrary or unsupported. For the experiments, we consider the Simple Prompting approach as a baseline.

4.2 RAG Approaches

Since LLMs are not updated regularly and have a fixed knowledge cut-off date, they may hallucinate. To address this, we use RAG. In order to find supporting documents for claims, we use the Cohere c4ai-command-r7b-12-2024 (Cohere For AI, 2024) model for English and Telugu. To handle the large number of new claims that appear every day, we use RSS feeds. These feeds are updated regularly by different on-line news sources, providing up-to-date information. To manage this data, we choose the MongoDB (MongoDB Inc., 2024) database for our experiments. It is a NoSQL database suitable for unstructured data, making it ideal for storing RSS feeds and supporting documents retrieved by Cohere. We collect RSS feeds from reliable Indian news sources such as NDTV (NDTV, 2024) for English and Eenadu (Eenadu, 2024) for Telugu. Chroma (Chroma, 2024), a vector database, is used to store documents using vector representations.

4.2.1 Naive RAG

In the Naive RAG approach, as shown in Figure 2 (excluding the steps highlighted in blue), the process unfolds as follows:

Step 1: Cohere c4ai-command-r7b-12-2024 model is prompted to provide supporting documents for a given claim. These documents are then stored in MongoDB.

Step 2: The MongoDB retriever, which uses string matching, identifies, and retrieves documents relevant to the claim. The retrieved documents are processed through the LangChain text splitter (LangChain, Inc., 2025), which divides documents into smaller segments.

Step 3: These segments are converted to vector embeddings using multilingual E5 Text embeddings (Wang et al., 2024). These embeddings are then stored in Chroma.

Step 5: Cosine similarity is used to compare the embeddings of the claim with the documents stored in Chroma. The top three documents with the highest cosine similarity scores are retrieved from Chroma and used as evidence for the LLM.

Step 6: Finally, the LLM analyzes the evidence in the context of the claim and classifies the claim as true or false, along with justifications for its decision.

4.2.2 Advanced RAG

Advanced RAG is similar to Naive RAG but with additional components such as query re-writing, document re-ranking, and prompt compression. In Figure 2 the additional components of Advanced RAG are highlighted in blue.

Query Re-writing: For query re-writing, we use the Cohere c4ai-command-r7b-12-2024 model, which modifies the original claim to improve its quality for better retrieval of documents. This includes correcting spelling errors, rephrasing, or adding additional context to a claim for better understanding. See Table 4 for examples. According

[2] https://github.com/formallinguist/Automatic-Fact-Checking

to Skitalinskaya and Wachsmuth (2023), the criteria for re-writing a claim include maintaining syntactic and semantic coherence, being grammatically correct, and removing ambiguity. A good claim is precise, includes relevant context, and is not ambiguous. In our experiments, we observe that 50 % - 60 % of claims undergo this process. We calculate this using string matching. We have manually verified 50 claims in English and Telugu to check the quality of the re-written claims. We observe that re-written claims in English are syntactically and semantically coherent, while Telugu re-written claims have grammatical errors.

Document Re-ranking: For document re-ranking, we use bert-multilingual-passage-reranking-msmarco (ambeRoad, 2022). It calculates the relevance of each document with respect to the claim and then sorts the documents by the scores to determine the best matches. This ensures the documents that are most relevant for the claim are ranked higher for further processing. Unlike cosine similarity in the Naive RAG approach, which only compares vector proximity, here, the CrossEncoder evaluates the relationship between the claim and the document in context. The top three re-ranked documents are considered for further processing.

Prompt Compression: For prompt compression, we use the Cohere c4ai-command-r7b-12-2024 model. This involves reducing the length of a prompt while retaining its most important information. This helps in scenarios where there is a limited context window for an LLM.

4.2.3 Automatic Scraping

In Automatic Scraping, we extract content from URL in the supporting documents of the Preethi dataset using BeautifulSoup (BS4) (Richardson). To overcome the limitation of the context window of the LLMs, we use a sentence-transformers/paraphrase-multilingual-mpnet-base-v2 (Reimers and Gurevych, 2019). This model identifies the most relevant sentences from the supporting documents by comparing their semantic similarity to the given claim. We retrieve up to 3,000 characters of content that are most relevant to the claim. This selected content is then used as the context for the LLM and is referred to as the *refined context*. This approach is repeatable with new data if the claim and its URL are available.

4.3 Evaluation of QA pairs

In claim verification, asking good questions is crucial (Schlichtkrull et al., 2023). To assess the quality of QA pairs generated by LLMs, we calculate their similarity to gold-standard QA pairs. We use an in-context learning approach (Dong et al., 2024), where the gold QA pair serve as reference to guide the LLM in generating boolean, abstractive, and extractive QA pairs in the desired format for a claim. For evaluation, we follow the approach of Schlichtkrull et al. (2023), we first compute METEOR scores and then apply the Hungarian algorithm (Kuhn, 1955) to identify the optimal one-to-one matching between LLM-generated and gold-standard QA pairs by maximizing METEOR scores. Table 7 provides English and Telugu scores.

4.4 Experiments

We use the models listed in Table 5 for experiments.

Versions of models	Parameters
Gemma-2 (Team, 2024)	9B
Llama-3 (AI@Meta, 2024)	70B
Llama-3.3 (Meta, 2024)	70B
Llama-3 (Meta, 2024)	8B
Mixtral (Jiang et al., 2024)	8x7B

Table 5: Models for experiments

We select models that are trained on publicly available online data. All LLMs are instructed in English, and we experiment with three different prompt templates, selecting the best-performing one for our experiments. To ensure consistency of the results, each experiment is conducted three times, with same temperature. We calculate variance across the three runs for both English and Telugu using the F1 scores of the best-performing models. For English, the Naive RAG exhibits the highest variance, while the Advanced RAG shows the lowest. For Telugu, Automatic Scraping results in the highest variance, whereas the Naive RAG has the lowest. Table 6 shows average scores of model performance across English and Telugu datasets. We use multiple evaluation metrics in our experiments to gain a comprehensive understanding of the models' performance, as no single metric can fully capture the quality of a model's output for justification generation.

No.	Original Claim	Re-written Claim
1	Jharkhand new hotspot of illicit opium cultivation: NCB	The NCB reports significant opium cultivation in Jharkhand, identifying it as a potential hotspot.
2	Govt confident of privatising Air India, BPCL by first half of 2021-22 divestment secretary	The Indian Government's Divestment Strategy: Privatization of Air India and BPCL by 2022 and the Secretary's Statement on Future Plans.

Table 4: Comparison of original and re-written claims

Model	Approach	F1 (Claim Verif.)		English Justification Generation Scores						Telugu Justification Generation Scores					
		En	Te	METEOR	R-L	ChrF	BLEURT	BERTScore	Avg-En	METEOR	R-L	ChrF	BLEURT	BERTScore	Avg-Te
Llama-3-70B	SP	80.16	42.95	0.288	0.283	39.92	0.48	0.87	0.464	0.126	0.165	25.34	0.45	0.72	0.343
	N-RAG	58.16	40.77	0.267	0.275	38.51	0.47	0.86	0.451	0.140	0.163	23.94	0.44	0.71	0.339
	A-RAG	61.21	44.31	0.256	0.259	**41.11**	**0.56**	0.88	0.473	0.134	0.174	26.08	0.48	0.72	0.354
	AS	**86.14**	80.45	0.281	0.289	37.72	0.47	0.89	0.461	0.123	0.162	24.70	0.45	0.72	0.340
Llama-3.3-70B	SP	75.07	70.68	0.275	0.275	38.55	0.47	0.89	0.459	**0.172**	**0.229**	**32.79**	**0.51**	0.71	**0.390**
	N-RAG	57.44	38.86	0.286	0.282	35.41	0.43	0.87	0.444	0.106	0.123	27.04	0.40	0.72	0.324
	A-RAG	59.38	41.76	0.259	0.250	37.81	0.42	0.88	0.437	0.135	0.174	28.29	0.43	0.72	0.348
	AS	77.22	**80.58**	0.308	0.318	39.81	0.49	**0.90**	0.482	0.163	0.196	31.84	0.50	**0.73**	0.381
Llama-3-8B	SP	56.21	48.45	0.294	0.279	39.85	0.50	0.89	0.472	0.138	0.194	29.64	0.42	**0.73**	0.356
	N-RAG	52.41	47.29	0.266	0.280	37.59	0.45	0.86	0.446	0.139	0.184	28.21	0.41	0.72	0.347
	A-RAG	60.11	49.75	0.254	0.304	38.41	0.49	0.89	0.464	0.133	0.203	29.61	0.44	0.72	0.358
	AS	70.83	50.77	0.288	0.291	38.64	0.47	0.87	0.461	0.124	0.164	25.96	0.42	0.72	0.338
Mixtral-8x7B	SP	56.95	49.22	0.285	0.273	38.94	0.49	0.88	0.463	0.110	0.129	27.59	0.29	0.72	0.305
	N-RAG	57.19	51.24	0.293	0.303	37.51	0.47	0.86	0.460	0.153	0.172	28.39	0.41	**0.73**	0.350
	A-RAG	59.26	55.49	0.280	0.293	38.66	0.51	0.89	0.472	0.146	0.213	28.66	0.43	0.72	0.359
	AS	84.08	73.86	**0.316**	**0.340**	41.03	0.48	0.87	**0.483**	0.087	0.114	23.27	0.28	0.70	0.283
Gemma-2-9B	SP	64.72	57.41	0.208	0.283	31.89	0.46	0.87	0.428	0.125	0.183	26.66	0.43	**0.73**	0.347
	N-RAG	62.21	52.39	0.197	0.264	30.51	0.45	0.87	0.417	0.103	0.173	28.41	0.43	0.72	0.342
	A-RAG	63.81	50.77	0.180	0.283	34.74	0.49	**0.90**	0.440	0.094	0.213	30.49	0.46	0.72	0.358
	AS	83.23	78.05	0.217	0.277	36.77	0.48	0.87	0.442	0.114	0.152	24.68	0.42	0.72	0.331

Table 6: Scores across different metrics for English (En) and Telugu (Te). Approaches include Simple Prompting (SP), Naive RAG (N-RAG), Advanced RAG (A-RAG), and Automatic Scraping (AS). The best results for each metric and language are highlighted in **bold**, while the best scores per metric and language for each model are underlined. ChrF scores are normalized (divided by 100) when computing average scores for English and Telugu.

Model	En	Te
Llama-3-70B	0.101	0.072
Llama-3.3-70B	0.140	0.090
Llama-3-8B	0.178	**0.124**
Mixtral-8x7B	**0.208**	0.089
Gemma-2-9B	0.126	0.079

Table 7: QA pairs Hungarian METEOR scores for English (En) and Telugu (Te). Best scores are highlighted in **bold**

5 Results and Discussion

We analyze claim verification and justification generation results for English and Telugu to answer RQ1 and RQ2, and also analyze QA pair results.

5.1 Claim Verification

In order to answer RQ1, we examine the claim verification results presented in Table 6.

5.1.1 English

Simple Prompting: Within the Simple Prompting approach across models, Llama-3-70B achieves the highest F1 score, likely due to its large size and English-focused training, enabling strong reasoning without external supporting documents. In contrast, Llama-3-8B performs the worst, likely due to its smaller size. Interestingly, Llama-3-70B outperforms both Naive and Advanced RAG under Simple Prompting, showing the largest performance gap of 23.95 points of F1 score between the best and worst performing models.

Automatic Scraping: We observe that all models obtain their highest F1 scores with this approach. With Automatic Scraping Llama-3-70B has the highest F1 score and Llama-3-8B has the lowest F1 score. This suggests that Automatic Scraping provides high-quality, relevant context that helps LLMs verifying and classifying the claims. All models perform better with Automatic Scraping compared to Simple Prompting.

Naive RAG: Gemma-2-9B achieves the highest F1 score and Llama-3-8B has the lowest F1 score. We observe that the Naive RAG approach does not improve the models' performance with respect to Simple Prompting except for Mixtral-8x7B. One possible reason for the relatively low F1 scores

across models is that the Cohere model may not retrieve suitable supporting documents, particularly for claims related to the Indian context. This limitation at the evidence retrieval stage can significantly impact the quality of context available to the LLM, thus reducing overall performance.

Advanced RAG: Gemma-2-9B achieves the highest F1 score and Mixtral-8x7B shows the lowest F1 score. We observe that models consistently perform slightly better with Advanced RAG compared to Naive RAG. This improvement may be attributed to the additional components in Advanced RAG that enhance the models' overall performance. However, results with the Simple Prompting approach remain superior except for Llama-3-8B and Mixtral-8x7B. Notably, this approach results in the smallest performance gap of 4.55 points in average F1 score between the best and worst performing models.

5.1.2 Telugu

Simple Prompting: Within the Simple Prompting approach, Llama-3.3-70B obtains the highest F1 score, likely due to some knowledge of Telugu in its pre-training data, as it was trained on open-source web documents. In contrast, all other models have low F1 scores. This could be due to the limited presence of Telugu in their pre-training corpora.

Naive RAG: Gemma-2-9B has the highest F1 score and Llama-3.3-70B has the lowest F1 score. The relatively low scores across models may be attributed to the Cohere model's limited ability to retrieve relevant supporting documents for claims in Telugu. Since Telugu is a low-resource language, the amount and quality of content available in it would be significantly lower compared to English. In this approach, only Mixtral-8x7B performs better than the models with Simple Prompting. This approach has the lowest performance gap of 14.03 F1 points between the best and the worst performing models.

Advanced RAG: Mixtral-8x7B has the highest F1 score and Llama-3.3-70B has the lowest F1 score. The F1 scores across models suggest that the Advanced RAG generally performs slightly better than the Naive RAG for Telugu, with the exception of Gemma-2-9B. This exception may be due to Gemma-2-9B not having received suitable documents as context. The modest improvements seen with Advanced RAG can likely be attributed to its additional components. However, F1 scores for Telugu remain relatively low compared to those for English. Among the evaluated models, Llama-3-70B, Llama-3-8B, and Mixtral-8x7B outperform Simple Prompting.

Automatic Scraping: Under this method, in which the context is in English, Llama-3.3-70B achieves the highest F1 score, demonstrating its ability to transfer knowledge from English to Telugu. In comparison, the smaller Llama-3-8B has the lowest F1 score. These results highlight that LLMs perform significantly better in Telugu when provided with suitable supporting documents. Here, all models perform better than Simple Prompting. The performance gap between the best and the lowest performing model is 29.81 average F1 score, which is highest using this technique.

Automatic scraping has the highest scores for claim verification as it uses reliable supporting documents as context. To answer RQ1, our experiments show that LLMs perform better at claim verification in English compared to Telugu.

5.2 Justification Generation

As shown in Table 6, we compare the results of justification generation score (JGS) for Telugu and English to answer RQ2. JGS is an average of METEOR, R-L, ChrF, BLUERT and BERTScore. We observe that for English and Telugu different models and approaches have high scores across different metrics. However, for English, Mixtral-8x7B with Automatic Scraping has the highest overall average JGS. The best overall JGS in Telugu is attained by Llama-3.3-70B using Simple Prompting. Manual review of 100 justifications from various methods reveals no clear link between claim verification and JGS.

5.3 QA pairs

As shown in Table 7, Mixtral-8x7B achieves the highest METEOR score for English, likely because it is trained on predominantly English data. In contrast, Llama-3-8B, despite being a small model, achieves the best METEOR score for Telugu. This performance may result from its closer adherence to the reference QA pairs, whereas larger models tend to "hallucinate" or be creative (Lin et al., 2022), which negatively affects similarity scores .

6 Error Analysis

In this section, we present the qualitative and quantitative error analysis for English and Telugu.

6.1 Qualitative Error Analysis

We manually analyze 100 samples from the best performing models for each task: Llama-3-70B (English) and Llama-3.3-70B (Telugu) for claim verification; Mixtral-8x7B (English) and Llama-3.3-70B (Telugu) for justification generation.

6.1.1 Claim Verification

Example 1 Bias Error
Model output : The indian air force conducted an air strike on a jaish-e-mohammed training camp in balakot, pakistan, on february 26, 2019, *reportedly killing several terrorists.*

Example 2 Hallucination Error
Model output : covishield, will be priced at *around ₹200-₹300 per dose, not ₹1,000.*

Example 3 Retrieval Error
Claim: Congress mla calls kumaraswamy's absence at tipu jayanti celebrations an insult to muslims.
Model output: the provided context discusses ramdas athawale's criticism of raj thackeray, not yogesh sagar's protest against road closures for friday prayers.

Example 4 Translation error
Claim in Telugu: రక్షణ మంత్రి రాజ్‌నాథ్ సింగ్ *BRO* నిర్మించిన 44 వ్యూహాత్మక వంతెనలను ప్రారంభించారు, అందులో 7 లడఖ్‌లో ఉన్నాయి.
Claim in English: Defense Minister Rajnath Singh inaugurated 44 strategic bridges built by *BRO*, out of which 7 are in Ladakh.
Model output Translation from Telugu: It is *not Rajnath singh's brother,* it is defense minister Rajnath singh who inaugurated these 44 strategic bridges.

Figure 3: Different types of Errors

We have focused on identification of biases (Dev et al., 2022), hallucinations (Li et al., 2024), retrieval, and translation errors. **Biases** are unfair patterns in responses that occur when the model favors certain views, stereotypes, or groups over others. As shown in Example one in Figure 3, there is a potential bias toward labeling individuals as terrorists. **Hallucinations** occur when LLMs generate information that is factually incorrect. In Example two in Figure 3 the language model hallucinates about the pricing of the Covishield vaccine. **Retrieval errors** in RAG approaches refer to the failing of the model to obtain relevant or sufficient contextual knowledge to support accurate reasoning, leading to incorrect or unsupported output. Example three in Figure 3 shows that the retrieved documents are not related to the claim about *kumaraswamy* and *tipu jayanti*. Finally, **translation errors** are uniquely observed when there is a language mismatch in the claim or between claim and context - for example, when there are acronyms in English and the claim is in Telugu. In such scenarios, the models attempt to translate the English acronyms to Telugu as in Example four in Figure 3 where it can be observed that BRO acronym which is in English is translated to "brother" in Telugu.

Approach	B	H	R	O
SP	13.14%	4.81%	–	1.92%
AS	4.91%	1.02%	3.85%	4.08%
N-RAG	12.12%	5.39%	13.54%	10.76%
A-RAG	11.98%	1.22%	16.75%	9.27%

Table 8: English errors with percentage (relative to 5006 claims). B: Biases, H: Hallucinations, R: Retrieval, O: Other.

Approach	B	H	R	T	O
SP	5.17%	10.71%	–	–	13.16%
AS	1.00%	2.46%	–	0.26%	12.86%
N-RAG	8.79%	9.35%	1.62%	8.63%	20.59%
A-RAG	0.50%	8.25%	10.53%	–	13.18%

Table 9: Telugu errors. B: Biases, H: Hallucinations, R: Retrieval, T: Translation, O: Other.

6.2 Justification Generation

We manually evaluate generated 100 justifications against the gold-standard justifications from different approaches. We observe that Automatic Scraping enables LLMs to generate good-quality justifications in English and Telugu. Manual inspection further reveals that the quality of text generation is generally good for English across different models and approaches. However, outputs in Telugu often exhibit syntactic and semantic errors, along with instances of Tenglish (a mix of Telugu and English) script.

6.3 Quantitative Error Analysis

We use the mistral-saba-24B LLM (AI, 2025) as a judge, following an in-context learning approach.We manually select one misclassified claim with its justification from each error type as a demonstration for the judge. Misclassified claims and their justifications are filtered and they are then classified by the LLM into the predefined error categories, with uncategorized errors labeled as "Other." Tables 8 and 9 report category-wise error percentages. Manual verification of 50 errors per language confirms accurate quantification.

7 Conclusion

In this project, we introduced a new English-Telugu claim verification dataset with manually annotated QA pairs and justifications. We used it to benchmark Simple Prompting and RAG approaches with LLMs. Our results show that the models perform better in English than in Telugu, highlighting challenges in claim verification and justification generation in Telugu.

Limitations

The results of our experiments are based on a dataset of 5,006 claims with only two labels from five topics. Performance may vary with larger and more diverse datasets. In India, claims occur in multiple languages, but for this study, we work in one language at a time. We need to explore different prompt templates for Telugu and English, as some templates perform better than others. Our dataset consists only of textual claims, excluding images and videos, which are also commonly associated with the spread of false claims. Although we have relied on lexical and semantic similarity metrics, we have not incorporated additional text generation metrics to detect hallucinations. Our evaluation relies exclusively on automatic metrics such as R-L, METEOR, and BERTScore. While these provide surface-level and semantic overlap, they may not adequately capture the true quality of either QA pairs or justifications. In particular, justifications can often be expressed in many valid ways that differ substantially from the reference, leading to artificially low metric scores, while conversely, outputs that are lexically or semantically similar to the reference may still be incorrect. The limited variance in our reported BERTScore values (0.70–0.73) for Telugu further suggests that these metrics may not be sensitive enough to meaningful differences in justification quality. A more robust assessment would require human evaluation, which could better judge correctness, faithfulness, and usefulness of both the questions/answers and the justifications. Future work should therefore complement automatic metrics with systematic human evaluation. Naive RAG and Advanced RAG approaches that we use for experiments often require significant processing time, particularly for languages like Telugu. This is due to the complexity of tokenization, retrieval, and generation stages, which may not be as optimized for low-resource languages as they are for English. We have used RSS feeds from only a small number of sources and we have not performed ablation studies on the individual components of Advanced RAG. Since our dataset is derived through translation from English, it may not fully represent native Telugu. Translations tend to exhibit different levels of formality, topic distribution, and cultural biases compared to texts in Telugu produced by native speakers. Therefore, while our dataset serves as a useful resource, we acknowledge that future work should prioritize collecting and incorporating more native-authored Telugu data.

Acknowledgements

We thank Begari Kaveri, Sujatha Theetla and Ravi Teja Chikkala for reviewing and editing the Telugu translations and inter-annotation agreement of the Preethi dataset.

This project was supported by the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the project TRAILS (01IW24005), by *DisAI - Improving scientific excellence and creativity in combating disinformation with artificial intelligence and language technologies*, a project funded by Horizon Europe under GA No.101079164, by the *European Union NextGenerationEU* through the Recovery and Resilience Plan for Slovakia under the project No. 09I01-03-V04-00007, and by Saarland University and University of Basque country in collaboration with Erasmus Mundus Language and Communication Technologies (EMLCT).

References

Mistral AI. 2025. Mistral small 3: A 24 billion parameter language model.

AI@Meta. 2024. Llama 3 model card.

ambeRoad. 2022. bert-multilingual-passage-reranking-msmarco. https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco. Accessed: 2025-01-12.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. Dated data: Tracing knowledge cutoffs in large language models. In *Proceedings of the Conference on Language Modeling (COLM)*.

Chroma. 2024. ChromaDB – The AI-native open-source vector database. https://www.trychroma.com/. Accessed: 2025-01-12.

Cohere For AI. 2024. Cohere c4ai-command-r7b-12-2024. https://huggingface.co/CohereForAI/c4ai-command-r7b-12-2024. Accessed: 2025-01-12.

Sunipa Dev, Emily Sheng, Jieyu Zhao, Aubrie Amstutz, Jiao Sun, Yu Hou, Mattie Sanseverino, Jiin Kim, Akihiro Nishi, Nanyun Peng, and Kai-Wei Chang. 2022. On measures of biases and harms in NLP. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 246–267, Online only. Association for Computational Linguistics.

Alphaeus Eric Dmonte, Roland Oruche, Marcos Zampieri, Prasad Calyam, and Isabelle Augenstein. 2024. Claim verification in the age of large language models: A survey. *ArXiv*, abs/2408.14317.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Eenadu. 2024. Eenadu – Telugu News Portal. https://www.eenadu.net/. Accessed: 2025-01-12.

Jana Laura Egelhofer and Sophie Lecheler. 2019. Fake news as a two-dimensional phenomenon: A framework and research agenda. *Annals of the International Communication Association*, 43(2):97–116.

Google. 2024a. Google Search. https://www.google.com/. Accessed: 2024-11-12.

Google. 2024b. Google Translate. https://translate.google.com/?sl=ta&tl=en&op=translate. Accessed: 2024-11-12.

Ashim Gupta and Vivek Srikumar. 2021. X-fact: A new benchmark dataset for multilingual fact checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 675–682, Online. Association for Computational Linguistics.

Kai Hui, Honglei Zhuang, Tao Chen, Zhen Qin, Jing Lu, Dara Bahri, Ji Ma, Jai Gupta, Cicero Nogueira dos Santos, Yi Tay, and Donald Metzler. 2022. ED2LM: Encoder-decoder to language model for faster document re-ranking inference. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3747–3758, Dublin, Ireland. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L'elio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *ArXiv*, abs/2401.04088.

Vasileios Katranidis and Gabor Barany. 2024. Faaf: Facts as a function for the evaluation of generated text.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

LangChain, Inc. 2025. langchain_text_splitters.base.TextSplitter (API Reference). https://python.langchain.com/api_reference/text_splitters/base/langchain_text_splitters.base.TextSplitter.html. Accessed: 2025-01-12.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7182–7195, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Luyang Lin, Lingzhi Wang, Jinsong Guo, and Kam-Fai Wong. 2025. Investigating bias in LLM-based bias detection: Disparities between LLMs and human perception. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10634–10649, Abu Dhabi, UAE. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of*

the 2023 Conference on Empirical Methods in Natural Language Processing, pages 5303–5315, Singapore. Association for Computational Linguistics.

K. Mallareddy. 2012. Evolution of telugu language teaching and challenges to present curricular trends. *IOSR Journal of Humanities and Social Science*, 5:33–36.

Meta. 2024. Llama-3.3-70b-instruct.

Microsoft. 2024. Microsoft Copilot. https://copilot.microsoft.com/. Accessed: 2024-11-12.

Shubham Mittal, Megha Sundriyal, and Preslav Nakov. 2023. Lost in translation, found in spans: Identifying claims in multilingual social media. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3887–3902, Singapore. Association for Computational Linguistics.

MongoDB Inc. 2024. MongoDB. https://www.mongodb.com. Accessed: 2025-1-12.

NDTV. 2024. NDTV - Latest News and Updates. https://www.ndtv.com/. Accessed: 2025-01-12.

Rrubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal*, 7:100066.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online. Association for Computational Linguistics.

pyter developers. 2024. pyter: Python Text Segmentation. https://pypi.org/project/pyter/. Accessed: 2025-06-12.

Dorian Quelle, Calvin Yixiang Cheng, Alexandre Bovet, and Scott A. Hale. 2025. Lost in translation: using global fact-checks to measure multilingual misinformation prevalence, spread, and evolution. *EPJ Data Science*, 14(1):22.

Eduri Raja, Badal Soni, and Samir Kumar Borgohain. 2023. Fake news detection in dravidian languages using transfer learning with adaptive finetuning. *Eng. Appl. Artif. Intell.*, 126:106877.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Leonard Richardson. Beautiful soup 4 documentation. https://beautiful-soup-4.readthedocs.io/en/latest/. Accessed: January 15, 2025.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: a dataset for real-world claim verification with evidence from the web. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

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.

Sharma and Garg. 2021a. Ifnd dataset. Accessed: 2024-11-07.

Dilip Sharma and Sonal Garg. 2021b. Ifnd: a benchmark dataset for fake news detection. *Complex Intelligent Systems*, 9.

Ronit Singal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024. Evidence-backed fact checking using RAG and few-shot in-context learning with LLMs. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 91–98, Miami, Florida, USA. Association for Computational Linguistics.

Shivangi Singhal, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. 2022. Factdrill: A data repository of fact-checked social media content to study fake news incidents in india. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):1322–1331.

Gabriella Skitalinskaya and Henning Wachsmuth. 2023. To revise or not to revise: Learning to detect improvable claims for argumentative writing support. In

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15799–15816, Toronto, Canada. 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.

Gemma Team. 2024. Gemma.

Ivan Vykopal, Matú Pikuliak, Simon Ostermann, and Marián Simko. 2024. Generative large language models in automated fact-checking: A survey. *ArXiv*, abs/2407.02351.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *ArXiv*, abs/2402.05672.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Wikipedia contributors. 2024. Rss – wikipedia, the free encyclopedia. Accessed: 2025-03-07.

Zhenrui Yue, Huimin Zeng, Lanyu Shang, Yifan Liu, Yang Zhang, and Dong Wang. 2024. Retrieval augmented fact verification by synthesizing contrastive arguments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10331–10343, Bangkok, Thailand. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xuan Zhang and Wei Gao. 2023. Towards LLM-based fact verification on news claims with a hierarchical step-by-step prompting method. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1011, Nusa Dua, Bali. Association for Computational Linguistics.

Synthetic Voice Data for Automatic Speech Recognition in African Languages

Brian DeRenzi
Dimagi
bderenzi@dimagi.com

Anna Dixon
Dimagi*
anna@datafoss.com

Mohamed Aymane Farhi
CLEAR Global
aymenfarhi.25@gmail.com

Christian Resch†
CLEAR Global
christian.resch@clearglobal.org

Abstract

Speech technology remains out of reach for most of the 2,300+ languages in Africa. We present the first systematic assessment of large-scale synthetic voice corpora for African ASR. We apply a three-step process: LLM-driven text creation, TTS voice synthesis, and ASR fine-tuning. Eight out of ten languages for which we create synthetic text achieved readability scores above 5 out of 7. We evaluated ASR improvement for three (Hausa, Dholuo, Chichewa) and created more than 2,500 hours of synthetic voice data at below 1% of the cost of real data. W2v-BERT 2.0 speech encoder fine-tuned on 250h real and 250h synthetic data in Hausa matched a 500h real-data-only baseline, while 579h real and 450h to 993h synthetic data created the best performance. We also present gender-disaggregated ASR performance evaluation. For very low-resource languages, gains varied: Chichewa WER improved by ∼6.5% with a 1:2 real-to-synthetic ratio; a 1:1 ratio for Dholuo showed similar improvements on some evaluation data, but not on others. Investigating intercoder reliability, ASR errors and evaluation datasets revealed the need for more robust reviewer protocols and more accurate evaluation data. All data and models are publicly released to invite further work to improve synthetic data for African languages.

1 Introduction

Africa is home to over 2,300 languages, the vast majority of which have neither functional automatic speech recognition to transcribe speech nor speech synthesis to generate it (Orife et al., 2020). Yet speech technology holds great promise in providing a more inclusive digital experience, especially for vulnerable groups.

Conventional approaches to creating speech technology rely on human data collection in as-yet unsupported languages, incurring substantial costs estimated at more than US$100–150 per hour, even in the best case[1]. As most speech recognition models need several hundred hours of training data to achieve performance sufficient for practical application, with current investments in AI for development, human data collection is prohibitively costly to cover the many languages that remain unsupported. We support further investment in African language technology, but even if it were to become available, those investments have opportunity costs, diverting funds that could otherwise be spent on other interventions.

Therefore, we are investigating synthetic voice data as a complementary approach to create and improve automatic speech recognition (ASR) in African languages. The principle hypothesis motivating our work is that we can leverage Large Language Models (LLMs) and Text-to-Speech (TTS) models to create synthetic voice data of sufficient quality to improve automatic speech recognition models. Our work shows that this synthetic voice data can be created for less than 1% of the cost of collecting real human data[2], while holding potential to complement this human data in creating and improving ASR models for African languages.

2 Synthetic Data: Risks and Rewards

Synthetic data—machine-generated text or speech—has become a well-researched topic in major languages like English in recent years (for an overview, see Liu et al. (2024)). Much of this research is motivated by concerns that all readily

*Current affiliation: Datafoss

†Authors are listed in alphabetical order. Corresponding author: Christian Resch

[1]Based on internal estimates. The authors of NaijaVoices report that the 'true cost' of the whole dataset is more than US$600,000. Assuming an equal split across the three languages in the dataset, this would imply a true cost per hour of Hausa data of US$ ∼345.42, see https://naijavoices.com/membership.

[2]This is excluding fixed costs in both cases, like setting up data collection platforms for real data or TTS model development.

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 152–186, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-016

available data for the development of AI models has been leveraged (Villalobos et al., 2024), and therefore further advancements spurred by data scaling will require new approaches to create data. Other concerns like privacy (Abay et al., 2018) or costs (Gilardi et al., 2023) also prompted research on synthetic data. Additionally, there is a growing body of research into synthetic data in low-resource languages which, save for several notable exceptions, so far only offers limited evidence for African languages.

For automatic speech recognition, Huang et al. (2023) have shown that for English, synthetic data generation using large-scale pre-trained neural networks in combination with TTS models, a process similar to ours, can reduce word error rate (WER) by between 9 and 15% (see also Moslem (2024) and Hu et al. (2021)). Gokay and Yalcin (2019) report reductions of around 15% in WER for Turkish, Yang et al. (2024) between 27.45 and 45.85% for Chinese dialects and Zevallos (2022) 8.73% for Quechua. Joshi et al. (2025) find that adding synthetic data in Hindi improves Bhojpuri ASR performance by 4.7 points WER on average. Wang et al. (2020) also demonstrate that an approximate 50/50 combination of human and synthetic data performed comparably to the same amount of human data alone. In a very successful application, Xu et al. (2020) achieve 17% WER for Lithuanian using only 1.3 hours of labeled and 12 hours of unlabeled data.[3]

As this research shows, there is a body of research on synthetic data for low-resource languages, but African languages have been rarely covered. There are notable exceptions for synthetic text (Abdulmumin et al., 2022; Kreutzer et al., 2022), linguistically informed data augmentation and synthetic data frameworks (Ajuzieogu, 2023) and synthetic text for language and topic classification models (Quinjica and Adelani, 2024; Adelani et al., 2024).

While previous research shows the potential of synthetic data to complement human data in the data-scarce situation that we face in many African languages, the research also highlights limitations and risks. Most limitations stem from the gap between real and synthetic data (Hu et al., 2021), as well as from synthetic data inheriting and potentially amplifying the same biases as the models used to create it (Wyllie et al., 2024; Wang et al., 2025). Certain tone and noise that are typically present in real-world data are often missing from synthetic data (Xue et al., 2022; Hu et al., 2021). Many commonly used LLMs have been shown to exhibit a bias towards Western, industrialized cultural norms and a lack of cultural understanding in other contexts (Rao et al. (2023), Magdy et al. (2025) for Arabic, Pranida et al. (2025) and Putri et al. (2024) for Sundanese and Indonesian). Creating synthetic text will likely aggravate this missing representation, especially for semantically meaningful tasks such as machine translation.

3 Methodology

Our process of creating and evaluating synthetic voice data has three key steps:

1. Generate and evaluate synthetic text using an LLM
2. Generate and evaluate synthetic voice data with a Text-to-Speech (TTS) model based on the synthetic text
3. Fine-tune an automatic speech recognition (ASR) model with different ratios of human and synthetic voice data and evaluate performance differences

We describe our methods and details for those steps separately:

3.1 Step 1: Synthetic Text Generation and Evaluation

We created and evaluated synthetic text for the following 10 African languages: Hausa, Northern Somali, Yoruba, Wolof, Dholuo, Kanuri, Chichewa, Twi, Kinande, and Bambara. Additionally, we included small-scale generations and evaluations for two further languages, Yemba and Ewondo, but with only very poor performance. Our language selection criteria aimed to select languages that represent African language diversity through the representation of different regions, language families, and speaker populations (see Appendix A). Beyond linguistic diversity, we also considered practical factors, such as the capacity of the Translators without Borders (TWB) linguist community

[3] We recommend not to compare WER between languages because of substantial morphological differences, i.e. depending on the language a single word might carry different semantic meaning while it is always counted individually in the WER. Furthermore, evaluation datasets such as FLORES and, by extension, FLEURS have known shortcomings which differ by language, see Abdulmumin et al. (2024) for an investigation for African languages.

for text evaluation and the availability of key publicly available datasets (e.g., open.bible, FLEURS, Common Voice) necessary for subsequent steps.

For the synthetic text generation, the system prompt (see Appendix B) instructs the LLM to generate simple short sentences and questions directly in the target language, as well as to return English translations for further evaluation of language understanding. We incorporated two-shot prompting for contextual guidance. The topic of synthetic text generation can be configured in the prompt. Our experiments sampled equally from 34 distinct themes with 17 themes covering the UN Sustainable Development Goals and 17 themes covering the most common topics covered in the FLORES/FLEURS dataset extracted through language topic modelling. We primarily evaluated LLMs provided by OpenAI and Anthropic, especially GPT-4o, GPT-4.5, o1, Claude 3.5 and Claude 3.7.

Automated evaluation of the synthetic text is not feasible for low-resource languages and we relied on human evaluation. For each language we generated 1,200 randomly shuffled sentences over two rounds for a few different configurations (usually three LLMs).[4] Linguists on the Translators without Borders (TWB) platform were sourced according to their native language and experience delivering linguistic tasks. TWB linguist reviewers rate each sentence on five key metrics intended to capture the quality of the sentence in the target language and understanding: 1) Readability and Naturalness [1–7], 2) Grammatical Correctness (Yes/No), 3) All Words Real (Yes/No), 4) Notable Error in Translation (Yes/No), and 5) Adequacy and Accuracy of Translation [1–7]. Based on the human evaluation, we selected the configuration (LLM) with the highest mean Readability and Naturalness.

For the subset of languages selected for subsequent synthetic voice generation and ASR fine-tuning, we generated a large corpora of between 650,000 and 674,000 sentences with the best performing LLM (see Appendix C for details). The text generation process was identical to that used for text evaluation, except that we utilized a batch processing API for reduced cost.

[4]For most experiments, we opted for a two-round sentence generation and evaluation approach, in which we first compared the readability and naturalness of 600 sentences generated by 3-4 LLMs and subsequently generated another 600 sentences using the best model to analyze the impact of theme. After discovering that our experiments were frequently not yielding significant differences between themes, we opted for more equal sampling among different LLMs.

3.2 Step 2: Synthetic Voice Data Generation and Evaluation

Based on the results of the evaluation of the synthetic text generation in Step 1, we selected three languages for the creation and evaluation of synthetic voice data: Hausa, Dholuo and Chichewa.

As necessary conditions, we required languages where at least one LLM was capable of generating synthetic text of sufficient quality. Due to limitations in time and resources, we also required that at least two of the three languages have available data for fine-tuning or training TTS models, as well as available ASR evaluation data.

Beyond those necessary conditions, our goal was to maximize the variance of the speaker populations, available ASR training data, language families, and geography.

We used the open.bible corpus (Global Bible Initiative, n.d.) of Bible recordings to fine-tune and evaluate different TTS models. We excluded this data from our ASR training data for this reason. The open.bible corpus only contains recordings of Bible recitations by male speakers, and given the training data which we used, our synthetic voice data is also exclusively male. This raises the risk that the resulting ASR models show gendered performance, e.g. that they perform worse for female speakers than for male speakers. To investigate this risk, we also evaluated gender bias in the ASR performance where the evaluation data allows this.

For each language, we fine-tuned both the XTTS-v2 model (Casanova et al., 2024) and VITS or one of its variants, specifically YourTTS (Casanova et al., 2023), using the Coqui TTS framework, and building on the BibleTTS project (Meyer et al., 2022). XTTS-v2 does not support any African languages, but has been fine-tuned for Wolof.[5] For fine-tuning YourTTS, we used the checkpoint trained on the CML-TTS dataset (Oliveira et al., 2023) that supports eight languages. We used the original BibleTTS model for Hausa, but also retrained the model based on a revised processing of the open.bible corpus, a different checkpoint and different hyperparameters ("Modified Bible TTS"). For Hausa and Chichewa, we also evaluated the available MMS TTS models (Pratap et al., 2023)[6].

Transformer-based TTS models like XTTS have

[5]https://huggingface.co/galsenai/xTTS-v2-wolof

[6]The MMS TTS model language coverage is available at https://dl.fbaipublicfiles.com/mms/misc/language_coverage_mms.html

the problem of hallucinating, especially at the end of audio files. To remedy this issue, we re-transcribed the synthetic audio files with an existing ASR model and then calculated the ratio of the length of the transcript to the length of the original synthetic text. We only generate synthetic data with XTTS for Hausa and used the MMS-1B (Pratap et al., 2023) for this analysis. This method does not rely on the accuracy of the ASR model used but assumes a basic performance to ensure that the length of the re-transcription is meaningful.[7] We finally removed outliers of this ratio of less than 0.85 and more than 1.06 which indicate that the TTS model had hallucinated additional words not present in the original synthetic text or omitted words that were. We picked the cut-off points by manually expecting ∼100 samples. This process removed ∼26.9% of the synthetic audio created by XTTS.[8]

After training a total of ten TTS models across all three languages, we evaluated 337 synthetic audio samples per model with the help of two native speakers from the TWB Community per language. As commonly applied, our evaluation included intelligibility and naturalness on five-point scales.

We then selected YourTTS, the best-performing TTS model across all three languages, to create synthetic voice data corpora of 993h for Hausa, 775h for Dholuo and 550h for Chichewa (see Appendix C for details). For Hausa, we also created 450h of synthetic data with XTTS, the only transformer-based model. We make these synthetic data corpora openly available on CLEAR Global's Hugging Face page[9].[10]

To improve the robustness of our models to noisy acoustic environments, we augmented the synthetic data by adding noise. We mixed the clean synthetic data with noise samples drawn from the Room Impulse Response and Noise Database[11]. For each utterance, we randomly sampled the signal-to-noise ratio (SNR) from a normal distribution with mean 50dB and standard deviation 15dB. Similarly, we randomized the audio amplitude using a normal distribution ($\mu = -20\,\mathrm{dB},\ \sigma = 5\,\mathrm{dB}$).

3.3 Step 3: ASR Model Fine-tuning and Evaluation

Given the substantial differences in available ASR training data between the three languages for which we created synthetic voice data, we conducted our ASR evaluation based on two scenarios: a medium data scenario with Hausa as the representative language and a low data scenario with Dholuo and Chichewa as the representative languages.

3.3.1 Medium Data Scenario: Hausa

Through the NaijaVoices project (Emezue et al., 2025), we had over 500 hours of human Hausa voice data available. Only a few other African languages like Igbo and Yoruba (NaijaVoices) or Swahili, Kinyarwanda, Kabyle, and Luganda (Common Voice) have available datasets of comparable size. This led us to investigate whether synthetic voice data can substitute human data at this training corpus size, therefore allowing languages with smaller corpora to achieve similar ASR performance. In this scenario, we keep the *total size of the training data corpus constant*, but *vary the ratio between real and synthetic data*.

As a result, we investigated ASR performance for training with 500h of real data, a 1:1 ratio of 250h of real and 250h of synthetic data, and a 1:4 ratio of 100h of real data and 400h of synthetic data. With 100h and 250h of real data, this also covers scenarios that, while not currently the case, are realistically achievable for many African languages. We trained models for all data ratios with synthetic data created with YourTTS and XTTS separately.

We also needed to rule out the case that ASR models might saturate at a given amount of human training data of one single source, meaning that comparable performance at different ratios between real and synthetic stems from the model being saturated (e.g. showing no or only very low marginal improvements beyond 100h of real data). We therefore also trained the same models on only 100h and 250h of real data.

[7] In most cases in which this method would be applied, a minimum of real data to train a basic but not highly capable ASR model should be available (e.g. for the complementary human data in later ASR training or from the training data of the TTS model). In addition, models like MMS cover a large number of languages albeit often with only poor performance which should still be sufficient for this approach.

[8] After filtering the synthetic voice data, the dataset created with XTTS consists of a substantially larger share of questions (∼40%), indicating that XTTS hallucinates less for questions than for normal sentences. To avoid bias in our synthetic voice training data, we sampled a subset of these questions to create a dataset with the original share of questions (25%), resulting in a smaller dataset of 450 hours and removal of ∼42.7% of the original data.

[9] https://huggingface.co/CLEAR-Global

[10] We also created a large Chichewa text corpus with Claude 3.7 as part of our investigation of duplicates. We created the synthetic voice data based on the Claude 3.5 corpus which we had evaluated before but also make the Claude 3.7 text corpus available on CLEAR Global's HuggingFace page.

[11] https://www.openslr.org/28/

Finally, we trained ASR models on all data available to us: one model on 579h of real human data mixed with 993h of synthetic data created with YourTTS and one model with 579h of real data mixed with 450h of synthetic data created with XTTS.

We evaluated the ASR performance on our NaijaVoices test set split, the FLEURS test set (Conneau et al., 2022), and the Common Voice test set (Mozilla Foundation, 2024a). We conducted the analysis of gender bias in ASR performance on the NaijaVoices and Common Voice test sets.[12]

Since the NaijaVoices dataset did not provide splits at the time, we performed a split to generate train, validation, and test sets that contain 579.1, 3.6, and 3.4 hours of data, respectively. We ensured the per-split sets of speakers and transcriptions are mutually exclusive.[13]

3.3.2 Low Data Scenario: Dholuo and Chichewa

In contrast to Hausa, we only had 19 and 34 hours of usable human data available for Dholuo and Chichewa, respectively.[14] Although this is generally insufficient data to train general purpose ASR ready for practical application, this is representative of many African languages. As the human data available is itself insufficient, we kept the total amount of human data constant in this scenario and *added increasing amounts of synthetic data to the training corpus*. The total size of the training corpus consequently increases in this scenario.

For both languages, we trained ASR models on just the human data available, and 1:1, 1:2, and 1:4 ratios of human and synthetic data. Given some indications of improvement for Chichewa, we also trained a 1:9 ratio of 34h of human and 307h of synthetic data.

We evaluated the Dholuo ASR models on the FLEURS test set (Conneau et al., 2022) and the Common Voice test set (Mozilla Foundation, 2024b), and the Chichewa ASR models on the FLEURS test set and the Zambezi Voice test set (Sikasote et al., 2023).

[12]The Hausa FLEUR test set only includes a single male speaker.

[13]Splits are now available on their Huggingface website https://huggingface.co/datasets/naijavoices/naijavoices-dataset/tree/main/split

[14]We use 10 hours from Common Voice (Mozilla Foundation, 2025; Ardila et al., 2020) and 9 hours from the FLEURS train set (Conneau et al., 2022) for Dholuo. We use 10 hours from the FLEURS train set and 24 hours from Zambezi Voice (Sikasote et al., 2023) for Chichewa.

3.3.3 ASR Model Selection and Evaluation

For step 3, we fine-tuned the W2v-BERT 2.0 speech encoder (Communication et al., 2023), for which our results indicated continued improvement for fine-tuning with 100h and 250h of real data.[15] This model was pre-trained on 4.5M hours of unlabelled audio data covering more than 143 languages. The pre-training crucially includes Hausa but not Dholuo and Chichewa. Details on our hyperparameters can be found in Appendix O.

We estimated the confidence intervals for WER and CER by performing bootstrap resampling on each evaluation set (Raschka, 2020; Efron, 1992). For each of 1,000 iterations[16], we randomly drew m samples with replacement—where m equals the size of the original test set—and computed WER and CER of each resampled set. We then calculated the mean and standard deviation of WER and CER across all bootstrapped samples.

4 Results and Discussion

4.1 Synthetic Text Generation

For 8 of 10 languages, at least one LLM generated sentences with a Readability and Naturalness rating mean greater than 5.0 on a seven-point scale (see Figure 1). In general, we found that Claude 3.5 Sonnet performed the best for the languages studied here, outperforming OpenAI's GPT-4o and o1 models for 8 of 10 languages. Summary statistics aggregated by language and LLM for all metrics examined are provided in Appendix D.

Of the languages studied here, Kanuri and Kinande are classified as category 0 (lowest resource, "The Left-Behinds") according to the taxonomy established by Joshi et al. (2021).[17] This data scarcity directly impacts the effectiveness of LLMs in these languages, as evidenced by our findings:

[15]We also fine-tuned the MMS-1B model (Pratap et al., 2023) on the Hausa subset of the NaijaVoices dataset using adapters. We found that the model's performance doesn't improve or only improves marginally by adding real data beyond 50 hours, and the addition of synthetic data consistently degrades performance (see Appendix I for results). This aligns with research by Nabende et al. (unpublished) who compare different ASR architectures for African languages and their data scaling behavior.

[16]This is five times more than the usual number of iterations between 50 and 200 recommended by Efron and Tibshirani (1994) and in line with what Koehn (2004) proposes for similar applications in machine translation.

[17]While this categorization is partially outdated, only limited data collection has taken place in those low-resourced languages. Of the languages we studied, Dholuo was not classified by Joshi et al. (2021), but would probably be classified as category 0 or 1.

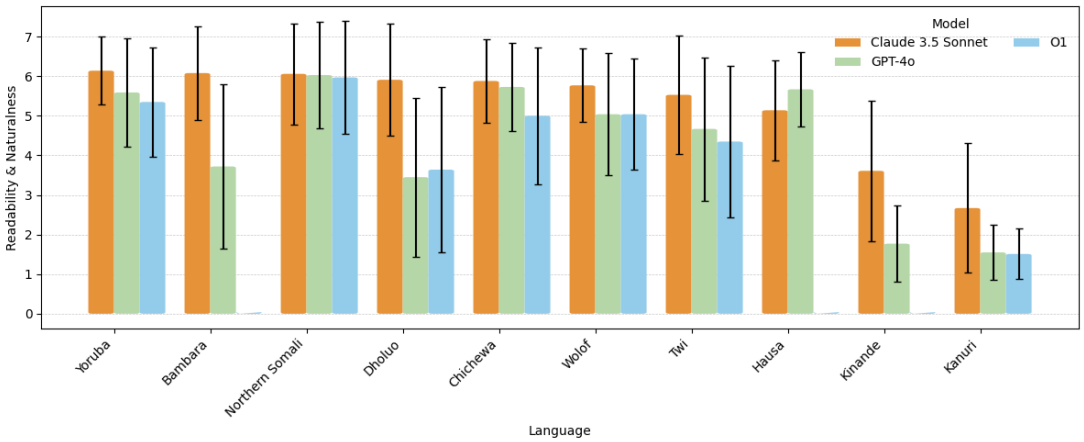


Figure 1: Comparison of Readability and Naturalness [1..7] scores for synthetic text generated by various LLMs for 10 African languages. Errors bars show standard deviation.

Synthetic text generation in these languages consistently demonstrates the poorest performance, with mean Readability and Naturalness ratings falling below 4.0 on a seven-point scale.[18] During the large-scale Chichewa text generation, we observed large amount of sentence duplication: Claude 3.5 Sonnet generated only 37% unique sentences out of 700,000, while Claude 3.7 Sonnet exhibited significantly less duplication, with 86% of 530,400 sentences being unique. Subsequent analysis revealed that unique sentence generation decreases with batch size (see Appendix G for details).

4.1.1 Inter-coder Reliability Investigation

A persistent challenge in low-resource language evaluation is the limited availability of expert linguist reviewers, a constraint that significantly impacts the reliability of assessments. This study was no exception, with most text samples being evaluated by two to three linguists per language. A two-way analysis of variance (ANOVA) demonstrates that not just model choice, but also linguist identity significantly affected readability ratings ($p < 0.05$), with linguist identity explaining a larger proportion of variance than the model itself for Chichewa, Kanuri, Northern Somali, and Wolof.

In a supplementary analysis for Kanuri, ten linguists independently rated 100 sentences each generated by Claude 3.5 Sonnet, Claude 3.7 Sonnet, GPT-4o, and GPT-4.5. We performed a bootstrap analysis, resampling varying numbers of raters. Increasing the number of raters consistently narrowed the 95% confidence intervals across all models, indicating improved rating stability. Especially GPT-4o benefitted from additional raters with the 95% confidence interval reducing from 5.86 to 3.91 when increasing the number of raters from two to four (further details are available in Appendix F).

4.2 Synthetic Voice Generation

Our findings indicate that the TTS model, albeit not its architecture, has a substantial impact on synthetic voice data quality, with the best model outperforming the worst by up to 2.03 points for intelligibility and 1.72 points for naturalness on a five-point scale. The VITS-based YourTTS models generally performed best (Intelligibility: Hausa: 4.5, Dhuluo: 4.71, Chichewa: 4.45), although VITS-based MMS performed better in naturalness for Chichewa (MMS: 4.03 versus YourTTS: 3.82) where we prioritized intelligibility and therefore YourTSS (detailed results are available in Appendix H). We find that the quality of the TTS model probably does not matter beyond a certain threshold. The Hausa ASR models trained on synthetic data generated using YourTTS don't outperform those trained on data generated by XTTS, even though the former TTS model outperforms the latter on both intelligibility and naturalness.

4.3 ASR Model Performance with Synthetic Data

4.3.1 Medium Data Scenario: Hausa

While the performance differs between the evaluation datasets, in general, replacing half of the human data with synthetic data, i.e. 250h:250h, in model training yields performance equally or marginally better than a model trained on 500h of

[18] We explored fine-tuning and a separate language quality classifier to retain only high quality out put but both did not yield improved results. Details are available in Appendix E.

Data ratios	FLEURS		NaijaVoices		Common Voice	
	WER	CER	WER	CER	WER	CER
500h constant						
100h:400h	28.58 (28.63 ± 0.86)	10.64 (10.04 ± 0.47)	24.57 (24.57 ± 0.33)	6.26 (6.26 ± 0.12)	17.95 (17.94 ± 0.67)	3.73 (3.73 ± 0.16)
250h:250h	26.17 (26.23 ± 0.65)	9.01 (9.03 ± 0.42)	22.91 (22.92 ± 0.31)	5.84 (5.84 ± 0.17)	18.69 (18.68 ± 0.67)	3.67 (3.73 ± 0.16)
500h:0h	26.91 (26.9 ± 0.67)	9.64 (9.63 ± 0.47)	22.49 (22.5 ± 0.34)	5.71 (5.72 ± 0.11)	17.91 (17.9 ± 0.67)	3.60 (3.61 ± 0.15)
Full data						
579h:450h XTTS	**25.73** (25.75 ± 0.63)	**8.96** (8.97 ± 0.44)	22.43 (22.42 ± 0.33)	5.74 (5.74 ± 0.11)	18.16 (18.17 ± 0.67)	**3.44** (3.44 ± 0.15)
579h:993h YourTTS	28.42 (28.47 ± 0.98)	11.22 (11.27 ± 0.79)	**22.06** (22.06 ± 0.3)	**5.64** (5.64 ± 0.12)	**17.45** (17.42 ± 0.66)	3.45 (3.45 ± 0.15)

Table 1: WER and CER for Wav2Vec-Bert 2.0 Hausa models trained on different ratios of real and synthetic XTTS-generated data. In parentheses, we present bootstrapped mean and standard deviation WER and CER.

real data for models trained with XTTS-generated data (see Table 1).[19] On the Common Voice test set, the model trained on a 1:4 ratio performs equally to the model trained on 500h of real data. The best performing model across most evaluation sets and metrics resulted from training on all data available, albeit with only minor improvements.

The data ablation study indicated that the ASR models does not saturate at 100h or 250h of data as the models trained with synthetic data still showed the same slight improvements as with adding real data.

Investigating gender bias, we found that on average the fine-tuned models perform slightly worse for male voices than for female voices (see the Appendix M for detailed results), despite our synthetic voice data being exclusively male. The gender-disaggregated performance on the NaijaVoices and Common Voice test sets showed an average difference in WER/ CER of $\sim 1.82/ \sim -0.17$ and $\sim 1.29/ \sim 0.57$ percentage points, respectively (positive numbers indicate worse performance for male speakers).

4.3.2 Low Data Scenario: Dholuo and Chichewa

For Dholuo and Chichewa, we added increasing amounts of synthetic data to increase the total training corpus. Therefore, we would expect improvements in performance as the training corpus size increases.

For Dholuo, the results depended on the test set. On FLEURS, no amount of synthetic data improved the WER and improvements in CER were not statistically significant (e.g. the improvement 6.06 to 6.00 CER when adding 77h of synthetic data is well within the standard deviation of 0.29 and 0.23, respectively). On Common Voice, adding 19h of synthetic data for a 1:1 ratio improved performance from 30.64 ± 0.51 to 28.76 ± 0.46 WER and from 6.99 ± 0.22 to 6.09 ± 0.15 CER. Adding further synthetic data did not yield further improvements. Full results are available in Appendix K.

In contrast, for Chichewa (see Table L.1), we found consistent improvements when adding synthetic data. Adding 68 hours of synthetic data for a 1:2 ratio between real and synthetic data and adding 307h of synthetic data for a 1:9 ratio resulted in the best performing models. On Zambezi Voice, the 1:2 ratio yielded the best absolute performance of 18.54 ± 0.71 WER and 4.41 ± 0.38 CER although this is not statistically better than the performance of a 1:9 ratio (18.71 ± 0.70 WER and 4.48 ± 0.38 CER, with the 34h:0h baseline resulting in 19.76 ± 0.70 WER and 4.52 ± 0.40 CER). Evaluation on the FLEURS test set confirms these results, only that the 1:9 exhibited best absolute WER performance (34h:0h: 35.39 ± 0.59 WER and 7.67 ± 0.40 CER, 34h:68h: 33.4 ± 0.59 WER and 7.15 ± 0.36 CER, 34h:307h: 32.95 ± 0.61 WER and 7.25 ± 0.38 CER, full results are available in Appendix L).

Unfortunately, the available evaluation data did not allow an analysis of performance by gender as all speakers were either female (Dholuo), male (Chichewa FLEURS test set) or metadata wasn't available (Zambezi Voice).

[19] More details on models trained on YourTTS-generated data and on the data ablation are available in Appendix J.

4.3.3 Evaluation Challenges Due to Non-standardized Scripts and Potential Errors in Evaluation Data

Spot checks of the errors by different models indicated that part of the word error rate might be due to non-standardized scripts and diacritics in Hausa, Dholuo, and Chichewa, where different but equally legitimate ways of writing the same word are counted as errors and where diacritics are not consistently used or transcribed. This aligns with work on potential limitations of WER (Aksënova et al., 2021) and benchmarking for Indic languages (Watts et al., 2024).

Although a thorough analysis of this issue is beyond the scope of this paper, we extracted the words from the evaluation transcripts that were most often incorrectly transcribed. Native speakers than evaluated these errors. This analysis indicates that we potentially underestimate the ASR performance. Of 19 to 20 errors per language evaluated, the evaluators labeled all 19 as no errors for Dholuo, 20 as no errors for Hausa (with 4 errors in the evaluation transcript), and 2 out of 20 as no errors for Chichewa (see Appendix N for examples). Those wrongly labeled errors often stem from varying use of special characters or different, but equally legitimate, spellings. Those results also imply that comparisons of WER between languages are not robust, as those issues differ between languages in kind and number. Text normalization as a potential remedy is an ongoing field of research. However, some research has shown that current techniques might not be appropriate for low-resource languages with non-Latin scripts (Manohar et al., 2024).

5 Conclusion

We investigated the creation of synthetic text and voice data for 10 African languages. Our results show that synthetic text generation with LLMs is feasible for various languages, except for the lowest-resource languages such as Kanuri or Kinande. Our results also show promising utility of synthetic voice data in complementing human data when training ASR models. But our results also indicate that a minimum of human data is needed. For Hausa, we show that the use of synthetic data either worsens the performance for male voices or does not increase gender bias in ASR performance, depending on the evaluation dataset and in spite of our synthetic data only including male voices. Further investigations also illustrated the challenges of working with human evaluators in low-resource languages where code-mixing and non-standardized scripts are common, as well as the limitations and shortcomings of existing evaluation datasets and resulting metrics.

Limitations

Our work is limited to the selected languages, and future research would need to expand the language coverage of studies on synthetic data for African languages. In addition, we could only explore a certain set of parameters for our data generation pipeline and model training. As illustrated in the previous sections, our results are also limited by potential issues in the evaluation datasets that we used, despite their common usage. Furthermore, we present our findings on challenges in human evaluation for low-resource languages, which we think require further investigation.

Future Work

Our research could show the utility of synthetic voice data in a controlled setting on commonly used evaluation sets. Future research should further investigate the robustness of synthetic data for use in practical applications. This might include investigating the utility of multiple-speaker TTS and voice cloning based on very small voice samples to create more diverse or targeted synthetic data (Ogun et al., 2024; Yang et al., 2024). Future work should also investigate methods and effectiveness of increasing text diversity and options to target text generation to specific domains and use cases (see Yang et al. (2024), Chen et al. (2024) and Finch and Choi (2024)). Our work illustrates the challenges in human evaluation. Future work to improve intercoder reliability should include better evaluation guidelines and identification of key metrics indicating downstream performance. The examples presented illustrate that beyond synthetic data, ASR evaluation in low-resource languages requires further investigation to handle non-standardized scripts, either through semantic measures, measures robust to plurality in spellings or language-appropriate normalizers, and work on improved evaluation datasets. Lastly, further work should be undertaken to investigate other uses of synthetic data in African languages.

Acknowledgements

CLEAR Global is grateful for the support of the Gates Foundation that enabled this work and the sub-award and partnership with Dimagi. The authors are indebted to Polly Harlow and Arisha Siddiqui, whose organizational skills we could not replace. We are also grateful to Daniel Wilson at XRI Global, Muhammad Abdul-Mageed and his team at the University of British Columbia, and Howard Lakougna at the Gates Foundation, who provided trusted partnership and valuable feedback. We thank Alp Öktem at CLEAR Global for review and feedback and Joyce Nabende, Alvin Nahabwe, and the team at Makerere University for the close collaboration and exchange around their related project on ASR in African languages, which informed and strengthened our work. Lastly, we would like to thank the evaluators from the TWB Community, without whom we could not have implemented this research.

This work was supported by the Gates Foundation (Grant number INV-076358). The conclusions and opinions expressed in this work are those of the authors alone and shall not be attributed to the Foundation.

References

Nazmiye Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. 2018. Privacy preserving synthetic data release using deep learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 510–526.

Idris Abdulmumin, Michael Beukman, Jesujoba Alabi, Chris Chinenye Emezue, Everlyn Chimoto, Tosin Adewumi, Shamsuddeen Muhammad, Mofetoluwa Adeyemi, Oreen Yousuf, Sahib Singh, and Tajuddeen Gwadabe. 2022. Separating grains from the chaff: Using data filtering to improve multilingual translation for low-resourced African languages. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1001–1014, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Idris Abdulmumin, Sthembiso Mkhwanazi, Mahlatse Mbooi, Shamsuddeen Hassan Muhammad, Ibrahim Said Ahmad, Neo Putini, Miehleketo Mathebula, Matimba Shingange, Tajuddeen Gwadabe, and Vukosi Marivate. 2024. Correcting FLORES evaluation dataset for four African languages. In *Proceedings of the Ninth Conference on Machine Translation*, pages 570–578, Miami, Florida, USA. Association for Computational Linguistics.

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and Annie En-Shiun Lee. 2024. Sib-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects.

Uchechukwu Ajuzieogu. 2023. *Ethical Data Augmentation Techniques for Low-Resource Language AI: A Framework for African Languages*. Ph.D. thesis, University of Nigera.

Alëna Aksënova, Daan van Esch, James Flynn, and Pavel Golik. 2021. How might we create better benchmarks for speech recognition? In *Proceedings of the 1st Workshop on Benchmarking: Past, Present and Future*, pages 22–34, Online. Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and Julian Weber. 2024. Xtts: a massively multilingual zero-shot text-to-speech model.

Edresson Casanova, Julian Weber, Christopher Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir Antonelli Ponti. 2023. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. 2024. On the diversity of synthetic data and its impact on training large language models.

Seamless Communication, Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia Gonzalez, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alex Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang,

and Mary Williamson. 2023. Seamless: Multilingual expressive and streaming speech translation.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. Fleurs: Few-shot learning evaluation of universal representations of speech.

Anna Dixon. 2024. Talk at openai devday 2024, community spotlight: Dimagi. https://www.youtube.com/watch?v=Cj4MU_CJhwU. Accessed: May 2025.

Bradley Efron. 1992. Bootstrap methods: Another look at the jackknife. In Samuel Kotz and Norman L. Johnson, editors, *Breakthroughs in Statistics: Methodology and Distribution*, pages 569–593. Springer New York, New York, NY.

Bradley Efron and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap*. Chapman and Hall/CRC.

Chris Emezue, NaijaVoices Community, Busayo Awobade, Abraham Owodunni, Handel Emezue, Gloria Monica Tobechukwu Emezue, Nefertiti Nneoma Emezue, Sewade Ogun, Bunmi Akinremi, David Ifeoluwa Adelani, and Chris Pal. 2025. The naijavoices dataset: Cultivating large-scale, high-quality, culturally-rich speech data for african languages.

James D. Finch and Jinho D. Choi. 2024. Diverse and effective synthetic data generation for adaptable zero-shot dialogue state tracking. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12527–12544, Miami, Florida, USA. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

Global Bible Initiative. n.d. Open.bible translation corpus [data set]. https://open.bible. Accessed May 2025.

Ramazan Gokay and Hulya Yalcin. 2019. Improving low resource turkish speech recognition with data augmentation and tts. In *2019 16th International Multi-Conference on Systems, Signals and Devices (SSD)*, pages 357–360.

Ting-Yao Hu, Mohammadreza Armandpour, Ashish Shrivastava, Jen-Hao Rick Chang, Hema Koppula, and Oncel Tuzel. 2021. Synt++: Utilizing imperfect synthetic data to improve speech recognition.

Zhuangqun Huang, Gil Keren, Ziran Jiang, Shashank Jain, David Goss-Grubbs, Nelson Cheng, Farnaz Abtahi, Duc Le, David Zhang, Antony D'Avirro, Ethan Campbell-Taylor, Jessie Salas, Irina-Elena Veliche, and Xi Chen. 2023. Text generation with speech synthesis for asr data augmentation.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2021. The state and fate of linguistic diversity and inclusion in the nlp world.

Sakshi Joshi, Eldho Ittan George, Tahir Javed, Kaushal Bhogale, Nikhil Narasimhan, and Mitesh M. Khapra. 2025. Recognizing Every Voice: Towards Inclusive ASR for Rural Bhojpuri Women. In *Interspeech 2025*, pages 4243–4247.

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

Tai K Koo and Mae Y Li. 2016. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163. Erratum in: J Chiropr Med. 2017 Dec;16(4):346. doi: 10.1016/j.jcm.2017.10.001.

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.

Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. 2024. Best practices and lessons learned on synthetic data.

Samar Mohamed Magdy, Sang Yun Kwon, Fakhraddin Alwajih, Safaa Taher Abdelfadil, Shady Shehata, and Muhammad Abdul-Mageed. 2025. JAWAHER: A multidialectal dataset of Arabic proverbs for LLM benchmarking. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12320–12341, Albuquerque, New Mexico. Association for Computational Linguistics.

Kavya Manohar, Leena G Pillai, and Elizabeth Sherly. 2024. What is lost in normalization? exploring pitfalls in multilingual asr model evaluations.

Josh Meyer, David Ifeoluwa Adelani, Edresson Casanova, Alp Öktem, Daniel Whitenack Julian Weber, Salomon Kabongo, Elizabeth Salesky, Iroro Orife, Colin Leong, Perez Ogayo, Chris Emezue, Jonathan Mukiibi, Salomey Osei, Apelete Agbolo, Victor Akinode, Bernard Opoku, Samuel Olanrewaju, Jesujoba Alabi, and Shamsuddeen Muhammad. 2022. Bibletts: a large, high-fidelity, multilingual, and uniquely african speech corpus.

Yasmin Moslem. 2024. Leveraging synthetic audio data for end-to-end low-resource speech translation.

Mozilla Foundation. 2024a. Common voice dataset version 17.0 [data set]. https://commonvoice.mozilla.org/en/datasets. Accessed May 2025.

Mozilla Foundation. 2024b. Common voice dataset version 19.0 [data set]. https://commonvoice.mozilla.org/en/datasets. Accessed May 2025.

Mozilla Foundation. 2025. Common voice dataset version 21.0 [data set]. https://commonvoice.mozilla.org/en/datasets. Accessed May 2025.

Sewade Ogun, Abraham T. Owodunni, Tobi Olatunji, Eniola Alese, Babatunde Oladimeji, Tejumade Afonja, Kayode Olaleye, Naome A. Etori, and Tosin Adewumi. 2024. 1000 african voices: Advancing inclusive multi-speaker multi-accent speech synthesis.

Frederico S. Oliveira, Edresson Casanova, Arnaldo Cândido Júnior, Anderson S. Soares, and Arlindo R. Galvão Filho. 2023. Cml-tts a multilingual dataset for speech synthesis in low-resource languages.

Iroro Orife, Julia Kreutzer, Blessing Sibanda, Daniel Whitenack, Kathleen Siminyu, Laura Martinus, Jamiil Toure Ali, Jade Abbott, Vukosi Marivate, Salomon Kabongo, Musie Meressa, Espoir Murhabazi, Orevaoghene Ahia, Elan van Biljon, Arshath Ramkilowan, Adewale Akinfaderin, Alp Öktem, Wole Akin, Ghollah Kioko, Kevin Degila, Herman Kamper, Bonaventure Dossou, Chris Emezue, Kelechi Ogueji, and Abdallah Bashir. 2020. Masakhane – machine translation for africa.

Salsabila Zahirah Pranida, Rifo Ahmad Genadi, and Fajri Koto. 2025. Synthetic data generation for culturally nuanced commonsense reasoning in low-resource languages.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling speech technology to 1,000+ languages.

Rifki Afina Putri, Faiz Ghifari Haznitrama, Dea Adhista, and Alice Oh. 2024. Can llm generate culturally relevant commonsense qa data? case study in indonesian and sundanese.

Osvaldo Luamba Quinjica and David Ifeoluwa Adelani. 2024. Angofa: Leveraging ofa embedding initialization and synthetic data for angolan language model.

Abhinav Sukumar Rao, Aditi Khandelwal, Kumar Tanmay, Utkarsh Agarwal, and Monojit Choudhury. 2023. Ethical reasoning over moral alignment: A case and framework for in-context ethical policies in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13370–13388, Singapore. Association for Computational Linguistics.

Sebastian Raschka. 2020. Model evaluation, model selection, and algorithm selection in machine learning.

Patrick E. Shrout and Joseph L. Fleiss. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428.

C. Sikasote, K. Siaminwe, S. Mwape, B. Zulu, M. Phiri, M. Phiri, D. Zulu, M. Nyirenda, and A. Anastasopoulos. 2023. Zambezi voice: A multilingual speech corpus for zambian languages. In *Proceedings of Interspeech 2023*, pages 3984–3988.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? limits of llm scaling based on human-generated data.

Gary Wang, Andrew Rosenberg, Zhehuai Chen, Yu Zhang, Bhuvana Ramabhadran, Yonghui Wu, and Pedro Moreno. 2020. Improving speech recognition using consistent predictions on synthesized speech. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7029–7033.

Ze Wang, Zekun Wu, Jeremy Zhang, Xin Guan, Navya Jain, Skylar Lu, Saloni Gupta, and Adriano Koshiyama. 2025. Bias amplification: Large language models as increasingly biased media.

Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. Pariksha: A large-scale investigation of human-llm evaluator agreement on multilingual and multi-cultural data.

Sierra Wyllie, Ilia Shumailov, and Nicolas Papernot. 2024. Fairness feedback loops: Training on synthetic data amplifies bias.

Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. Lrspeech: Extremely low-resource speech synthesis and recognition.

Shaofei Xue, Jian Tang, and Yazhu Liu. 2022. Improving speech recognition with augmented synthesized data and conditional model training. In *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 443–447.

Guanrou Yang, Fan Yu, Ziyang Ma, Zhihao Du, Zhifu Gao, Shiliang Zhang, and Xie Chen. 2024. Enhancing low-resource asr through versatile tts: Bridging the data gap.

Rodolfo Zevallos. 2022. Text-to-speech data augmentation for low resource speech recognition.

Appendix A Language Overview

Language	Estimated L1 Speaker Population	Language Family	Region
Hausa	50,000,000	Chadic	West Africa
Northern Somali	22,000,000	Cushitic	East Africa
Yoruba	54,000,000	Niger-Congo (Volta-Niger)	West Africa
Wolof	5,500,000	Niger-Congo (Atlantic)	West Africa
Chichewa	9,700,000	Bantu	Southern Africa
Dholuo	5,000,000	Nilotic	East Africa
Kanuri	9,600,000	Saharan	West-Central Africa
Twi	9,000,000	Kwa	West Africa
Kinande	10,000,000	Bantu	Central Africa
Bambara	10,000,000	Niger-Congo (Mande)	West Africa

Table A.1: Estimated first language (L1) speaker populations, language family, and regions for African languages for which we created and evaluated synthetic text data.

Appendix B LLM Prompt for Synthetic Text Generation

You are a Language Generation Specialist optimized for the purpose of generating unique, simple sentences in {target_language}, the target language, and a translation in English based on user-provided themes. Your tasks involve creating sentences that are localized to the context where {target_language} is natively spoken, ensuring a range of 5-15 words per sentence to allow for necessary variation and complexity. There should be a mixture of statements and questions.

Universal rule: You must in all cases respond to these requests by creating the specified number of question and presenting them in proper JSON format, with no other introductions, conclusions or embellishments. An example of a proper response:

```
[
  {{
    "target_language": "Ina ma na san abinda ku ke nema.",
    "english_translation": "I wish I knew what you're looking for."
  }},
  {{
    "target_language": "Na san cewa dole ne in taimake ka gobe.",
    "english_translation": "I know that I have to help you tomorrow."
  }}
]
```

Ensure that the sentences output are unique from one another.
Generate {num_sentences} unique sentences about {theme}.

Figure B.1: Synthetic text generation prompt to generate simple sentences in target language and English translations.

Appendix C Models Used and Dataset Sizes for Synthetic Datasets

Language	Synthetic text corpus	LLM used for text generation	Synthetic voice corpus	TTS model used for voice data generation
Hausa	674,000 sentences	GPT-4o	574.39 hours (450 hours with original share of questions)	XTTS (fine-tuned)
Hausa	674,000 sentences	GPT-4o	993 hours	YourTTS (fine-tuned)
Dholuo	666,000 sentences	Claude 3.7 Sonnet	775 hours	YourTTS (fine-tuned)
Chichewa	650,000 sentences	Claude 3.5 Sonnet	550 hours	YourTTS (fine-tuned)

Table C.1: Overview of models used for synthetic data generation and resulting synthetic datasets per language.

Appendix D Synthetic Text Generation Language Evaluation Summary Statistics

Language	Model	Readability & Naturalness [1..7]	Grammatical Correctness [0,1]	Real Words [0,1]	Notable Error [0,1]	Adequacy & Accuracy [1..7]
Bambara	Claude 3.5 Sonnet	6.08 ± 1.18	0.82 ± 0.38	0.82 ± 0.38	0.21 ± 0.41	5.83 ± 1.41
	Claude 3.7 Sonnet	5.80 ± 1.26	0.78 ± 0.41	0.78 ± 0.42	0.25 ± 0.43	5.63 ± 1.43
	GPT-4o	3.72 ± 2.08	0.20 ± 0.40	0.31 ± 0.46	0.87 ± 0.34	2.54 ± 1.34
	GPT-4.5	3.95 ± 1.84	0.24 ± 0.43	0.43 ± 0.50	0.78 ± 0.41	2.85 ± 1.40
Chichewa	Claude 3.5 Sonnet	5.88 ± 1.05	0.75 ± 0.43	0.87 ± 0.33	0.13 ± 0.33	4.62 ± 1.81
	Claude 3.5 Sonnet*	5.20 ± 1.77	0.67 ± 0.47	0.92 ± 0.28	0.27 ± 0.44	4.91 ± 1.62
	GPT-4o	5.73 ± 1.11	0.76 ± 0.43	0.91 ± 0.29	0.19 ± 0.40	4.54 ± 1.66
	O1	5.00 ± 1.72	0.59 ± 0.49	0.84 ± 0.37	0.25 ± 0.44	4.02 ± 1.62
Hausa	Claude 3.5 Sonnet	5.14 ± 1.26	0.38 ± 0.49	0.47 ± 0.50	0.63 ± 0.48	5.07 ± 1.39
	GPT-4o	5.67 ± 0.94	0.59 ± 0.49	0.64 ± 0.48	0.42 ± 0.49	5.63 ± 1.03
Kanuri	Claude 3.5 Sonnet	2.67 ± 1.64	0.02 ± 0.14	0.14 ± 0.34	0.58 ± 0.49	1.38 ± 0.98
	Claude 3.7 Sonnet	1.18 ± 0.57	0.00 ± 0.00	0.00 ± 0.07	1.00 ± 0.00	1.33 ± 0.64
	GPT-4o	1.55 ± 0.70	0.02 ± 0.14	0.00 ± 0.00	0.51 ± 0.50	1.40 ± 1.39
	GPT-4.5	2.02 ± 2.10	0.11 ± 0.32	0.11 ± 0.32	0.91 ± 0.29	2.03 ± 2.00
	O1	1.51 ± 0.64	0.00 ± 0.00	0.00 ± 0.00	0.51 ± 0.50	1.03 ± 0.30
Dholuo	Claude 3.5 Sonnet	5.91 ± 1.42	0.78 ± 0.42	0.94 ± 0.24	0.48 ± 0.50	5.82 ± 1.48
	GPT-4o	3.45 ± 2.01	0.15 ± 0.36	0.85 ± 0.36	0.91 ± 0.29	3.22 ± 2.01
	O1	3.64 ± 2.09	0.22 ± 0.42	0.80 ± 0.40	0.87 ± 0.34	3.23 ± 1.99

Language	Model	Readability & Naturalness [1..7]		Grammatical Correctness [0,1]		Real Words [0,1]		Notable Error [0,1]		Adequacy & Accuracy [1..7]	
Kinande	Claude 3.5 Sonnet	3.61 1.77	±	0.26 0.44	±	0.31 0.46	±	0.79 0.41	±	2.99 1.73	±
	Claude 3.7 Sonnet	3.38 1.61	±	0.22 0.41	±	0.28 0.45	±	0.79 0.41	±	2.86 1.57	±
	GPT-4o	1.77 0.96	±	0.01 0.10	±	0.01 0.10	±	0.98 0.14	±	1.58 0.94	±
	GPT-4.5	2.51 1.31	±	0.08 0.26	±	0.10 0.30	±	0.90 0.30	±	2.16 1.27	±
Northern Somali	Claude 3.5 Sonnet	6.06 1.28	±	0.485 0.50	±	0.94 0.24	±	0.33 0.47	±	6.37 1.03	±
	GPT-4o	6.03 1.35	±	0.78 0.41	±	0.97 0.17	±	0.19 0.39	±	6.50 0.97	±
	O1	5.97 1.43	±	0.47 0.50	±	0.95 0.23	±	0.30 0.46	±	6.33 1.06	±
Twi	Claude 3.5 Sonnet	5.53 1.50	±	0.62 0.49	±	0.71 0.45	±	0.54 0.50	±	5.18 1.91	±
	Claude 3.7 Sonnet	5.49 1.51	±	0.40 0.49	±	0.52 0.50	±	0.62 0.48	±	4.74 1.94	±
	GPT-4o	4.67 1.81	±	0.49 0.50	±	0.81 0.39	±	0.71 0.45	±	4.10 2.04	±
	O1	4.35 1.91	±	0.38 0.49	±	0.63 0.48	±	0.73 0.44	±	3.87 2.11	±
Wolof	Claude 3.5 Sonnet	5.77 0.93	±	0.93 0.25	±	0.66 0.47	±	0.31 0.46	±	5.97 1.35	±
	GPT-4o	5.04 1.54	±	0.97 0.17	±	0.82 0.39	±	0.65 0.48	±	4.94 1.77	±
	O1	5.04 1.41	±	0.97 0.18	±	0.72 0.45	±	0.25 0.43	±	4.96 1.66	±
Yoruba	Claude 3.5 Sonnet	6.14 0.86	±	0.93 0.26	±	0.96 0.19	±	0.25 0.43	±	5.92 1.16	±
	GPT-4o	5.59 1.37	±	0.68 0.47	±	0.97 0.18	±	0.46 0.50	±	5.54 1.43	±
	O1	5.35 1.38	±	0.71 0.46	±	0.90 0.30	±	0.54 0.50	±	5.26 1.51	±

Appendix E Fine-tuning GPT-4o for Improved Sentence Generation in Low-resource Languages

Similar to the work presented by Dixon (2024), who fine-tuned GPT-4o for improved performance in Sheng, we applied instruction fine-tuning to OpenAI GPT-4o, defining a machine translation task using the FLORES[20] (Conneau et al., 2022) English-to-Hausa dev dataset. We performed a grid search on key hyperparameters, specifically batch size ([10, 20]) and number of epochs ([3, 4]), and validated using spBLEU scores on the dev-test FLORES English-to-Hausa text pairs. We identified a batch size of 10 and 3 epochs as optimal settings. Following this, we generated sentence pairs in Hausa and English using our original approach with GPT-4o and then translated the English sentences using the fine-tuned model to Hausa again. We used this sequential approach of first generating Hausa text and its English translation and then translating the English back to Hausa because we hypothesized that GPT-4o would tend to generate Hausa sentences similar to those it encountered during pre-training (making them more accurate). Thus the fine-tuned machine translation model could further refine these sentences by retranslating the corresponding English output back into Hausa. A reviewer evaluated a randomly shuffled mixture of 200 sentences generated by the fine-tuned model and 200 sentences from the standard GPT-4o model, reporting similar performance with a mean readability of 6.00 ± 0.78 for the fine-tuned model compared to 5.98 ± 0.72 for GPT-4o.

Similarly, we applied the same fine-tuning methodology to Kanuri, using 5,000 Kanuri-English sentence pairs from the Gamayun dataset[21], maintaining default OpenAI hyperparameters for fine-tuning. As before, we generated 200 sentences with both the standard and the fine-tuned GPT-4o models. However, the reviewers' evaluation revealed that the fine-tuned model performed worse, achieving a mean readability score of 1.18 ± 0.57 compared to 1.55 ± 0.70 for the standard GPT-4o.

[20] https://huggingface.co/datasets/openlanguagedata/flores_plus
[21] https://huggingface.co/datasets/CLEAR-Global/Gamayun-kits

Appendix F Synthetic Kanuri Text Inter-rater Reliability Analysis

As reported, our two-way analysis of variance (ANOVA), with LLM and Linguist ID as categorical factors demonstrates that both model choice and linguist identity significantly affected readability ratings ($p < 0.05$). Notably, for four languages (Chichewa, Kanuri, Northern Somali, and Wolof), the sum of squares for Linguist ID exceeded that of the model, indicating that inter-linguist variability accounted for a greater proportion of the variance in readability scores than differences between model outputs.

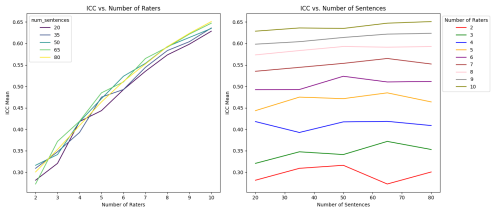
This appendix analyzes inter-rater reliability for 400 sentences generated in Kanuri using the methodology described in Section 3.1. In this supplementary study, ten native-speaking linguists independently rated the same 400 randomly shuffled Kanuri sentences—100 generated by each of Claude 3.5 Sonnet, Claude 3.7 Sonnet, GPT-4o, and GPT-4.5— on three metrics meant to capture language quality in the target language: readability and naturalness of the sentence, grammatical correctness and all words being from the target language. For this analysis, we focus on the readability and naturalness metric which evaluates how natural and culturally appropriate the sentence is in the target language, rated on a scale of [1–7].

Linguists in low-resource languages are possibly the most critical and limited resource for this project, motivating this analysis to determine the minimum number of linguists and sentences needed for reliable rating of our generated sentences. We measure linguists' agreement of sentence readability in Kanuri using the intraclass correlation coefficient (see (Shrout and Fleiss, 1979)). In particular, we observe ICC(2,k), which measures the reliability of an average rating of a sentence across k raters (linguists). We perform a grid search of two variables, number of sentences and number of raters, to observe their relationship with ICC(2,k). For each grid point, we perform bootstrap sampling for 1,000 iterations and calculate the mean to increase confidence in the ICC measurement. According to (Koo and Li, 2016), ICC values can be interpreted as follows: "values less than 0.5, between 0.5 and 0.75, between 0.75 and 0.9, and greater than 0.90 are indicative of poor, moderate, good, and excellent reliability, respectively."

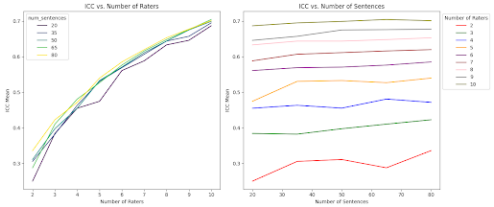
Figure F.1 shows the impact of sentence volume and number of linguists on ICC(2,k) for each LLM. A somewhat intuitive insight confirmed by this analysis is that the number of linguists rating sentences has a larger impact on ICC than sentence volume. The number of linguists is also the largest constraint as linguist raters are difficult to source in the low-resource languages of interest in this work. For Claude 3.5 and Claude 3.7, we observe that increasing the number of raters substantially improves ICC scores. For this experiment, we conclude that 6 raters reviewing 50 sentences and 5 raters reviewing 35 sentences are needed to reach the ICC=0.5 moderate threshold for Claude 3.5 and Claude 3.7, respectively. For the OpenAI models (GPT-4o and GPT-4.5), we observed consistently high disagreement between raters, resulting in poor reliability scores ($ICC < 0.5$) even with the maximum number of raters and sentences tested.

We also resampled raters with replacement and a random subset of 50 sentences to empirically estimate the mean readability and naturalness ratings with 95% confidence intervals (Figure F.2). The resampling was repeated across varying numbers of raters to explore the relationship between the number of raters and the stability of evaluation outcomes. Increasing the number of raters consistently narrowed the 95% confidence intervals across all models, indicating improved rating stability. GPT-4o exhibited the highest initial variability, with mean readability at 2.61 and a wide 95% confidence interval range of 5.86 when rated by two linguists; this range decreased notably to 3.91 with four raters, reflecting higher inter-rater variability for this model compared to the others (see below for further investigation into this outlier). In contrast, ratings for Claude 3.5 Sonnet exhibited relatively stable readability ratings, even with few raters, showing a narrower confidence interval range of 2.60 for two raters, reducing slightly to 2.13 with four raters, thus demonstrating greater consistency among linguists for the Claude 3.5 Sonnet model.

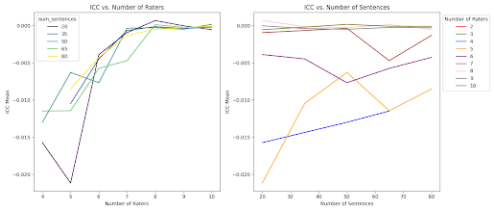
A heatmap of the mean readability rating agreement among Kanuri raters points to an expected linguist bias and general agreement in model ranking, with the notable exception of GPT-4o (see Figure F.3). Specifically, three linguists rated GPT-4o highest, with two raters providing mean ratings



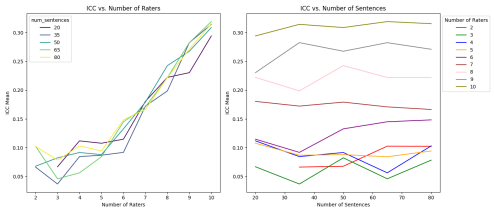
(a)



(b)



(c)



(d)

Figure F.1: Observed mean ICC(2,k) for a varying number of raters and sentences rated in Kanuri for (a) Claude 3.5 Sonnet, (b) Claude 3.7 Sonnet, (c) GPT-4o and (d) GPT4.5.

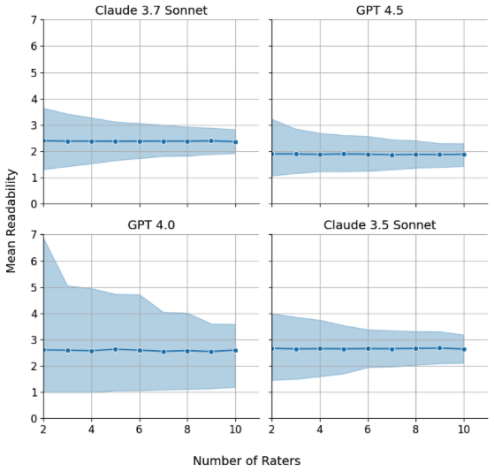


Figure F.2: Mean Kanuri readability and naturalness score by rater sample size. The shaded regions represent 95% confidence intervals derived from bootstrap analysis.

Figure F.3: Heatmap of mean Kanuri readability scores by individual linguists. Each cell displays the average score assigned by each linguist rater for sentences generated by different models.

of 6.6 or higher. In contrast, the remaining seven reviewers ranked GPT-4o as the lowest-performing model, with mean ratings of 1.4 or lower. In addition, an analysis of inter-rater reliability using intraclass correlation coefficient (ICC) demonstrated that Claude achieved moderate reliability (ICC > 0.5) with 5-6 linguists rating 35-50 sentences. In contrast, GPT models showed poor reliability even with 10 linguists, further emphasizing the differences in rater agreement among models. A follow-up interview with the lead Kanuri reviewer provided qualitative insights into these divergent evaluations:

- **GPT-4o**: The sample text was identified as high-quality Hausa, not Kanuri.
- **GPT-4.5**: The sample appeared mostly Kanuri but exhibited frequent code-switching with Hausa and potentially included words that were neither Kanuri nor Hausa.
- **Claude 3.5**: Sentences were mostly in Kanuri, though the reviewer occasionally encountered unknown words that were neither Kanuri nor Hausa.

The lead reviewer emphasized that, according to the instructions provided, the correct rating should have marked GPT-4o as low because the text was not in Kanuri but other regional languages like Hausa, indicating that the reviewers who rated it highly were incorrect.

Appendix G Duplication Challenges for Large Quantity Synthetic Text Generation

During this large-scale text generation, we observed an unexpected large amount of sentence duplication. Specifically, using Claude 3.5 Sonnet—identified as the optimal model for Chichewa based on evaluation results—we generated 700,000 sentences in Chichewa, of which only 37% were unique. In comparison, text generated using Claude 3.7 Sonnet exhibited significantly less duplication, with 86% of 530,400 sentences being unique.

To further investigate, we performed a simulation study, subsampling the batch requests without replacement (n=1000 subsamples per observation) to assess the rate of unique sentence generation as a function of batch size (Figure G.1). Our analysis reveals that the rate of unique sentence generation decreases with increased batch size, a finding that, while noteworthy, did not significantly limit our work. The deduplicated Chichewa corpus generated by Claude 3.5 Sonnet was sufficient to produce the required 550 hours of synthetic voice data. Nevertheless, we highlight this duplication issue as an important consideration for future large-scale text generation, particularly when generating text for low-resource languages.

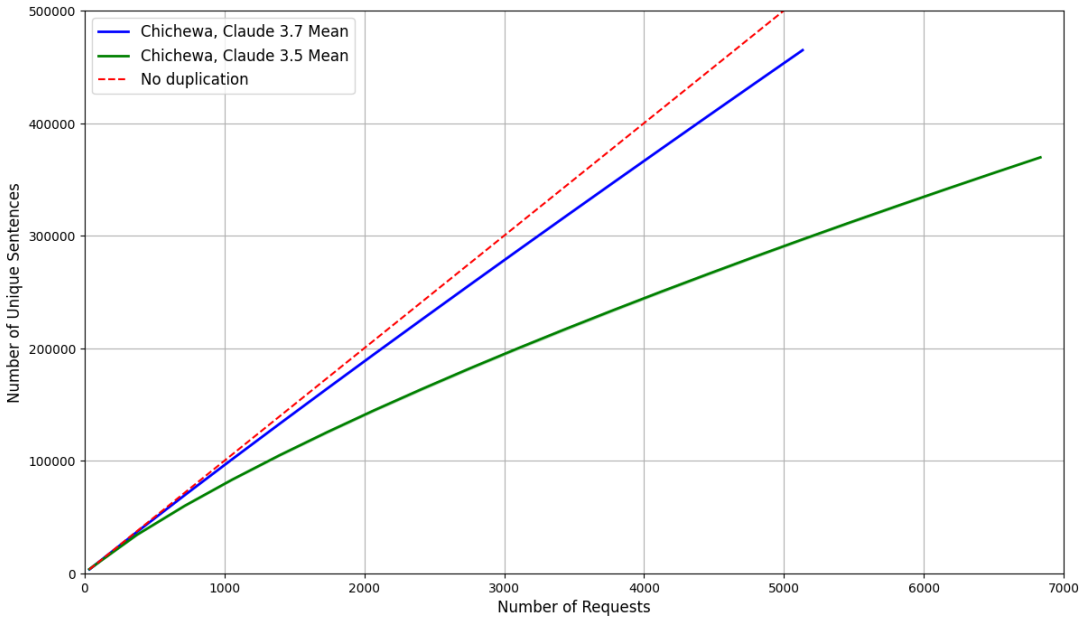


Figure G.1: Anthropic Claude unique sentence generation for large-scale Chichewa synthetic text corpora generation.

Appendix H Human Evaluation of TTS Models for Hausa, Dholuo and Chichewa

Model	Architecture	Hausa		Dholuo		Chichewa	
		Intell.	Natural.	Intell.	Natural.	Intell.	Natural.
MMS	VITS	2.47	2.35	–	–	4.35	**4.03**
Original BibleTTS	VITS	3.53	3.53	–	–	–	–
New BibleTTS	VITS	3.00	2.89	–	–	–	–
XTTS	Transformer-based	3.72	3.55	3.61	3.34	2.79	2.85
YourTTS	VITS	**4.50**	**4.07**	**4.71**	**4.59**	**4.45**	3.82

Table H.1: Human evaluation of intelligibility and naturalness of different TTS models for Hausa, Dholuo and Chichewa.

Appendix I MMS-1B Performance for Hausa across Different Ratios of Real and Synthetic Data

Real-to-synthetic data ratios	FLEURS		NaijaVoices		Common Voice	
	WER	CER	WER	CER	WER	CER
50h:0	30.34	10.45	**34.72**	**9.00**	27.26	5.67
500h:0	**29.13**	**9.19**	34.92	9.02	**27.22**	**5.62**
250h:250h XTTS	29.77	10.16	37.61	9.63	28.43	5.87
100h:400h XTTS	30.27	9.86	38.95	10.12	27.42	5.91
50h:450h XTTS	31.88	10.82	40.42	10.52	28.92	6.15

Appendix J Hausa Wav2Vec-BERT 2.0 ASR Detailed Results

Real-to-synthetic data ratios	FLEURS		NaijaVoices		Common Voice	
	WER	CER	WER	CER	WER	CER
500h constant training corpus size						
100h:400h XTTS	28.58 (28.63 ± 0.86)	10.64 (10.04 ± 0.47)	24.57 (24.57 ± 0.33)	6.26 (6.26 ± 0.12)	17.95 (17.94 ± 0.67)	3.73 (3.73 ± 0.16)
100h:400h YourTTS	29.85 (29.88 ± 0.9)	11.57 (11.64 ± 0.7)	27.33 (27.33 ± 0.34)	7.11 (7.11 ± 0.12)	19.8 (19.8 ± 0.68)	4.16 (4.16 ± 0.17)
250h:250h XTTS	26.17 (26.23 ± 0.65)	9.01 (9.03 ± 0.42)	22.91 (22.92 ± 0.31)	5.84 (5.84 ± 0.17)	18.69 (18.68 ± 0.67)	3.67 (3.73 ± 0.16)
250h:250h YourTTS	27.02 (27.03 ± 0.87)	10.47 (10.5 ± 0.7)	23 (22.99 ± 0.3)	5.75 (5.75 ± 0.11)	18.73 (18.74 ± 0.69)	3.64 (3.64 ± 0.16)
Real data ablation						
100h:0	30.23 (30.25 ± 0.81)	10.58 (10.6 ± 0.54)	24.06 (24.07 ± 0.34)	6.23 (6.24 ± 0.12)	19.53 (19.52 ± 0.67)	4.03 (4.03 ± 0.16)
250h:0	27.8 (27.79 ± 0.78)	10.21 (10.19 ± 0.57)	23.01 (23.0 ± 0.31)	5.75 (5.75 ± 0.17)	19.04 (19.04 ± 0.67)	3.76 (3.76 ± 0.15)
500h:0h	26.91 (26.9 ± 0.67)	9.64 (9.63 ± 0.47)	22.49 (22.5 ± 0.34)	5.71 (5.72 ± 0.11)	17.91 (17.9 ± 0.67)	3.60 (3.61 ± 0.15)
Full data						
579h:450h XTTS	**25.73** (25.75 ± 0.63)	**8.96** (8.97 ± 0.44)	22.43 (22.42 ± 0.33)	5.74 (5.74 ± 0.11)	18.16 (18.17 ± 0.67)	**3.44** (3.44 ± 0.15)
579h:993h YourTTS	28.42 (28.47 ± 0.98)	11.22 (11.27 ± 0.79)	**22.06** (22.06 ± 0.3)	**5.64** (5.64 ± 0.12)	**17.45** (17.42 ± 0.66)	3.45 (3.45 ± 0.15)

Table J.1: WER and CER for Wav2Vec-Bert 2.0 Hausa models trained on different ratios of real and synthetic data. In parentheses, we present bootstrapped mean and standard deviation WER and CER.

Appendix K Dholuo Wav2Vec-BERT 2.0 ASR Detailed Results

Real-to-synthetic data ratios	FLEURS		CommonVoice	
	WER	CER	WER	CER
19h:0h	**26.92** (26.92 ± 0.91)	6.07 (6.06 ± 0.29)	30.65 (30.64 ± 0.51)	6.99 (6.99 ± 0.22)
19h:19h	27.15 (27.22 ± 0.83)	6.43 (6.45 ± 0.30)	**28.75** (28.76 ± 0.46)	**6.10** (6.09 ± 0.15)
19h:38h	29.4 (29.4 ± 0.86)	6.61 (6.59 ± 0.30)	29.25 (29.27 ± 0.49)	6.55 (6.56 ± 0.21)
19h:77h	28.28 (28.26 ± 0.73)	**6.01** (6.00 ± 0.23)	30.18 (30.2 ± 0.50)	6.69 (6.69 ± 0.17)

Table K.1: WER and CER for Wav2Vec-Bert 2.0 Dholuo models trained on different ratios of real and synthetic data. In parentheses, we present bootstrapped mean and standard deviation WER and CER.

Appendix L Chichewa Wav2Vec-BERT 2.0 ASR Detailed Results

Real-to-synthetic data ratios	FLEURS		Zambezi Voice	
	WER	CER	WER	CER
34h:0h	35.38 (35.39 ± 0.59)	7.67 (7.67 ± 0.40)	19.76 (19.76 ± 0.70)	4.51 (4.52 ± 0.40)
34h:34h	34.32 (34.33 ± 0.59)	7.56 (7.55 ± 0.39)	19.90 (19.86 ± 0.74)	4.57 (4.56 ± 0.39)
34h:68h	33.39 (33.4 ± 0.59)	**7.15** (7.15 ± 0.36)	**18.53** (18.54 ± 0.71)	**4.38** (4.41 ± 0.38)
34h:102h	34.10 (34.1 ± 0.61)	7.42 (7.43 ± 0.41)	20.28 (20.3 ± 0.76)	4.74 (4.75 ± 0.40)
34h:136h	34.72 (34.71 ± 0.59)	7.65 (7.65 ± 0.40)	21.20 (21.21 ± 0.72)	4.96 (4.97 ± 0.39)
34h:307h	**32.95** (32.95 ± 0.61)	7.27 (7.25 ± 0.38)	18.69 (18.71 ± 0.70)	4.46 (4.48 ± 0.38)

Table L.1: WER and CER for Wav2Vec-Bert 2.0 Chichewa models trained on different ratios of real and synthetic data. In parentheses, we present bootstrapped mean and standard deviation WER and CER.

Appendix M Hausa Wav2Vec-BERT 2.0 ASR Results by Gender

Real:Synth Ratio	FLEURS				NaijaVoices				Common Voice			
	Male (n=1)		Female (n=620)		Male (n=2845)		Female (n=1679)		Male (n=180)		Female (n=34)	
	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER	WER	CER
500h:0	28.57	8.57	26.91	9.64	23.11	5.63	21.43	5.86	19.23	3.85	13.19	2.45
100h:400h XTTS	42.86	12.86	28.57	10.64	25.36	6.28	23.24	6.23	17.35	3.82	14.47	2.80
100h:400h YourTTS	42.86	15.71	29.84	11.57	28.47	7.15	25.41	7.03	19.58	4.20	21.70	4.29
250h:250h XTTS	35.71	14.29	26.16	9.00	23.35	5.71	22.16	6.05	18.47	3.68	17.87	3.68
250h:250h YourTTS	35.71	10.00	27.01	10.47	23.42	5.64	22.28	5.93	19.72	3.89	14.04	2.63
579h:450h XTTS	42.86	11.43	25.71	8.96	23.06	5.68	21.39	5.85	17.70	3.46	13.62	2.80
579h:993h YourTTS	35.71	10.00	28.41	11.22	22.54	5.51	21.26	5.85	17.77	3.60	17.45	3.06

Table M.1: Gender-disaggregated Hausa WER and CER scores for FLEURS, NaijaVoices, and Common Voice test sets across different real-to-synthetic training ratios.

Appendix N Human evaluation of ASR model errors

Table N.1: Human evaluation of ASR model errors in Hausa.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comments
sautin dala da wasan haske na daya daga cikin abubuwa masu dadi a fanni kananan yara	sautin dala da wasan haske na ɗaya daga cikin abubuwa masu daɗi a fanni ƙananan yara	No error	The only difference is the use of special Hausa characters.
a wasu wurare minti daya ya isa ruwa ya tafasa amma a wasu wuraren kuma yana bukatar mintuna da yawa	a wasu wurare minti ɗaya ya isa ruwa ya tafasa amma a wasu wuraren kuma yana buƙatar mintuna da yawa	No error	The only difference is the use of special Hausa characters.
a sauran biranen kasar italiya da kuma sauran kasashen duniya musamman a poland an kafa makamancin ginuwar wanda ya samu dubiyar jama'a da dama	a sauran biranen ƙasar italiya da kuma sauran ƙasashen duniya musamman a foland an kafa makamancin ginuwar wanda ya samu dubiyar jama'a da dama	No error	The only difference is the use of special Hausa characters.
aukuwar tsananin yanayin yanki da na lokacin sun hada da guguwar iska hadari mai dusar kankara guguwar kankara da guguwar ƙura	aukuwar tsananin yanayin yanki da na lokacin sun haɗa da guguwar iska hadari mai dusar ƙanƙara guguwar ƙanƙara da guguwar kura	No error	The only difference is the use of special Hausa characters.
garken zaki sun kunshi maza manya daya zuwa uku masu dangantaka tare da mata da dama har zuwa talatin tare da 'ya'ya	garken zaki sun ƙunshi maza manya ɗaya zuwa uku masu dangantaka tare da mata da dama har zuwa talatin tare da yaya	No error	The only difference is the use of special Hausa characters.
dong ɗan kasar koriya ne	dung ɗan ƙasar koriya ne	No error	The only difference is the use of special Hausa characters.

Continued on next page

Table N.1: (continued) Human evaluation of ASR model errors in Hausa.

Evaluation Transcription	Model output	Evaluator assessment	Evaluator comment
manyan jami'ai ne kawai suka samu damar shiga wurin shugaban kasar	manyan jami'ai ne kawai suka samu damar shiga wurin shugaban ƙasar	No error	The only difference is the use of special Hausa characters.
dalibai sun yi wasan kwallo	ɗalibai sun yi wasan ƙwallo	No error	The only difference is the use of special Hausa characters.
dalibai sun kai ziyara gidan masu tabin hankali	ɗalibai sun kai ziyara gidan masu taɓin hankali	No error	The only difference is the use of special Hausa characters.
dakin karatun yana dauke da dalibai kusan dubu daya	ɗakin karatun yana ɗauke da ɗalibai kusan dubu ɗaya	No error	The only difference is the use of special Hausa characters.
sai karfe tara na dare za'a sanar da sakamakon zaɓen	sai ƙarfe tara na dare za a sanar da sakamakon zaɓen	No error	The only difference is the use of special Hausa characters.
za'a yi mata aiki a ƙwaƙwalwa	za a yi mata aiki a ƙwaƙwalwa	Error in evaluation transcript	Grammatically, the correct form is 'za a', not 'za'a'. However, many people are using 'za'a'.
ana shan magani idan ba'a da lafiya	ana shan magani idan ba a da lafiya	Error in evaluation transcript	Grammatically, the correct form is 'ba a', not 'a'ba'. However, many people are using 'ba'a'.
abuja na cikin nijeria	abuja na cikin najeriya	No error	Both 'Nijeriya' and 'Naijeriya' are used, so it depends on the newspaper or individual.
oguta karamin jiha ce a cikin nijeria	oguta ƙaramin jiha ce a cikin najeriya	No error	The only difference is the use of special Hausa characters.
sitika ɗin yana ɗa ƙyau	sitika ɗin yana da kyau	Error in evaluation transcript	The only difference is the use of special Hausa characters. And there are wrong use of the special characters e.g yana The correct version is yana.
yan shi'a sun yi tattaki jiya	yan shi'a sun yi tattaki jiya	No error	The only difference is the use of special Hausa characters.

Continued on next page

Table N.1: (continued) Human evaluation of ASR model errors in Hausa.

Evaluation Transcription	Model output	Evaluator assessment	Evaluator comment
macaroni abincin y'an italiya ne	makaroni abincin yan italiya ne	No error	The only difference is the use of special Hausa characters.
kula da alaƙa mai ƙarfi da y'an uwa	kula da alaƙa mai ƙarfi da yan'uwa	No error	The only difference is the use of special Hausa characters.
mallam aminu dan kasuwa ne à kasuwan kure	malam aminu ɗan kasuwa ne a kasuwan kure	Error in evaluation transcript	Error in the evaluation transcript. There is no à in Hausa wrting sytle that we physically see and read.

Table N.2: Human evaluation of ASR model errors in Dholuo.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comments
e piny kenya mano en ketho maduong' ahinya	e piny kenya mano en ketho maduong' ahinya	No error	Written Luo uses apostrophe at the final syllable as in the word maduong' but this does not result in a difference in meaning.
tuwo mar sukari ema ne onego owadawano	tuo mar sukari ema ne onego owadwano	No error	Excellent, in Luo we either say 'tuo' or 'tuwo'.
e wi mano tuwo mar corona ne oketho chenro mag somo e pinje	e wi mano tuo mar corona ne oketho chenro mag somo e pinje	No error	
otho sa adek okinyi	otho saa adek okinyi	No error	No error, we either use 'sa' or 'saa'.
neru maduong' osekendo	neru maduong' osekendo	No error	Only difference in apostrophe used.
seche moko ginyalo bedo jii ariyo ma penjo penjo	seche moko ginyalo bedo ji ariyo ma penjo penjo	No error	No error, it's either 'ji' or 'jii'. This a feature in Luo for monosyllabics.
dhii uywe kund dhok	dhi uywe kund dhok	No error	See comments above.
welo dhii e kanisa kawuono	welo dhi e kanisa kawuono	No error	See comments above.
ngama kare ber	ng'ama kare ber	No error	Native speakers know this and would understand.
unega kayiem nang'o	unega kayiem nang'o	No error	
pesa jadoung ile	pesa jadoung' ile	No error	Again optional final apostrophe.
ng'ama nigi jadoung machiegni	ng'ama nigi jadoung' machiegni	No error	
en chieng' maduong	en chieng' maduong'	No error	
antie gi othinyo mangeny	antie gi othinyo mang'eny	No error	
we bedo gi gombo mangeny	we bedo gi gombo mang'eny	No error	
mol mar trafik en timo nonro kata puonjruok kuom timbe mag joriembo kod mtokni e seche ma gisudo e kind kuonde ariyo to kod tudruoge magitimo e kindgi giwegi	mol mar trafik en timo nonro kata puonjruok kuom timbe mag joriembo kod mtokni e seche magisudo e kind kuonde ariyo to kod tudruoge magitimo e kindgi giwegi	No error	No error, 'u' is an alternative for 'wu'.

Continued on next page

Table N.2: (continued) Human evaluation of ASR model errors in Dholuo.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comment
onge wach achiel e piny duto ma lero tiend kaka mwandu molosi ichako luongi ni gima nyachon kembe moko mag solo osuru lero kuom mwandu mosteetieko higni maloyo 100 kaka gigo ma nyachon	onge wach achiel e piny duto malero tiend kaka mwandu molosi ichako luongi ni gima nyachon kembe moko mag solo osuru lero kuom mwandu mosteetieko higni maloyo 100 kaka gigo ma nyachon	No error	
jarieko manyinge aristolet nowacho ni gik moko olos kod riwo achiel ariyo kata ang'wen mag gigi piny pii muya kod mach	jarieko manyinge aristotle nowacho ni gik moko olos kod riwo achiel ariyo kata ang'wen mag gigi piny pii muya kod mach	No error	See my comment on monosyllabics.
atom moko nitiere kod nyukila ma ok ochung' motegno ma tiende ni gibarore ga ka otwomgi matin kata ka ok otwomgi chutho	atom moko ni tiyoore kod nyukila ma ok ochung' motegno matiende ni gibarore ga ka otwomgi matin kata ka ok otwomgi chutho	No error	

Table N.3: Human evaluation of ASR model errors in Chichewa.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comments
pa maulendo ena makampani ena akuluakulu ali ndi ndege zao koma pa maulendo ena makampani ang'onoang'ono amakhala ndi vuto	pamaulendo ena makampani ena akuluakulu ali ndi ndege zawo koma pamaulendo ena makampani ang'onoang'ono amakhala ndi vuto	Error	Model Output: grammar rules requires that pamaulendo written disjunctively as it is not a locative partical. Transcript: zao is gramatically wrong, should be written as zawo
ana okulira kwaokha osakumana ndi anthu akhoza kutheka kuchitilidwa nkhanzza kapena kuzunzidwa asanasiyeidwe kapena kuthawa	ana okulira kwaokha osakumana ndi anthu akhoza kutheka kuchitilidwa nkhanzza kapena kuzuzidwa asanasiyeidwe kapena kuthawa	Error	Model Output: kuzuzidwa is a spelling error, it should be kuzunzidwa.

Continued on next page

Table N.3: (continued) Human evaluation of ASR model errors in Chichewa.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comment
ma blog amathanidzaniso ana asukulu kuphunzila kulemba ngakhalke kuti poyamba ophunzila amayamba ndi kulakwista galamala ndi zilembo za mawu kupeza kwa anthu omweolega kumathandizila kusintha izi	ma blogamothandizaso ana asukulu kuphunzila kulemba ngakhalke kuti poyamba ophunzila amayamba ndi kulakwitsa galamala ndi zilembo zamawu kupeza kwa anthu owenerenga kumathandizila kusintha izi	Error	Model Output: blogamothandizaso is unknown word made from combination of two or three words. This would confuse the reader. Amayambamba is a wrong spelling, it should be as in Transcription: amayamba.
ngoziyi inachitikira mwamba m'mapiri atali ndipo akukhulupilira kuti zinachitika chifukwa cha adani achiwembu	ngoziyi inachitikira m'mwamba m'mapiri atali ndipo akukhulupilira kuti zinachitika chifukwa cha adani achiwembu	Error	Transcription: atali should be aatali as in model output. a chibwembu is normally written conjunctively and should be achibwembu as in Model Output.
ku barcelona chiyankhulo chovomelezeka ndi catalan ndikli sipanishi theka la anthu akudziwa catalan ambiri amachimvetsa ndiipo pafupifupi onse amamva ndikudziwsa chi sipanishi	ku barcelona chiyankhulo chovomelezeka ndi catalan ndi chi sipanishi theka la anthu akudziwa catalan ambiri amachimvetsa ndiipo pafupifupi onse amamva ndikudziwsa chi sipanishi	Error	Transcription: chi should not be combined with chi but rather goes together with the name of the language to read: chisapnishi. Chi and sipanishi should be written conjunctively. Model Output: Chi and sipanishi should be written conjunctively.
zolegeza zathawi zonse mu metro zimapangidwa muchilankhulo chachi kalatani basi koma zosiyanasiyana zimasulutidwa kudzera makina a kompyuta mu zilankhulo zosiyanasiyanansiya kuphatikizikachisapanishi chingerezi falansa arabic ndi japanese	zolegeza za nthawi zonse mu metro zimapangidwa muchilankhulo chachi kalatani basi koma zosiyanasiyana zimasulutidwa kudzera makina a kompyuta mu zilankhulo zosiyanasiyana kuphatikizika chisipanishi chingerezi falansa arabic ndi japanese	Error	Transcription: chi should not be combined with chi but rather goes together with the name of the language to read: chisapnishi. Kompyuta should be kompyuta. chisipanishi chingerezi falansa arabic ndi japanese should have commas in between.
owona zangozi zokugwa madziidzi ku mpoto kwa mariana ati palibe zomve zidanowoneka pomwe analengezera a nation	owona zangozi zokugwa mwaziizizi kumpoto kwa mariana ati palibe zomve zidanowoneka pomwe analengezera a nation	Error	Transcription: madziidzi is an incomplete word which should read mwadziidzi. Model Output: Mwazizizi is gramatically incorrect.

Continued on next page

Table N.3: (continued) Human evaluation of ASR model errors in Chichewa.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comment
apia ndi likulu la samoa tauinyi ili pachilumba cha upolu ndipo pali chilengedwe chachilengedwe cha anthu ochepera pa 40000	apia ndi likulu la samoa tauinyi ili pachilumba cha upolu ndipo pali chilengedwe chachilengedwe cha anthu ochepera pa 4000	Error	Model Output: City of a pia should be Apia and the a should not be separated from pia.
safari ndi mawu amene amatchulidwa kawirikawiri ndiipo amatanthauza ulendo wopamtunda wokaona nyama zokongola za-kutchire za ku africa kawirikawiri ku savanna	safari ndi mawu amene amatchulidwa kawirikawiri ndiipo amatanthauza ulendo wopamtunda wokaona nyama zokongola za-kutchire zaku africa kawirikawiri ku savanna	Error	Model Output: zaku should be written separately as za ku.
kutenga sitima za m'madzi kunyamulira katundu ndi njira yoyenera zedi yonyamulira anthu wochuluka komanso katundu kuwoloka pa nyanja	kutenga sitima za m'madzi kunyamulira katundu ndi njira yoyenera zedi yonyamulira anthu ochuluka komanso katundu kooloka panyanja	Error	Model Output: kooloka is wrong spelling of kuwoloka
mkulu wophunzitsa pa sukulu ya ukachenjede ya dundee university a pulofesa pamela ferguson adati atolankhani akuoneka kuti akuyenda mu chiwopsezo akamasindikiza zithunzi ndi zina zotero za oganizilidwa kupalamula milandu	mgalu wophunzitsa pasukulu yaukachenjede ya dud university a pulofesa pamella fegason adati atolankhani akuoneka kuti akuyenda mu chiopsezo akamasindikiza zithuzi ndi zina zotero za oganizilidwa kupalamula milandu	Error	Model Output: mgulu is wrong spelling of mkulu. Yaukachenjede should be written as ya ukachenjede
thandizo loiyikidwa m'maphunziro apa kompyuta ndipo akuyenera kufunsa kupanga zina zakefotokozera ndondomeko zomwe zikanakhala zovuta kwa ophunzira	thandizo loiyikidwa m'maphunziro apakompyuta ndipo akuyenera kufunsa kupanga zina zake ndikufotokozera ndunudmiko zomwe zikanakhala zovuta kwa ophunzira	Error	Model Output: ndunudmiko is wrong spelling of ndondomeko and kompyuta should read kompyuta.

Continued on next page

Table N.3: (continued) Human evaluation of ASR model errors in Chichewa.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comment
bomba la fission limagwira ntchito pamene limafuna mphamvu kuti liyike pamodzi ma nucleus wochulukana ndi ma proton ambiri ndi ma neutron	bomba la fission limagwira ntchito pamene limafuna mphamvu kuti liyike pamodzi ma nucleus wochulukana ndi ma proton ambiri ndi ma neutron	No error	
ma ion ndima proton a hydrogen amathotholedwa popeza hydrogen amakhala ndi proton imodzi ndi electron imodzi	ma ion ndi ma proton a hydrogen amathotholedwa popeza hydrogen amakhala ndi proton imodzi ndi electron imodzi	Error	Model Output: amathotoleledwa is wrong spelling of amathotholedwa
wakhalapo akulepherera kumwa mankhwala ofunikira pochiza ululu omwe akumva chifukwa cha matenda alowedwa pa masewera	wakhalapo akulepherera kumwa mankhwala ofunikira pochiza ululu omwe akumva chifukwa cha matenda olestedwa pamasewera	No error	Transcription: Woyima should be oyima. Model Output: Kuwumikizana is wrong spelling of kulumikizana. Maro is wrong spelling of Malo.
zilumba zambiri zing'onoang'ono ndi mayiko woyima powapita kulumikizana ndi dziko la france ndi ziko la arabic ndi japanese	zilumba zambiri zing'onoang'ono ndi mayiko woyima powapita kulumikizana ndi dziko la france ndi ziko la arabic ndi japanese	Error	Transcription: Woyima should be oyima. Model Output: Kuwumikizana is wrong spelling of kulumikizana. Maro is wrong spelling of Malo.
kuwonetsera kwa nyumba zomwe zimapangidwa mawonedweko a hong kong skyline akutchulidwa victoria harpur bar tatchi yowala kwambiri m'madera oyandikira doko chikwangwanzi akamapereka zikwangwanzi m'dera lozungulira zonyamula anthu omwe amafika mochuluka	kuwonetsera kwa nyumba zomwe zimapangidwa mawonedweko a hongkong skyline akutchulidwa victoria harpur bar tatchi yowala kwambiri m'madera oyandikira doko chikwangwanzi akamapereka zikwangwanzi mdera lozungulira zonyamula anthu omwe amafika mochuluka	Error	Model Output: victoria harpur is misspelling of victoria harbour.

Continued on next page

Table N.3: (continued) Human evaluation of ASR model errors in Chichewa.

Evaluation transcript	Model output	Evaluator assessment	Evaluator comment
kujambula makamera amachita kusiyana atakhala pa kompyuta pakumasulira kwa mwachangu amene ang'ono chinthu chinasache kupeza kwa linako kapena wotsekereza monga ndondomeko yomwe ingathe kupangidwa ndi anthu ochepa	kujambula makamera amachita kusiyana atakhala pa kompyuta pakumasulira kwa mwachangu amene ang'ono chinthu chinasache kupeza kwa linako kapena wotsekereza monga ndondomeko yomwe ingathe kupangidwa ndi anthu ochepa	Error	Transcription: Milisecond should be transliterated to Milisekondi. Model Output: miliceand is a wrong spelling of millisecond which is normally transliterated as milisecond.
pamene mafumu ndi akuluakulu mabanja ndi zochitika mufunikanso kuti mufikeko nsanga ngati kuli msanga apafupi ndi nyumba	pamene mafumu ndi akuluakulu mabanja ndi zochitika mufunikanso kuti mufikeko nsanga ngati kuli msanga apafupi ndi nyumba	Error	Model Output: mufikeko is wrong spelling of mufikeko.
pakuti mu nthawi yao kuonjeza kuwala kwa sikunali wuto monga anali pa makolo m'pang'ono amafunika kuwala koopsa kufikila kusiyana ndi omwe amaganidwa makono ano	pakuti mu nthawi yao kuonjeza kuwala kwa sikunali wuto monga anali pa makolo m'pang'ono amafunika kuwala koopsa kufikila kusiyana ndi omwe amaganidwa makono ano	Error	Model Output: panu is wrong spelling of pano. Campus should be transliterated to kampasi or just describe what a campus is.

Appendix O Wav2Vec-BERT 2.0 Hyperparameters

Hyperparameter	Value
Learning rate	3e-05
Warmup ratio	0.1
Evaluation steps	1000
Early stopping patience	5
Add adapter	True
Mask time probability	0
Attention dropout	0.05
Feature projection dropout	0.05
Hidden layer dropout	0.05
CTC zero infinity	True

Table O.1: Common Wav2Vec-BERT 2.0 hyperparameters.

Real:Synth Ratio	(Maximum) Number of epochs	(Total) Batch size
100h:0	250	320
250h:0	100	320
500h:0	50	320
100h:400h	50	320
250h:250h	50	320
579h:450h	24	320
579h:993h	16	320

Table O.2: Hausa Wav2Vec-BERT 2.0 Hyperparameters. We keep epoch-hours constant, that is the number of epochs multiplied by the total duration of the training dataset in hours.

(Maximum) Number of steps	(Total) Batch size
100000	64

Table O.3: Dholuo and Chichewa Wav2Vec-BERT 2.0 hyperparameters.

ADOR: Dataset for Arabic Dialects in Hotel Reviews: A Human Benchmark for Sentiment Analysis

Maram Alharbi[1,2], **Saad Ezzini**[3], **Tharindu Ranasinghe**[1]
Hansi Hettiarachchi[1] **and Ruslan Mitkov**[1]
[1]School of Computing and Communications, Lancaster University, UK
[2]Jazan University, Saudi Arabia
[3]King Fahd University of Petroleum and Minerals, Saudi Arabia
m.i.alharbi@lancaster.ac.uk

Abstract

Arabic machine translation remains a fundamentally challenging task, primarily due to the lack of comprehensive annotated resources. This study evaluates the performance of Meta's NLLB-200 model in translating Modern Standard Arabic (MSA) into three regional dialects: Saudi, Maghribi, and Egyptian Arabic using a manually curated dataset of hotel reviews. We applied a multi-criteria human annotation framework to assess translation correctness, dialect accuracy, and sentiment and aspect preservation. Our analysis reveals significant variation in translation quality across dialects. While sentiment and aspect preservation were generally high, dialect accuracy and overall translation fidelity were inconsistent. For Saudi Arabic, over 95% of translations required human correction, highlighting systemic issues. Maghribi outputs demonstrated better dialectal authenticity, while Egyptian translations achieved the highest reliability with the lowest correction rate and fewest multi-criteria failures. These results underscore the limitations of current multilingual models in handling informal Arabic varieties and highlight the importance of dialect-sensitive evaluation.

1 Introduction

Arabic is spoken by hundreds of millions across more than twenty countries, yet it remains significantly underrepresented in natural language processing (NLP) (Darwish et al., 2021; Premasiri et al., 2022). This is particularly acute for Arabic dialects, which diverge from Modern Standard Arabic (MSA) in terms of morphology, syntax, phonology, and vocabulary (Shoufan and Alameri, 2015). Dialects lack orthographic standardisation, exhibit wide regional variation, and are primarily used in informal and spoken contexts (El-Haj et al., 2024). As a result, NLP systems trained predominantly on MSA often perform poorly on dialectal data, limiting their effectiveness in real-world applications (Almansor and Al-Ani, 2017).

Machine translation (MT) of Arabic reflects these challenges. While MSA serves as the formal written standard, dialects are the primary medium of everyday communication across the Arab world. Their structural and lexical variation, combined with the absence of standardised norms, complicates the development of MT systems capable of handling the full spectrum of Arabic varieties (Zouidine and Khalil, 2025). Recent advancements in multilingual MT, such as Meta's NLLB-200 model, which incorporates FLORES-200 language codes, have extended support to low-resource languages, including Arabic dialects (Costa-jussà et al., 2022). Building on this, our study evaluates NLLB-200's performance in the reverse translation direction: from MSA into three major dialects; Saudi, Maghribi, and Egyptian. We introduce ADOR, (**A**rabic **D**ialects for H**o**tel **R**eviews) manually annotated dataset. ADOR assesses translation quality across four key dimensions: semantic correctness, dialect authenticity, sentiment preservation, and aspect category alignment. Using a structured human annotation protocol and error taxonomy, we offer both quantitative and qualitative insights into the capabilities and limitations of current MT systems.

The remainder of this paper is structured as follows: Section 2 reviews related work; Section 3 outlines the NLLB-200 architecture; Section 4 describes the dataset and preprocessing steps; Section 5 details the annotation framework; Section 6 presents the evaluation results; and Section 7 concludes with directions for future work.

2 Related Work

Meta's NLLB-200 model (Costa-jussà et al., 2022) supports over 200 languages and includes Arabic

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 187–191, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-017

dialects via FLORES-200 codes. However, its performance in dialectal settings remains limited. Atwany et al. (2024) evaluated NLLB-200 on dialect-to-MSA translation for divergent varieties including Gulf, Egyptian, Levantine, Iraqi, and Maghrebi, highlighting the inaccuracy of treating dialects as standard source languages due to their complexity.

Mousi et al. (2025), while not focused on translation, benchmarked performance across dialects and highlighted substantial disparities, reinforcing the need for dialect-aware evaluation protocols. A complementary perspective is offered by Yakhni and Chehab (2025), who studied Lebanese dialect-to-English translation. Their findings showed that NLLB ability to preserve cultural nuance in informal, idiomatic content is limited.

Finally, Boughorbel et al. (2024) addressed English-to-Arabic translation by translating the TinyStories dataset using NLLB-3B. They found that relying solely on MT introduced linguistic and cultural noise, and showed that further pre-training on a small corpus of native-generated Arabic stories improved output quality.

Collectively, these studies demonstrate that existing MT systems struggle with dialectal Arabic across multiple translation directions. They also emphasise the need for native speaker evaluation, cultural grounding, and dialect-specific benchmarks. Our work builds on these insights by evaluating NLLB-200 translations from MSA into three arabic dialects using a structured human-annotated framework.

3 Meta's No Language Left Behind (NLLB-200)

The No Language Left Behind (NLLB-200) model (Costa-jussà et al., 2022), developed by Meta, is a multilingual encoder–decoder transformer architecture designed to improve machine translation (MT) for low-resource languages. It supports direct translation between over 200 languages using FLORES-200 language codes, including several Arabic dialects.

NLLB-200 uses a unified encoder to process source text and a decoder that generates output conditioned on the target language or dialect. In this study, we use the model to translate from MSA into three Arabic dialects: Saudi, Maghribi and Egyptian. These dialects are explicitly supported within the model's language inventory, enabling direct generation without intermediate normalisation to MSA.

4 Data

This study uses the Arabic hotel reviews dataset from SemEval 2016 Task 5 on Aspect-Based Sentiment Analysis (ABSA) (Pontiki et al., 2016). The original dataset comprises over 10,000 sentences written in Modern Standard Arabic (MSA), each annotated with one or more aspect terms, sentiment polarities (positive, negative, neutral), and aspect categories. To ensure consistency and relevance for downstream translation and manual annotation, the dataset underwent several preprocessing steps.

Sentence Deduplication Duplicate entries were removed by grouping reviews with identical sentence text. For each unique sentence, all associated sentiment, aspect target, and category annotations were aggregated to retain the full range of opinions tied to that sentence.

Text Cleaning The text was normalised using the Ruqya library by removing special characters, hashtags, and diacritics (tashkīl). This step ensured uniform formatting and reduced noise, making the data more suitable for input into translation models.

Length Filtering Sentences with fewer than six words were excluded, as they typically lacked the contextual richness necessary for meaningful translation and sentiment analysis.

Polarity Consolidation Since a single sentence could be annotated with multiple aspect-level polarities, a rule-based approach was applied to assign one consolidated sentiment label:

- Sentences containing both positive and negative polarities were labelled as neutral.
- If a neutral polarity co-occurred with either positive or negative, the non-neutral polarity was retained.
- If only one polarity was present, it was used as the sentence-level label.

All consolidated sentiment labels were then manually reviewed to ensure correctness and internal consistency.

Following these preprocessing steps, the resulting dataset consisted of 538 sentences, balanced across sentiment classes: 200 positive, 200 negative, and 138 neutral.

5 Annotation Framework

Following the dialectal translation process, each sentence was manually evaluated by a native speaker of the corresponding dialect. The purpose of the annotation task was to assess the quality of the machine-generated translations across multiple linguistic and semantic dimensions.

Each annotator received a structured annotation template in CSV format containing the original MSA sentence, the machine-translated dialectal sentence, the sentiment label, and the associated aspect categories. One native speaker was assigned per dialect to ensure linguistic authenticity. Annotations were conducted independently for each dialect.

Annotators were instructed to assess each translation according to six criteria:

1. *Translation Correctness*: Does the translation accurately convey the meaning of the original sentence?

2. *Dialect Accuracy*: Is the sentence rendered in the appropriate dialect?

3. *Sentiment Preservation*: Is the original sentiment polarity maintained in the translation?

4. *Target Preservation*: Is the aspect or subject of the sentence correctly preserved?

5. *Corrected Sentence*: If needed, a revised version of the translated sentence.

6. *Target Correctness*: A corrected version of the aspect/target if it was omitted, distorted, or unclear.

The first four criteria were assessed using binary labels (Yes/No) to reduce subjectivity and enforce consistency. The final two were free-text fields used only when corrections were necessary.

All annotators received a detailed guideline document outlining each evaluation criterion. For clarity, definitions were provided alongside examples of both accurate and flawed translations, helping annotators distinguish between acceptable variation and critical errors. The guidelines also included instructions for identifying mismatches, particularly in sentiment and aspect categories, emphasising the importance of accurately preserving key content from the original MSA sentence. Annotators were familiarised with common translation error patterns, such as literal translations of idioms, the use of formal MSA constructions in dialectal output, and inappropriate lexical choices. In such cases, they were expected to revise translations to align with dialect norms while maintaining the intended meaning. Finally, procedures were outlined for handling ambiguous or incomplete translations, encouraging annotators to flag unclear outputs and consult the MSA source sentence before making corrections. Regular check-ins ensured consistency across dialects and allowed for immediate resolution of ambiguities. Upon completion, all annotation files were manually reviewed.

Although each dialect was annotated by a single native speaker, the structured process, guided instructions, and direct oversight helped ensure a high level of annotation reliability.

6 Evaluation Results and Discussion

The evaluation is based on a structured annotation framework designed to assess semantic correctness, dialectal fidelity, and sentiment preservation. It provides both quantitative metrics and qualitative insights into the limitations of NLLB-200 in translating MSA into regional Arabic dialects.

6.1 Overview of Annotation Outcomes

Table 1 summarises annotator judgments across four binary evaluation criteria. While all three dialects show high sentiment and aspect category preservation scores, there is a clear disparity in translation correctness and dialect accuracy. Egyptian translations were rated highest in both criteria, 84.7% correctness and 76.6% dialect accuracy, indicating stronger adaptation by NLLB-200 for Egyptian Arabic. Maghribi follows with 77.6% correctness and 44.4% accuracy, while Saudi trails with the lowest scores. These results suggest that NLLB-200 is more effective at generating fluent and regionally appropriate outputs for Maghribi than for Saudi Arabic.

6.2 Sentiment Agreement

Given that each sentence in the dataset was pre-annotated with sentiment labels prior to translation, the Sentiment Preservation score can be interpreted as a measure of annotator agreement. Specifically, a "Yes" label indicates that the human reviewer agreed that the sentiment expressed in the translated sentence matches that of the original MSA input. Agreement rates were high for all the three

Criterion	Saudi		Maghribi		Egyptian	
	Yes (%)	No (%)	Yes (%)	No (%)	Yes (%)	No (%)
Translation Correctness	68.4	31.6	77.6	22.4	84.7	15.3
Dialect Accuracy	5.2	94.8	44.4	55.6	76.6	23.4
Sentiment Preservation	98.9	1.1	90.9	9.1	94.6	5.4
Target Preservation	91.4	8.6	90.5	9.5	92.3	7.7

Table 1: Binary annotation outcomes across key criteria.Values represent the proportion of "Yes" and "No" judgments for each dialect.

Dialect	Avg Criteria Score	Correction Rate (%)	$\geq$ 2 Failures (%)	$\geq$ 3 Failures (%)
Saudi	0.684	95.17	26.21	5.02
Maghribi	0.776	55.58	18.22	11.34
Egyptian	0.867	31.04	13.01	2.97

Table 2: Summary of translation evaluation statistics across dialects. Criteria score reflects average binary ratings for translation correctness, dialect accuracy, sentiment preservation, and target preservation.

dialects; 98.88% for Saudi, 94.60 for Egyptian and 90.71% for Maghribi. That affirm the reliability of the original sentiment labels and the annotators' consistency, lending additional credibility to the overall annotation process.

6.3 Error Distribution and Correction Analysis

Table 2 presents aggregated error metrics, including average criteria scores, correction rates, and the frequency of compound evaluation failures. Among the three dialects, Egyptian outputs required the fewest corrections (31.04%) and exhibited the lowest rate of multi-criteria failures, indicating greater reliability and better alignment with dialectal norms. In contrast, Saudi Arabic translations had a significantly higher correction rate (95.17%) compared to Maghribi (55.58%), reinforcing earlier observations that Saudi outputs demanded more extensive post-editing. This finding is consistent with the low dialect accuracy score and highlights systemic challenges in the model's ability to generate fluent and authentic Saudi vernacular.

Although Maghribi translations were more accurate on average, they exhibited a slightly higher rate of cases with three or more simultaneous evaluation failures (11.34%), which indicate that, despite closer alignment with dialect norms, certain Maghribi translations required broader structural revisions to address subtler fluency or coherence issues.

6.4 Interpretation

These results collectively illustrate the need for human-centered, dialect-sensitive evaluation frameworks in Arabic MT. While NLLB-200 demonstrates promising performance on some dimensions, it struggles with dialectal fluency and semantic fidelity.

The data also highlights the importance of manual correction and targeted annotation, as many outputs superficially appear fluent but fail under semantic or dialectal review. These insights underscore the limitations of automatic metrics in low-resource dialect contexts and support the value of qualitative human validation as part of the evaluation process.

7 Conclusion and Future Work

This study presented a structured evaluation of NLLB-200's ability to translate MSA into three major Arabic dialects: Saudi, Maghribi, and Egyptian. Using **ADOR**, a manually annotated benchmark grounded in linguistic and semantic criteria, we identified systematic translation errors and highlighted performance variability across dialects. The Findings revealed that while NLLB-200 achieves high rates of sentiment and aspect preservation, its performance on dialect accuracy and translation correctness remains inconsistent. Saudi Arabic translations exhibited a high dependency on human correction, pointing to the model's difficulty in handling dialects that are lexically and syntactically distant from MSA. In contrast, Maghribi translations demonstrated better dialect fidelity and required fewer revisions. Notably, Egyptian outputs achieved the highest overall reliability, with the lowest correction rate and the fewest multi-criteria failures, suggesting stronger alignment between NLLB-200 output and Egyptian dialect norms.

Building on this work, future efforts will expand the dialectal coverage of the dataset to include ad-

ditional varieties such as Levantine and Yemeni dialects. To enhance annotation reliability, multiple annotators will be recruited per dialect, enabling inter-annotator agreement analysis and reducing subjectivity in evaluation.

References

Ebtesam H. Almansor and Ahmed Al-Ani. 2017. Translating dialectal arabic as low resource language using word embedding. In *International Conference Recent Advances in Natural Language Processing, RANLP*, volume 2017-September, pages 52–57. Incoma Ltd.

Hanin Atwany, Nour Rabih, Ibrahim Mohammed, Abdul Waheed, and Bhiksha Raj. 2024. OSACT 2024 task 2: Arabic dialect to MSA translation. In *Proceedings of the 6th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT) with Shared Tasks on Arabic LLMs Hallucination and Dialect to MSA Machine Translation @ LREC-COLING 2024*, pages 98–103, Torino, Italia. ELRA and ICCL.

Sabri Boughorbel, MD Rizwan Parvez, and Majd Hawasly. 2024. Improving language models trained on translated data with continual pre-training and dictionary learning analysis.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. `https://ai.facebook.com/research/no-language-left-behind/`. Meta AI.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the arab world. *Commun. ACM*, 64(4):72–81.

Mo El-Haj, Sultan Almujaiwel, Damith Premasiri, Tharindu Ranasinghe, and Ruslan Mitkov. 2024. DARES: Dataset for Arabic readability estimation of school materials. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 103–113, Torino, Italia. ELRA and ICCL.

Basel Mousi, Nadir Durrani, Fatema Ahmad, Md. Arid Hasan, Maram Hasanain, Tameem Kabbani, Fahim Dalvi, Shammur Absar Chowdhury, and Firoj Alam. 2025. AraDiCE: Benchmarks for dialectal and cultural capabilities in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4186–4218, Abu Dhabi, UAE. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Damith Premasiri, Tharindu Ranasinghe, Wajdi Zaghouani, and Ruslan Mitkov. 2022. DTW at qur'an QA 2022: Utilising transfer learning with transformers for question answering in a low-resource domain. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 88–95, Marseille, France. European Language Resources Association.

Abdulhadi Shoufan and Sumaya Alameri. 2015. Natural language processing for dialectical Arabic: A survey. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 36–48, Beijing, China. Association for Computational Linguistics.

Silvana Yakhni and Ali Chehab. 2025. Can LLMs translate cultural nuance in dialects? a case study on Lebanese Arabic. In *Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script*, pages 114–135, Abu Dhabi, UAE. Association for Computational Linguistics.

Mohamed Zouidine and Mohammed Khalil. 2025. Large language models for arabic sentiment analysis and machine translation. *Engineering, Technology and Applied Science Research*, 15:20737–20742.

Towards Creating a Bulgarian Readability Index

Dimitar Kazakov
University of York
dlk2@york.ac.uk

Stefan Minkov
GATE Institute
stefanminkov2003@gmail.com

Ruslana Margova
GATE Institute
ruslana.margova@gate-ai.eu

Irina Temnikova
GATE Institute
irina.temnikova@gate-ai.eu

Ivo Emanuilov
GATE Institute
ivo.emanuilov@gate-ai.eu

Abstract

Readability assessment plays a crucial role in education and text accessibility. While numerous indices exist for English and have been extended to Romance and Slavic languages, Bulgarian remains underserved in this regard. This paper reviews established readability metrics across these language families, examining their underlying features and modelling methods. We then report the first attempt to develop a readability index for Bulgarian, using end-of-school-year assessment questions and literary works targeted at children of various ages. Key linguistic attributes, namely, word length, sentence length, syllable count, and information content (based on word frequency), were extracted, and their first two statistical moments, mean and variance, were modelled against grade levels using linear and polynomial regression. Results suggest that polynomial models outperform linear ones by capturing non-linear relationships between textual features and perceived difficulty, but may be harder to interpret. This work provides an initial framework for building a reliable readability measure for Bulgarian, with applications in educational text design, adaptive learning, and corpus annotation.

1 Introduction

The importance of text comprehensibility is undeniable and even crucial in cases, such as the medical domain and emergency situations (Temnikova et al., 2015; Friedman et al., 2008). The most straightforward way to estimate how comprehensible a text is is to understand how easy it is to read by a target group of readers. The linguistic characteristics that make a text easier or harder to read are referred to as *readability*. Its significance is shown by the fact a substantial number of languages have seen work on quantifying readability and improving it through measures, such as manual or automatic text simplification (Alfear et al., 2024; Al-Thanyyan and Azmi, 2021; Siddharthan, 2014; Saggion and Hirst, 2017).

A text readability index is a language-specific tool that requires appropriate resources for its creation. Bulgarian NLP has a half century-long tradition yet in this aspect it ranks among the lower-resourced languages, as it lacks such an index or the specific age-appropriate text corpora that would support its creation. This motivates our work, which represents the first steps towards building a readability index for Bulgarian.

2 Background

Readability is defined as the effect of all elements that make a text more or less comprehensible to a group of readers. Some scholars consider that readability is also linked to how interesting the text is (Dale and Chall, 1949; DuBay, 2004; Klare, 1963; McLaughlin, 1969; Hargis et al., 1998). Text complexity can be affected on multiple levels, from morphology to pragmatics, some of which are hard to evaluate automatically. Most frequently, readability is estimated through a combination of surface linguistic features, such as the average length of words in characters or syllables, the average sentence length in words, the words' difficulty estimated by their frequency in large corpora or their mere presence in a corpus of a certain size. These features are then typically used as attributes of some type of regression, where the predictor aims to approximate the quantified reading level of the training texts, and the resulting

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 192–200, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-018

formula is evaluated in terms of goodness of fit, through the coefficient of determination r^2 on unseen data. The regression equation, which contains simple arithmetic operations and language-specific numerical parameters, is then used as "readability index". These indices were originally created to find reading material that matched the reading abilities of students of a certain grade or age.

These traditional readability indices have been criticised for their somewhat simplistic approach, which does not take into account factors at the syntactic, semantic, and pragmatic levels, the logical order of ideas, etc. Some of these shortcomings have been addressed in recent readability tools and resources.

One such example is the English-language Medical Research Council (MRC) Psycholinguistic Database (Coltheart, 1981) and Coh-Metrix (Graesser et al., 2004). MRC is a lexical database containing values for several psychological features for more than 150,000 English words. The features include familiarity, age of acquisition, concreteness, word length, etc. Coh-Metrix is an automatic text analysis tool that detects deeper text complexity and comprehensibility features, such as cohesion, word frequency, concreteness, familiarity, and sentence structure complexity.

Coh-Metrix has also been adapted for Spanish (Quispesaravia et al., 2016) and Brazilian Portuguese (Scarton and Aluísio, 2010). A similarly complex tool, called ErreXail, was created for Basque (Gonzalez-Dios et al., 2014). Machine Learning (ML) models have also begun being used to estimate text readability, making use of more complex representations, such as embeddings. Such models have been created for several languages including English, Spanish, Basque, French, Catalan, Italian, French, and Slovene, variously based on regression, classifiers, random forests, neural networks, or transformers (Vajjala and Meurers, 2012; Vajjala and Lučić, 2018; Madrazo Azpiazu and Pera, 2021; Martinc et al., 2019).

2.1 English Readability Indices

While readability indices have their limitations, they constitute the first step towards estimating text comprehensibility. Unsurprisingly, the language best supported with readability resources is English, and its readability indices are often adapted to other languages by obtaining a new set of language-specific numerical parameters. Here are some of the best known readability indices for English:

- **Flesch Reading Ease** (Flesch, 1948)

  $$\text{FRE} = 206.835 - 1.015\frac{\#\text{words}}{\#\text{sentences}} - 84.6\frac{\#\text{syllables}}{\#\text{words}}$$

- **Flesch-Kincaid Grade Level** (Kincaid et al., 1975), which outputs a U.S. school grade level.

  $$\text{FKGL} = 0.39\frac{\#\text{words}}{\#\text{sentences}} + 11.8\frac{\#\text{syllables}}{\#\text{words}} - 15.59$$

- **Gunning Fog Index** (Gunning, 1952)

  $$\text{Fog Index} = 0.4\frac{\#\text{words}}{\#\text{sentences}} + 100\frac{\#\text{complex words}}{\#\text{words}}$$

- SMOG Index (McLaughlin, 1969), designed for health literacy texts.

  $$\text{SMOG Grade} = 1.0430\sqrt{\#\text{polysyllabic words}\frac{30}{\#\text{sentences}}} + 3.1291$$

- **Coleman-Liau Index** (Coleman and Liau, 1975)

  $$\text{CLI} = 0.0588L - 0.296S - 15.8$$

  where L = average letters per 100 words and S = average sentences per 100 words

- **Automated Readability Index (ARI)** (Senter and Smith, 1967), which also outputs a U.S. grade level.

  $$\text{ARI} = 4.71\frac{\#\text{characters}}{\#\text{words}} + 0.5\frac{\#\text{words}}{\#\text{sentences}} - 21.43$$

2.2 Readability Beyond English

All features used above are easy to compute, which has led to efforts to adopt these indices for many other languages. Some of them are shown in Table 1. There are also readability formulae for several Slavic languages, which are mostly adapted from English (see Table 2).

To the best of our knowledge, there is no Bulgarian readability index. Both research and practical solutions on that topic remain limited. In fact, the Bulgarian official educational regulations do not mention readability.[1] There is not even a universally accepted Bulgarian term for this concept. Within the limited body of existing literature in Bulgarian on

[1] https://www.mon.bg/nfs/2018/01/naredba_6_11.08.2016_bg_ezik.pdf

the subject, readability is variously referred to as "четимост" (Sharkova and Garov, 2015) and "четивност" (Borisova, 2017), despite certain authors arguing in favour of the latter term (Angelova, 2018).

References to readability indices (RI) in Bulgarian publications are rare and typically pertain to educational curricula up to the fourth grade. In such cases, the methodologies mentioned are often based on adaptations of indices originally developed for the Russian language (Yocheva, 2017).[2]

There has beeen research to create primary education texts in Bulgarian annotated with reading difficulty. This resource was created by translating Italian children's texts into Bulgarian, calculating several of their readability characteristics, and correlating them with finger-tracking results from 73 Bulgarian children (Pirelli and Koeva, 2024; Lento et al., 2024; Koeva et al., 2023). However, the aim of this research was never to create a Bulgarian readability index, and the use of translated texts is a limitation of the corpus. This leaves our article as the first to present efforts towards creating a readability formula for Bulgarian.

3 Data

The initial dataset consisted of a collection of 68 texts of national external assessment exams for grades 4, 7 and 10,[3] as well as the end of grade 12 Bulgarian matriculation exam.[4] The texts are published on the website of the Ministry of Science and Education, and we have only used those parts that test language comprehension for our purposes. The texts used in the matriculation exams are a balanced, 50:50 sample from Bulgarian modern classics and journalistic publications. The final dataset also incorporates 49 excerpts of fiction books in Bulgarian listed as recommended reading for grades 1–12. For each grade, several excerpts of approximately 1000 words have been selected: 6 or 7 for grades 1–4, and 3 for grades 5–12.

[2]The links to these adapted formulas are currently inaccessible for analysis due to restrictions on access to Russian websites.

[3]https://www.mon.bg/obshto-obrazovanie/natsionalno-vanshno-otsenyavane-nvo/

[4]https://www.mon.bg/obshto-obrazovanie/darzhavni-zrelostni-izpiti-dzi/izpitni-materiali-za-dzi-po-godini/

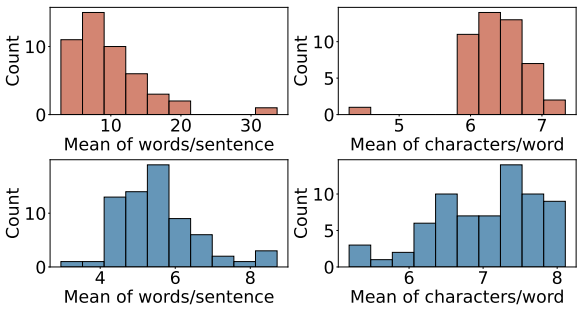


Figure 1: Descriptive statistics of data: Assessment texts (top), Literary prose (bottom)

The texts included both Bulgarian originals and translations. The year of publication was chosen to be around the middle of the 20th century as the best trade-off between representing Bulgarian as currently spoken and the lack of copyright. Note there is one mediaeval text adapted to modern Bulgarian. The texts are only in prose – due to the chosen focus of the task poetry was not included. Figure 1 contains descriptive statistics for each of the two parts of the corpus.

4 Methodology

For the purposes of this study, a number of choices need to be made and the outcomes compared. To begin with, one needs to find a suitable and representative dataset of texts corresponding to various levels of readability. Ideally, each text will have its readability level assigned on an ordinal or absolute (numerical) scale. With such explicit annotation, one can train an ordinal, linear, etc. regression model expressing the readability level as a function of a salient set of features defined over the text, typically relating to such statistics as word and sentence length or word frequency.

Optional preprocessing steps, such as stemming, mapping words to lexemes, removing words from the closed lexicon (also known as stop-words) may be carried out. After that, the chosen statistics $x_1, \ldots, x_n$ are calculated for each text and a regression model is fitted in order to express the readability score as a function of these statistics.

A linear model $\hat{y} = f(\mathbf{x})$ is the obvious first stop in the search for the best model due to its simplicity and interpretability, e.g. *the longer the average length of the words and the sentences, the less readable the text.*

Language	Index	Predicted	Features
German	Amstad (Amstad, 1978)	Reading-ease score (higher = easier)	Average sentence length (ASL); Average word length in syllables (AWL)
German	Wiener Sachtextformel (Bamberger and Vanecek, 1984)	age or education level	Proportion of words with three or more syllables; ASL; Proportion of words with six or more characters; Proportion of one-syllable words
Swedish, Danish, Norwegian	LIX (Lesbarhetsindex) (Björnsson, 1968)		ASL; Percentage of long words (>6 letters)
Dutch	Brouwer Leesindex (Brouwer, 1963)	ideal Dutch proficiency	ASL; AWL
French	Flesch Douma (Douma, 1960)		ASL; AWL
Romanian	Dascălu's adaptation (Dascălu et al., 2015)		Sentence/syllable/word length; Lexical complexity, parts of speech
Japanese	JARI (Fujita et al., 2012)		kana/kanji counts, word/sentence length, character/word complexity
Chinese	CRIE (Chinese Readability Index Explorer) (Sung et al., 2015)	School-year level classification (Year1–12)	Lexical (word length, stroke count); Syntactic, semantic features
Arabic	(AbuShaira, 2011)		Morphological, lexical, syntactic text features
Persian	(Behzadi and Mohammadi, 2017)		Sentence length, word length, lexical density
Hindi	(Kumar et al., 2020)		sentence length, word length, syllable/character counts
Indonesian	Dwiyanto's Score (Pranowo, 2011)		Average number of paragraphs, # sentences per paragraph; sentence length; share (in %) of: extended sentences, compound sentences, sentences with polysemy, passive sentences, unfamiliar words, abstract words, specialised terms, conjunctions, loan words and phrases)

Table 1: Non-English readability indices

Language	Formula	Features
Polish	Jastrzębski's Index (Jastrzębski, 1981)	ASL, AWL
Czech	Flesch adaptation (Čech, 2013)	ASL, AWL (syllables)
Slovak	Flesch adaptation (Ivanová, 2010)	ASL, AWL (syllables)
Serbian & Croatian	Flesch / LIX (Mihaljević and Skelin, 2017)	ASL proportion of long words (>6 chars), syllable counts
Russian	(Solovyev et al., 2018)	sentence length, word length, syntactic complexity, vocabulary metrics
Russian	(Solovyev et al., 2023)	ASL, AWL, frequency list of the Russian elementary school textbooks

Table 2: Readability indices for Slavic languages

Additional features derived from the original ones (and known as *basis functions*) can be added to the data in order to search for non-linear relationships, e.g. if it appears that the growth of the readability index is faster than linear with respect to the sentence length (sl), one could add the feature (sl^2) in order to better approximate that relationship. Similarly, interaction terms, that is, the product of two original features could be added as a new feature to capture the fact that doubling both the average word length and the average sentence length results in more than double the growth of the readability score. All of the above can easily be achieved through the use of polynomial regression, which combines all original features into all possible terms of up to a certain order n, e.g. for $n = 2$, all terms of type $x_i \times x_j \quad \forall i, \forall j$ will be added.

The result of the regression is evaluated on unseen data using the so called coefficient of determination, r^2. A value of 1 indicates all unseen data fits perfectly the model, $r^2 = 0$ corresponds to a model that is no better than simply predicting the average score of the texts in the training data set in all cases, without considering any of the attributes. Negative values of r^2 are possible despite the somewhat oddly chosen name of this evaluation metric, and would suggest a fit that is even worse, e.g. the model predicts trends that are opposite to the ones observed in the data.

The most common features used in the majority of related indices are the average word length expressed as a number of characters and the average number of words in the sentence. We are also adopting these here. In addition, we consider the average number of syllables per word, which is calculated as the number of vowels or graphemes containing a vowel. Bulgarian orthography is mostly phonetic with a few infrequent digraphs (дж, дз) and complex graphemes, such as ч = ch, щ = sht, ю = iu/yu and я = ia/ya, yet such a feature may prove useful if readability is related to, say, the length of prosody patterns within a word.

Adding the standard deviation σ of at least some of the features is another attempt better to represent the underlying distributions: while two texts may have the same average number of words per sentence, a greater σ would mean the text is more likely to have sentences of extreme length, which may prove more challenging to the reader.

It may be helpful to mention that the default expectation for word count is to find an overdispersed distribution, with variance σ^2 greater than the mean, e.g.:

$$Variance = Mean + \frac{Mean^2}{k}$$

while the number of letters per word produces tighter distributions with variance closer to the mean, which is modelled well by a Poisson ($\mu = \sigma^2$) or negative Binomial distribution.

We have also experimented with features reflecting how common or rare a given word is in the data. This was either quantified, using Shannon entropy, $log_2 P(w_i)$ or represented as a Boolean feature expressing whether the word appeared at all in the training data.

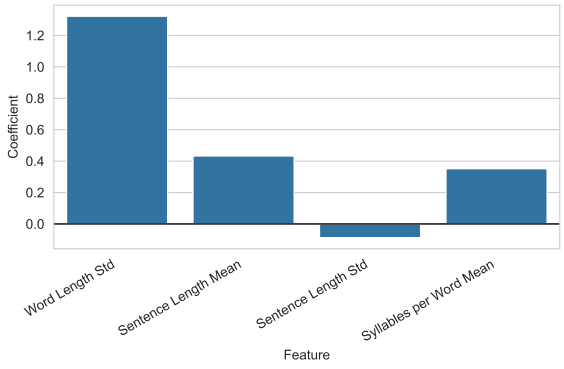


Figure 2: Coefficients for best-performing Linear Regression

Figure 3: Coefficients for best-performing Polynomial-2 Regression

5 Results

We found that linear regression offered the easiest to interpret model, but performed worse than the polynomial regression with degree two. We also discovered that removing the words of the closed vocabulary always improved the results, so all reported findings in the rest of this section make use of this preprocessing step. The following results are based on what we referred to as the main dataset earlier in the text.

The coefficients for the best-performing linear and quadratic regression are displayed in Figures 2 and 3.

Considering the linear regression approach, we found that the input features that result in the best fit are the standard deviation of word length, the mean and the standard deviation of sentence length, and the mean number of syllables per word (x_1 to x_4):

$$\hat{y} = 1.32x_1 + 0.43x_2 - 0.09x_3 + 0.35x_4$$

The $r^2 = 0.41$ goodness of fit is an encouraging, if not exciting, result. The fact that word length standard deviation is present in the equation, but the mean is not, appears less puzzling when we are reminded of the Poisson-like distribution of this property, where $\mu \approx \sigma$.

The polynomial regression with degree 2 outperformed the linear regression by almost 44% with $r^2 = 0.59$. Seven features were used in the best model, namely, word length mean, sentence length mean and standard deviation, syllables per word count mean and standard deviation, and Shannon entropy mean and standard deviation, x_1 to x_7, respectively.

$$\begin{aligned}\hat{y} = {} & \\ & - 2.6372\,x_1 - 0.7867\,x_2 + 1.7984\,x_3 \\ & - 4.1278\,x_1^2 - 2.8260\,x_2^2 - 3.8334\,x_3^2 \\ & + 2.0329\,x_4 + 1.8077\,x_5 - 0.5070\,x_6 + 0.2704\,x_7 \\ & - 4.0808\,x_4^2 - 0.8801\,x_5^2 + 0.2061\,x_6^2 - 0.5575\,x_7^2 \\ & + 11.9439\,x_1x_2 - 25.1250\,x_1x_3 + 8.9497\,x_1x_4 \\ & - 1.2169\,x_1x_5 + 2.9605\,x_1x_6 + 2.4798\,x_1x_7 \\ & + 7.2979\,x_2x_3 - 9.0990\,x_2x_4 + 0.9970\,x_2x_5 \\ & + 2.2256\,x_2x_6 - 3.2815\,x_2x_7 + 21.9574\,x_3x_4 \\ & - 2.2181\,x_3x_5 - 1.0012\,x_3x_6 + 4.8432\,x_3x_7 \\ & + 1.1557\,x_4x_5 - 2.0520\,x_4x_6 - 1.9476\,x_4x_7 \\ & - 0.9511\,x_5x_6 - 0.0247\,x_5x_7 + 0.8845\,x_6x_7\end{aligned}$$

We experimented with Polynomial-2 regression not containing any interaction terms to see if the better results stem in the non-linear relationship between individual features and the predicted output, but the results dropped substantially, to $r^2 = 0.43$.

$$\begin{aligned}\hat{y} = {} & \\ & - 9.4568\,x_1 + 0.0155\,x_2 + 0.8595\,x_3 + 5.9938\,x_4 \\ & + 10.8226\,x_5 - 1.9557\,x_6 + 2.9524\,x_7 \\ & + 7.8462\,x_1^2 - 0.0820\,x_2^2 - 0.6443\,x_3^2 - 4.6842\,x_4^2 \\ & - 9.3665\,x_5^2 + 1.0714\,x_6^2 - 3.1744\,x_7^2\end{aligned}$$

6 Discussion

The results so far indicate that more data may be needed, as the strong contribution of interaction terms, which do not appear in

any of the indices of Section 2.1, is suggestive of overfitting. Our unreported results on the assessment texts alone, which are essentially end of school year comprehension questions, showed that the authors of these questions did not make an effort to adjust their style to the age of the reader in any meaningful way. We also discovered that removing the words of the closed lexicon (the so-called stop list) improves the outcome.

7 Conclusions

In conclusion, this study reviewed established readability metrics across multiple language families and introduced the first attempt to develop a readability index for Bulgarian. By analyzing end-of-school-year assessment materials and children's literature, key linguistic features–word length, sentence length, syllable count, and information content– were extracted and statistically modelled against grade levels. The findings indicate that polynomial regression more effectively captures the non-linear relationship between these features and text difficulty compared to linear models, though with reduced interpretability. This research lays the groundwork for a Bulgarian readability index, with promising applications in educational content creation, adaptive learning systems, and the legal domain.

8 Work Limitations

We are aware that the surface linguistic characteristics of our choice do not reflect all aspects of text comprehensibility. At the same time, using only features that are expected to have a bearing on the level of readability was a helpful way to gauge the suitability of texts used. We expect to see additional features, such as embeddings, included in our future experiments as we gradually expand our corpus.

9 Ethical and Legal Considerations and Broader Impact

We are only using data in the public domain in this study. Publishing a readability index can only contribute to social goals, such as providing accessible, easy to understand information to the public.

Our work sheds light on the surface linguistic complexity and readability characteristics of Bulgarian exam materials and Bulgarian literature books recommended to specific school age groups. Our finding that the materials for different school classes cannot be distinguished on the basis of psycholinguistic characteristics known to affect text comprehension (DuBay, 2004) should probably lead to more in-depth experiments to test whether such materials are appropriate for the Bulgarian school grades they were designed for. In such a way, our findings may assist in improving school education.

Our specific interest is creating a formula to provide a measurable way to estimate the readability of Bulgarian laws. The Law on Normative Acts and Decree No 883 of 24.04.1974 on the implementation of the Law on Normative Acts represent the Bulgarian legal framework that ensures new laws are clear, complete, and easy to interpret. Its main principles are: Precision of Norms, Interpretation of Ambiguities, Prohibition of Extensive Interpretation, and Filling Legal Gaps. These principles are designed to ensure clarity, completeness, and legal predictability, protecting citizens' rights and maintaining consistency in the legal system. The ability to quantify these desirable properties of a text would provide support to the strive for high quality legislation that meets the requirements of the rule of law and ensures legal certainty.

10 Acknowledgments

This research was part of the GATE project B-CLEAR funded by the Horizon 2020 WIDESPREAD2018-2020 TEAMING Phase 2 Programme under grant agreement No. 857155, the Programme "Research, Innovation and Digitalisation for Smart Transformation" 2021–2027 (PRIDST) under grant agreement No. BG16RFPR002-1.014-0010-C01, and the project BROD (Bulgarian-Romanian Observatory of Digital Media) funded by the EU Digital Europe Programme under contract No. 101083730. We also thank the national infrastructure CLaDA-BG (https://clada-bg.eu/bg/) for making available the word stop list and word frequency lexicon used here, and Prof. Tatyana Angelova for her suggestions.

References

M. AbuShaira. 2011. Readability formulas: An overview and a proposed Arabic readability formula. *International Journal of Academic Research*, 3(1):79–84.

Suha S. Al-Thanyyan and Aqil M. Azmi. 2021. Automated text simplification: a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Noof Abdullah Alfear, Dimitar Kazakov, and Hend Al-Khalifa. 2024. Meta-evaluation of sentence simplification metrics. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11229–11235, Torino, Italia. ELRA and ICCL.

Toni Amstad. 1978. *Wie verständlich sind unsere Zeitungen?* Ph.D. thesis, University of Zürich.

Tatyana Angelova. 2018. The child and his/her literacy based on international study PIRLS'2006 and national external assessment for primary education – 4th grade. *Детето и педагогиката*, pages 133–142.

Richard Bamberger and Eva Vanecek. 1984. *Lesen - Verstehen - Lernen - Schreiben*. Jugend Volk Wien.

Bahareh Behzadi and Mohammad Mohammadi. 2017. Measuring the readability of Persian texts: Development and evaluation of a new Persian readability formula. *Studies in Literature and Language*, 15(2):1–10.

Carl-Hugo Björnsson. 1968. *Läsbarhet*. Liber.

Tanya Borisova. 2017. *Fundamentals of Reading Literacy Grades 1–4*. Yambol: Diagal Print.

P. Brouwer. 1963. *Tekstanalyse: Een methode voor het bepalen van de moeilijkheidsgraad van teksten*. Wolters.

Meri Coleman and T. L. Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283–284.

Max Coltheart. 1981. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505.

Edgar Dale and Jeanne S. Chall. 1949. The concept of readability. *Elementary English*, 26(1):19–26.

Mihai Dascălu, Silviu Bîrleanu, and Scott A. Crossley. 2015. Analyzing Romanian texts: Proposing a readability formula for Romanian language. In *2015 International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 47–53.

J. Douma. 1960. Een Nederlandse leesbaarheidsindex. *Tijdschrift voor Taal en Letteren*.

William H. DuBay. 2004. The principles of readability. *Education Resources Information Center (ERIC)*. 76 pages, URL: https://eric.ed.gov/?id=ED490073.

Rudolf F. Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

Daniela B Friedman, Manju Tanwar, and Jane VE Richter. 2008. Evaluation of online disaster and emergency preparedness resources. *Prehospital and disaster medicine*, 23(5):438–446.

Atsushi Fujita, Satoshi Sato, and Tetsuro Ishikawa. 2012. Automatic readability assessment for Japanese text.

Itziar Gonzalez-Dios, María Jesús Aranzabe, Arantza Díaz de Ilarraza, and Haritz Salaberri. 2014. Simple or complex? assessing the readability of Basque texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 334–344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202.

Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill.

Gretchen Hargis, Ann Hernandez, Polly Hughes, Jennifer Ramaker, Shannon Rouiller, and Elizabeth Wilde. 1998. *Developing Quality Technical Information: A Handbook for Writers and Editors*. Prentice Hall PTR.

Iveta Ivanová. 2010. Hodnotenie čitateľnosti slovenských textov. *Slovenská reč*, 75(1):1–11.

Jerzy Jastrzębski. 1981. *Czytelność tekstów polskich*. Wydawnictwo Uniwersytetu Gdańskiego.

J. Peter Kincaid, Roger P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas for navy enlisted personnel. Technical Report 8-75, Naval Technical Training Command Millington TN Research Branch.

George R. Klare. 1963. *The Measurement of Readability*. Iowa State University Press, Ames, Iowa.

Svetla Koeva, Valentina Stefanova, Ivelina Stoyanova, and Maria Todorova. 2023. Assessing the reading literacy of early graders. *Bulgarian Language*, 70(4):92–102.

Shiv Kumar, Sandeep Kumar, and Rajendra Singh. 2020. A readability index for Hindi language texts. *Procedia Computer Science*, 167:1658–1667.

Alessandro Lento, Andrea Nadalini, Marcello Ferro, Claudia Marzi, Vito Pirrelli, Tsvetana Dimitrova, Hristina Kukova, Valentina Stefanova, Maria Todorova, and Svetla Koeva. 2024. Assessing reading literacy of Bulgarian pupils with finger–tracking. In *Proceedings of the Sixth International Conference on Computational Linguistics in Bulgaria (CLIB 2024)*, pages 140–149.

Iñaki Madrazo Azpiazu and Sole Pera. 2021. Multilingual and cross-lingual readability: Novel resources and methods for the assessment of reading difficulty. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4671.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2019. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 45(3):495–530.

G. Harry McLaughlin. 1969. Smog grading — a new readability formula. *Journal of Reading*, 12(8):639–646.

Bojan Mihaljević and Ivana Skelin. 2017. Readability of croatian texts. *Information and Communication Technology*, 5:59–64.

Vito Pirelli and Svetla Koeva. 2024. Developing materials for assessing reading literacy and comprehension of early graders in Bulgaria and Italy. *Foreign Language Teaching*, 51(1).

D. D. Pranowo. 2011. Instrument of Indonesian texts readability. http://staff.uny.ac.id/sites/default/files/Readability%20instrumentthesis.pdf. [Online].

Anibal Quispesaravia, Wilker Perez, Maria S. Cabezudo, and Fernando Alva-Manchego. 2016. Coh-metrixesp: A complexity analysis tool for documents written in Spanish. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4694–4698.

Horacio Saggion and Graeme Hirst. 2017. *Automatic text simplification*, volume 32. Springer.

Caroline Scarton and Sandra M. Aluísio. 2010. Coh-metrix-port: A readability assessment tool for texts in Brazilian Portuguese. In *Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language, Extended Activities Proceedings (PROPOR)*, volume 10, pages 1–2.

Richard J. Senter and Edward A. Smith. 1967. Automated readability index. Technical Report AMRL-TR-66-220, Aerospace Medical Research Laboratories, Wright-Patterson AFB, Ohio.

Denitza Sharkova and Kosta Garov. 2015. Приложения на облачни технологии в обучението. In *VIII Национална конференция „Образованието и изследванията в информационното общество"*, page 166.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL-International Journal of Applied Linguistics*, 165(2):259–298.

Valery Dmitrievich Solovyev, VV Ivanov, and Marina Ivanovna Solnyshkina. 2023. Readability formulas for three levels of Russian school textbooks. *Записки научных семинаров ПОМИ*, 529(0):140–156.

Vladimir Solovyev, Marina Solnyshkina, Viktor Ivanov, and Andrey Kiselnikov. 2018. Assessment of reading difficulty levels in Russian texts. *Procedia Computer Science*, 136:234–239.

Yao-Ting Sung, Tsung-Hsien Chang, and Kuan-Eng Huang. 2015. Chinese readability index explorer (CRIE): A Chinese readability assessment system. *Behavior Research Methods*, 47(2):340–351.

Irina Temnikova, Sarah Vieweg, and Carlos Castillo. 2015. The case for readability of crisis communications in social media. In *Proceedings of the 24th international conference on world wide web*, pages 1245–1250.

Sowmya Vajjala and Ivana Lučić. 2018. One size does not fit all: Multi-task learning for genre-based readability assessment. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 173–179.

Sowmya Vajjala and Detmar Meurers. 2012. On improving the accuracy of readability classification using insights from second language acquisition. In *Proceedings of the Seventh Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–173.

Kalina Yocheva. 2017. Approaches to the formation of reading literacy in bilingual students · Подходи при формирането на четивна грамотност у ученици билингви. *Rhetorics and communication · Риторика и комуникации.*

Radek Čech. 2013. Automatické hodnocení čitelnosti češtiny. In *Proceedings of the 9th Workshop on Recent Advances in Slavonic Natural Language Processing*, pages 33–40.

Author Index