Massively Multilingual Language Modeling and Adaptation

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

eingereicht von
Peiqin Lin

München, den 18. März 2025

Erstgutachter: Prof. Dr. Hinrich Schütze
Zweitgutachter: Prof. Dr. André Martins
Drittgutachter: Prof. Dr. David Adelani

Tag der Einreichung: 18. März 2025
Tag der mündlichen Prüfung: 22. Oktober 2025

Eidesstattliche Versicherung
(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5.)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig ohne unerlaubte Beihilfe angefertigt ist.

München, den 18. März 2025.

_____
Peiqin Lin

Abstract

Pretrained language models have significantly advanced natural language processing (NLP), but they are often tailored to English or a limited number of high-resource languages, leading to a performance gap for low-resource languages. This thesis presents a series of publications aimed at improving language modeling and task adaptation for low-resource languages.

The first three publications, Glot500, MaLA500, and EMMA500, focus on creating massively multilingual language models. The Glot500 publication presents Glot500-c, a large-scale multilingual corpus covering 534 language-script pairs, sourced from diverse datasets after thorough cleaning and deduplication. Based on this corpus, Glot500-m extends the vocabulary and continues pretraining on XLM-R (a language model covering about 100 languages), achieving superior performance in both high- and low-resource languages. MaLA500 builds upon Glot500 by scaling the model to larger LLMs using LLaMA 2 and Glot500-c, resulting in a highly capable language model that sets new state-of-the-art results in in-context learning. EMMA500 further enhances MaLA500 by compiling the MaLA corpus and enriching it with curated resources from diverse domains, surpassing Glot500-c in both quality and quantity. Compared to MaLA500, EMMA500 exhibits robust performance across a broad set of benchmarks, including a wide array of multilingual tasks and open-ended generation benchmarks.

Additionally, the publication titled "A Recipe for Parallel Corpora Exploitation for Multilingual Large Language Models" investigated how the quality and quantity of parallel corpora, as well as training objectives and model size, influence the performance of multilingual large language models across various languages and tasks. The findings provide valuable insights into the best use of parallel corpora to enhance multilingual models, extending the generalizability of earlier research across a broader range of languages and scenarios.

The mPLM-Sim and XAMPLER publications present innovative approaches to cross-lingual transfer for both high- and low-resource languages. mPLM-Sim introduces a novel language similarity metric based on multilingual pretrained models, outperforming traditional linguistic similarity measures. This metric enhances zero-shot cross-lingual transfer for both syntactic and semantic tasks by identifying the best source languages. The XAMPLER project tackles the challenge of cross-lingual in-context learning using only annotated English data. XAMPLER trains a cross-lingual retriever using both positive and negative English examples generated by MaLA500, a large multilingual language model. By leveraging the retriever's cross-lingual capabilities, XAMPLER selects English examples to serve as few-shot samples for in-context learning in target languages. Experiments demonstrate that XAMPLER significantly improves in-context learning performance across a wide range of languages.

Zusammenfassung

Vortrainierte Sprachmodelle haben die Verarbeitung natürlicher Sprache (NLP) erheblich verbessert, sind jedoch häufig auf Englisch oder eine begrenzte Anzahl ressourcenintensiver Sprachen zugeschnitten, was zu Leistungslücken bei ressourcenarmen Sprachen führt. Diese Arbeit präsentiert eine Reihe von Veröffentlichungen, die darauf abzielen, die Sprachmodellierung und die Aufgabenanpassung für ressourcenarme Sprachen zu verbessern.

Die ersten drei Veröffentlichungen, Glot500, MaLA500 und EMMA500, konzentrieren sich auf die Erstellung von massiv mehrsprachigen Sprachmodellen. Die Glot500-Veröffentlichung präsentiert Glot500-c, ein umfangreiches mehrsprachiges Korpus, das 534 Sprach-Schrift-Paare umfasst, die nach gründlicher Bereinigung und Deduplizierung aus verschiedenen Datensätzen stammen. Basierend auf diesem Korpus erweitert Glot500-m das Vokabular und setzt das Vortraining auf XLM-R (ein Sprachmodell, das etwa 100 Sprachen umfasst) fort, wodurch sowohl in ressourcenintensiven als auch in ressourcenarmen Sprachen überlegene Leistungen erzielt werden. MaLA500 baut auf Glot500 auf, indem das Modell mithilfe von LLaMA 2 und Glot500-c auf größere LLMs skaliert wird, was zu einem hochleistungsfähigen Sprachmodell führt, das neue Spitzenergebnisse im kontextbezogenen Lernen erzielt. EMMA500 verbessert MaLA500 zusätzlich, indem es das MaLA-Korpus kompiliert und mit kuratierten Ressourcen aus verschiedenen Bereichen anreichert und Glot500-c sowohl in Qualität als auch in Quantität übertrifft. Im Vergleich zu MaLA500 zeigt EMMA500 eine robuste Leistung bei einer breiten Palette von Benchmarks, darunter eine breite Palette mehrsprachiger Aufgaben und offener Generierungs-Benchmarks.

Darüber hinaus untersuchte die Veröffentlichung mit dem Titel “A Recipe for Parallel Corpora Exploitation for Multilingual Large Language Models”, wie sich Qualität und Quantität paralleler Korpora sowie Trainingsziele und Modellgröße auf die Leistung mehrsprachiger Large Language Models in verschiedenen Sprachen und Aufgaben auswirken. Die Ergebnisse liefern wertvolle Einblicke in die optimale Verwendung paralleler Korpora zur Verbesserung mehrsprachiger Modelle und erweitern die Generalisierbarkeit früherer Forschung auf ein breiteres Spektrum von Sprachen und Szenarien.

Die Veröffentlichungen mPLM-Sim und XAMPLER präsentieren innovative Ansätze für den sprachübergreifenden Transfer sowohl für ressourcenintensive als auch für ressourcenarme Sprachen. mPLM-Sim führt eine neuartige Sprachähnlichkeitsmetrik ein, die auf mehrsprachigen, vorab trainierten Modellen basiert und traditionelle sprachliche Ähnlichkeitsmaße übertrifft. Diese Metrik verbessert die sprachenübergreifende Übertragung ohne Vorgabe für sowohl syntaktische als auch semantische Aufgaben, indem sie optimale Ausgangssprachen identifiziert. Das XAMPLER-Projekt stellt sich der Herausforderung des sprachübergreifenden

Lernens im Kontext, indem es ausschließlich annotierte englische Daten verwendet. XAMPLER trainiert einen sprachübergreifenden Retriever mit positiven und negativen englischen Beispielen, die von MaLA500, einem großen mehrsprachigen Sprachmodell, generiert werden. Indem XAMPLER die sprachübergreifenden Fähigkeiten des Retrievers nutzt, wählt es englische Beispiele aus, die als wenige Beispiele für das kontextbezogene Lernen in Zielsprachen dienen. Experimente zeigen, dass XAMPLER die Leistung des kontextbezogenen Lernens in einer Vielzahl von Sprachen erheblich verbessert.

Acknowledgement

I would like to express my deepest gratitude and highest respect to my supervisors, Prof. Hinrich Schütze and Prof. André Martins, for their invaluable guidance and support throughout my doctoral studies. Their profound knowledge, academic rigor, and insightful feedback have shaped my research and inspired me to pursue scientific excellence. I have learned not only how to conduct research, but also how to think critically, collaborate effectively, and stay curious in the face of challenges.

I am also deeply thankful to my colleagues and friends at CIS (Center for Information and Language Processing) and Sardine Lab, whose companionship, discussions, and encouragement made my PhD journey both intellectually stimulating and personally enriching. The collaborative environment and open exchange of ideas at CIS and Sardine have greatly broadened my horizons and contributed immensely to my research.

I would like to extend my sincere appreciation to all collaborators, mentors, and reviewers who have provided feedback and support throughout my publications and thesis work. Their comments and insights have been invaluable in refining my ideas and improving this dissertation.

Finally, I would like to thank my family and my friends for their unconditional love, understanding, and encouragement. Their constant support has been the strongest source of motivation behind my studies and personal growth.

Publications and Declaration of Co-Authorship

Chapter 2 corresponds to the following publication:

> Ayyoob Imani, **Peiqin Lin**, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André F. T. Martins, François Yvon, Hinrich Schütze. *Glot500: Scaling Multilingual Corpora and Language Models to 500 Languages.* In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL), July 2023.

I contributed to data collection by handling various data sources, including Earthlings, Huggingface datasets, LeipzigData, Oscar22, Tatoeba, and Wikipedia. I was responsible for implementing the model training, which involved extending the vocabulary and continuing pretraining. Additionally, I conducted the evaluation for tasks such as sequence labeling and sentence retrieval. I also participated in the paper's writing, focusing on the parts that I was responsible for.

Chapter 3 corresponds to the following publication:

> **Peiqin Lin**, Shaoxiong Ji, Jörg Tiedemann, Andre Martins, Hinrich Schütze. *MaLA-500: Massive Language Adaptation of Large Language Models.* In Arxiv 2401.13303.

I implemented the vocabulary extension and assisted with debugging the training process. Additionally, I conducted the evaluation of the pretrained model on both intrinsic tasks and downstream applications. Furthermore, I contributed to the writing of the paper, where I thoroughly documented the work I completed.

Chapter 4 corresponds to the following publication:

> Shaoxiong Ji, Zihao Li, Indraneil Paul, Jaakko Paavola, **Peiqin Lin**, Pinzhen Chen, Dayyán O'Brien, Hengyu Luo, Hinrich Schütze, Jörg

Tiedemann, Barry Haddow *Emma-500: Enhancing Massively Multilingual Adaptation of Large Language Models.* In Arxiv 2409.17892.

I contributed to the intrinsic evaluation of this work by analyzing two test sets spanning diverse languages. Additionally, I actively participated in project discussions and played a key role in drafting and refining the manuscript.

Chapter 5 corresponds to the following publication:

Peiqin Lin, Andre Martins, Hinrich Schütze. *A Recipe of Parallel Corpora Exploitation for Multilingual Large Language Models.* In Arxiv 2407.00436.

In this publication, I conducted the literature review and designed the research questions. I implemented the solutions for all the research questions and wrote the paper detailing the findings.

Chapter 6 corresponds to the following publication:

Peiqin Lin, Chengzhi Hu, Zheyu Zhang, Andre Martins, Hinrich Schütze. *mPLM-Sim: Better Cross-Lingual Similarity and Transfer in Multilingual Pretrained Language Models.* In The 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL): Findings, March 2024.

I led the project, beginning with a comprehensive literature review and feasibility study to define the project's methodology. I facilitated team meetings to brainstorm and address the four key research questions. My primary contributions included implementing the correlation analysis between mPLM-Sim and linguistic measures, as well as conducting cross-lingual transfer evaluations on sequence labeling and classification tasks. Additionally, I was the lead author of the paper.

Chapter 7 corresponds to the following publication:

Peiqin Lin, Andre Martins, Hinrich Schütze. *XAMPLER: Learning to Retrieve Cross-Lingual In-Context Examples.* In Arxiv 2405.05116.

In this publication, I conducted the literature review and introduced the methodology. I implemented the method and performed evaluations on text classification. Furthermore, I wrote the paper, detailing both the methodology and the evaluation results.

München, den 18. März 2025

Peiqin Lin

Declaration on Writing Aids

In writing this thesis, I used ChatGPT as a tool to assist with various aspects of the introduction chapter. Below, I outline the specific ways in which the tool was utilized to ensure transparency and adherence to ethical writing practices.

Writing Refinement For Sections 1.1, 1.2, 1.3, 1.4, and 1.6, I leveraged ChatGPT to revise and polish my initial drafts. This included fixing grammatical errors, rephrasing awkward sentences, and refining the flow. This iterative process involved integrating AI-generated suggestions with my original content, followed by further revisions to ensure alignment with my objectives for the chapter.

Foundation Section In drafting the foundation section, I provided ChatGPT with general instructions, key bullet points, initial content snippets, and the primary insights I wanted to emphasize. The tool was used to generate an initial draft, which underwent multiple iterations of refinement. I carefully reviewed, rewrote, and edited the output to produce a polished and cohesive final version, ensuring it met the required academic standards.

Mathematical Equations For standard formulas commonly used in deep learning and natural language processing, I leveraged ChatGPT to assist in generating LaTeX source code. I manually defined the necessary notation and provided specific prompts to guide the AI in drafting the equations. Each equation was reviewed and, where necessary, adjusted to ensure accuracy, mathematical integrity, and relevance to the thesis content.

Contents

Chapter 1

Introduction

1.1 Motivation

Language models have evolved significantly over time, starting from smaller models such as GPT (Radford et al., 2018), BERT (Devlin et al., 2019), and BART (Lewis et al., 2020), to large-scale models such as T5 (Raffel et al., 2020), GPT-3 (Brown et al., 2020), LLaMA (Touvron et al., 2023a,b; Dubey et al., 2024), Qwen (Bai et al., 2023), and Mistral (Jiang et al., 2023). Additionally, commercial products such as Gemini (Anil et al., 2023a; Rivière et al., 2024) and ChatGPT[1] have further demonstrated the practical impact of these advancements. These developments have brought transformative benefits across diverse domains, including daily life, scientific research, and software development.

Despite these advancements, most language models are predominantly designed and optimized for English or a limited subset of high-resource languages, resulting in a substantial performance gap for low-resource languages. Given that over 7,000 languages are spoken worldwide,[2] the vast majority remain underrepresented in today's technological advancements. This imbalance not only restricts equitable access to the transformative benefits of AI but also risks marginalizing speakers of these languages in digital spaces, deepening the existing digital divide. Bridging this gap is crucial for promoting inclusivity and ensuring that language technologies cater to a truly global audience.

A major challenge in training or adapting language models for low-resource languages lies in the severe scarcity of data (Joshi et al., 2020). As illustrated in Figure 1.1, the majority of languages fall into Group 0, characterized by a lack of both the unlabeled data required for language modeling and the labeled data essential for fine-tuning and evaluation. In contrast, only a small fraction

[1] https://openai.com/blog/chatgpt

[2] https://www.ethnologue.com/insights/how-many-languages/

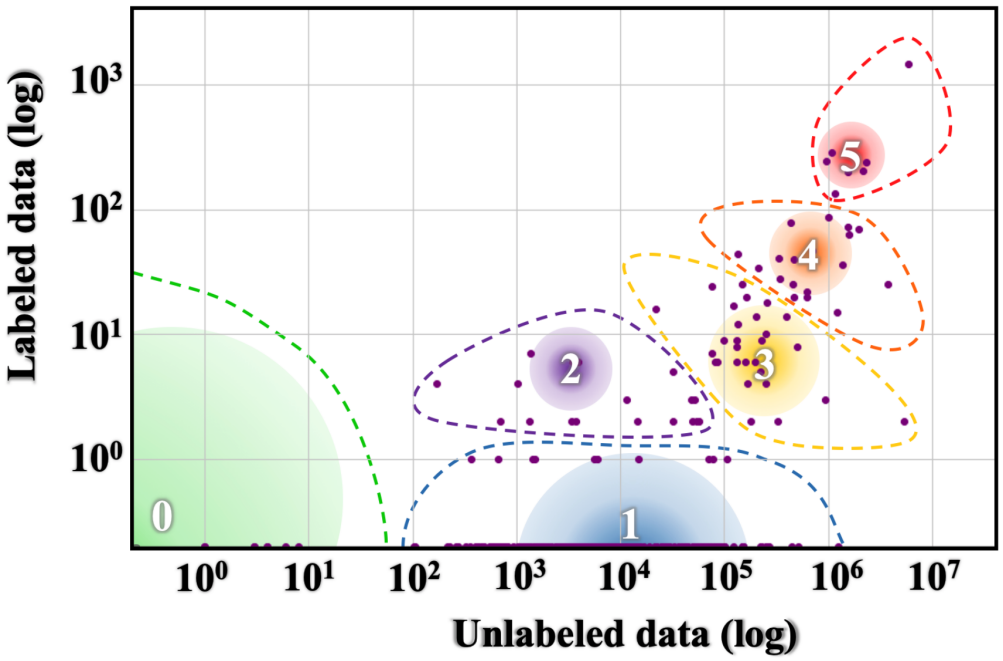


Figure 1.1: Language Resource Distribution. Based on available labeled data (LDC, LRE) and unlabeled data (Wikipedia, Web), Joshi et al. (2020) categorized the world's languages into several groups, i.e., The Winners (5), The Underdogs (4), The Rising Stars (3), The Hopefuls (2), The Scraping-Bys (1), and The Left-Behinds (0). The figure was taken from Joshi et al. (2020).

of languages belong to Group 5, which has sufficient unlabeled and labeled data for effective training and task adaptation. This pronounced data scarcity presents significant barriers to developing and optimizing robust language models for low-resource languages.

To address the challenge of data scarcity in low-resource languages, we investigate key research questions focusing on two critical phases of massively multilingual language models: modeling and adaptation.

Modeling: How can limited low-resource language data and parallel data enhance pre-training?

Multilingual language models primarily rely on monolingual data for pre-training, with optional use of parallel data to improve cross-lingual alignment of representations across languages. As detailed in Section 1.2, we emphasize the importance of collecting data specific to low-resource languages and assess its impact on pre-training outcomes for both small and large-scale massively multilingual language models. Furthermore, we explore best practices for effectively incorporating parallel corpora between high-resource and low-resource languages, aiming to enhance the overall performance and adaptability of multilingual language models.

Adaptation: How can labeled data from high-resource languages facilitate task adaptation for low-resource languages?

Obtaining labeled data for low-resource languages is considerably more challenging than collecting unlabeled data, posing a critical bottleneck for effective task adaptation. As detailed in Section 1.3, we propose two strategies leveraging cross-lingual transfer: utilizing labeled data from high-resource languages to support low-resource languages. We develop robust methods tailored to various task adaptation paradigms for both small and large language models, ensuring reliable and consistent performance for low-resource languages.

By adopting this comprehensive dual-phase approach, we aim to address the resource gap for low-resource languages. This effort seeks to foster inclusivity and enhance the effectiveness of multilingual language models, ensuring broader and more equitable access to language technologies.

1.2 Modeling

Joint training with monolingual corpora from multiple languages has emerged as the dominant paradigm for developing multilingual language models. These multilingual language models range from small-sized ones, such as multilingual BERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), mBART (Liu et al., 2020), and mT5 (Xue et al., 2021), to larger language models, including XGLM (Lin et al., 2021), mGPT (Shliazhko et al., 2022), BLOOM (Scao et al., 2022), and Aya (Üstün et al., 2024). By leveraging joint training across diverse languages, these models achieve robust generalization and transfer capabilities, effectively supporting both high-resource and low-resource languages.

In addition, several techniques have been proposed to adapt existing multilingual language models to unseen languages. Adapter-based methods (Pfeiffer et al., 2020; Üstün et al., 2020; Pfeiffer et al., 2020; Nguyen et al., 2021; Faisal and Anastasopoulos, 2022; Yong et al., 2022) introduce lightweight modules that can be fine-tuned for specific languages without modifying the core model, making adaptation computationally efficient. Vocabulary extension and substitution (Chau et al., 2020; Wang et al., 2020a; Müller et al., 2020, 2021a; Pfeiffer et al., 2021; Chen et al., 2023; Downey et al., 2023; Cui et al., 2023; Zhao et al., 2024) address the limitations of fixed vocabularies by incorporating new tokens or replacing existing ones to accommodate unseen languages. Continued pre-training (Ebrahimi and Kann, 2021; Yong et al., 2022; Cui et al., 2023; Zhao et al., 2024) on monolingual or multilingual data allows the existing language models to refine their understanding of new languages. In-context learning (Shi et al., 2022; Ahuja et al., 2023; Huang et al., 2023; Etxaniz et al., 2023; Zhang et al., 2023; Lu et al., 2023a; Tanzer et al., 2024; Zhang et al., 2024b,a) leverages contextual examples

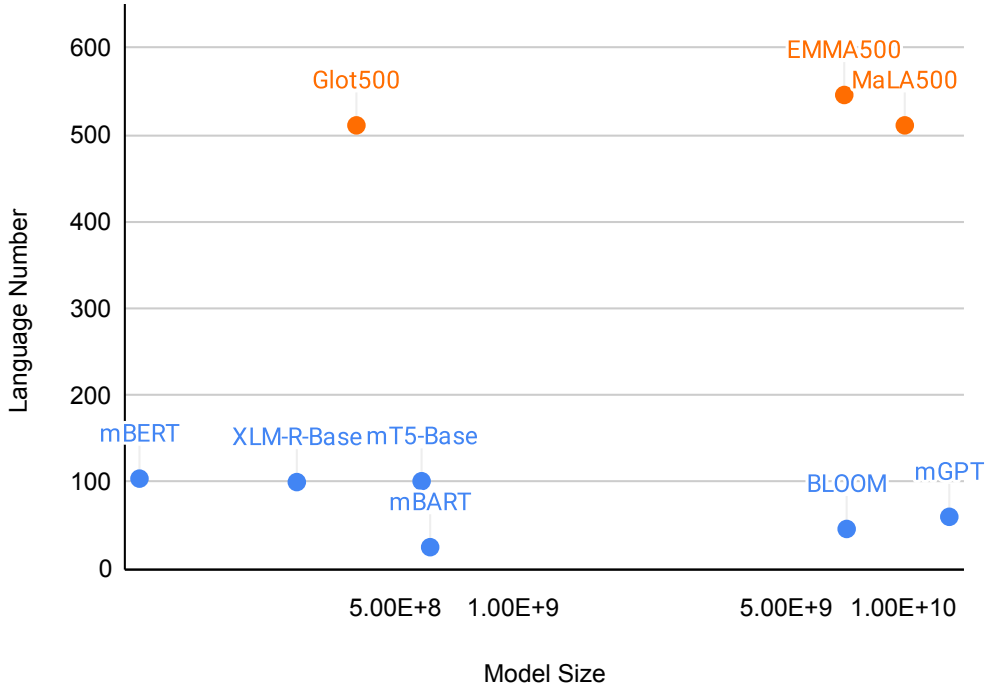


Figure 1.2: Comparison of language coverage and model size between existing multilingual models and Glot500, MaLA500, and EMMA500. Glot500, MaLA500, and EMMA500 support a significantly larger number of languages than existing multilingual language models.

and linguistic knowledge of the unseen language during inference, enabling models to dynamically adapt their predictions to the linguistic or task-specific nuances of unseen languages without additional training.

Despite significant progress, existing multilingual language models typically support only about 100 languages, focusing primarily on those with significant digital footprints or linguistic resources. Adaptation techniques, while effective, are often designed for incremental support, addressing one language at a time. This is still a long way from covering the approximately 7,000 languages spoken worldwide, highlighting a substantial gap in the development of truly massively multilingual models capable of handling a much broader and more diverse range of languages simultaneously.

As shown in Figure 1.2, Glot500 (Imani et al., 2023), MaLA500 (Lin et al., 2024b), and EMMA500 (Ji et al., 2024) push the boundary of language coverage, expanding from around 100 languages to over 500. Glot500 primarily targets smaller multilingual language models, while MaLA500 and EMMA500 are designed for larger, more powerful massively multilingual language models, en-

abling broader language support at scale.

Chapter 2 introduces the two primary contributions of Glot500 (Imani et al., 2023): Glot500-c and Glot500-m. Glot500-c is a large-scale multilingual corpus covering 534 language-script pairs. It was developed using a lightweight approach that builds upon previous advancements in multilingual data curation. Specifically, it aggregates data from existing multilingual datasets and carefully curated resources for low-resource languages, followed by thorough cleaning and deduplication. Building on this corpus, Glot500-m extends the vocabulary and continues pre-training on XLM-R, a compact, encoder-only multilingual language model that supports around 100 languages. This results in enhanced performance across both high- and low-resource languages on a broad range of tasks. Our analysis highlights several key factors that influence model quality, including corpus size and help from related languages.

Chapter 3 introduces MaLA500 (Lin et al., 2024b), which builds on Glot500 by scaling up the massively multilingual language model to larger LLMs using LLaMA 2, a large language model centered on English. This transition leverages the enhanced capabilities of LLaMA 2, resulting in a more powerful, massively multilingual model. As a result, MaLA500 achieves significant advancements in in-context learning, without requiring any parameter updates.

Chapter 4 presents EMMA500 (Ji et al., 2024), which further enhances the power of MaLA500. By compiling the MaLA corpus—a comprehensive multilingual dataset—and enriching it with curated resources from diverse domains, EMMA500 surpasses Glot500-c in its quality and quantity. Compared to MaLA500, EMMA500 exhibits robust performance across a broad set of benchmarks, including a wide array of multilingual tasks and PolyWrite, an open-ended generation benchmark developed as part of this study. Our results underscore the effectiveness of continual pre-training in expanding the language capacity of large language models, particularly for underrepresented languages. EMMA500 demonstrates significant improvements in cross-lingual transfer, task generalization, and language adaptability.

Parallel corpora are widely employed to boost the performance of multilingual language models, offering notable benefits for both smaller and larger multilingual language models. For smaller models, parallel corpora enhance performance through parallel training objectives (Conneau and Lample, 2019; Patra et al., 2022), enabling these models to effectively capture cross-lingual relationships. In the case of larger models, parallel corpora serve as valuable instruction data, helping refine the model's understanding and improving task performance across a wide range of languages (Cahyawijaya et al., 2023; Zhu et al., 2023; Li et al., 2023). Despite the widespread use of parallel corpora, the best practices for efficiently leveraging these parallel resources in large language models are still insufficiently explored.

Chapter 5 delves into the best practices for utilizing parallel corpora in large-scale multilingual language models (Lin et al., 2025a). It examines how the quality and quantity of parallel corpora, along with training objectives and model size, influence the performance of multilingual models across a broad spectrum of languages and tasks. Through a detailed analysis, the chapter highlights key factors that contribute to the effective integration of parallel data into multilingual training processes. The findings offer valuable insights into optimizing the use of parallel corpora, providing strategies for enhancing multilingual model performance and expanding the applicability of previous research to a greater number of languages and more diverse contexts. These insights pave the way for future advancements in multilingual language model development, facilitating their deployment in real-world applications that require broad language coverage and nuanced cross-lingual capabilities.

1.3 Adaptation

Acquiring labeled data for low-resource languages remains a labor-intensive challenge due to the limited availability of native speakers and linguistic resources. In contrast, obtaining labeled data for high-resource languages is comparatively easier due to their larger speaker bases and more established linguistic infrastructures. As a result, cross-lingual transfer has emerged as a powerful and widely adopted technique to evaluate low-resource languages. This approach leverages data from high-resource languages (source languages) and applies it to low-resource languages (target languages), eliminating the need for labeled data in the target languages and enabling models to perform well across language barriers.

In the era of small language models, fine-tuning plays a crucial role in adapting pre-trained language models to specific tasks. The standard workflow in cross-lingual transfer involves fine-tuning a pre-trained multilingual language model on available high-resource language data and then applying the fine-tuned model directly to low-resource language tasks, often with minimal additional training. This process enables models to generalize across languages, leveraging the information from high-resource languages to improve performance in low-resource contexts.

Chapter 6 introduces mPLM-Sim (Lin et al., 2024a), a novel approach aimed at enhancing cross-lingual transfer for small multilingual language models. mPLM-Sim proposes an innovative language similarity measure, leveraging learned representations from multilingual pre-trained language models (mPLMs) using multi-parallel corpora. In comparison to traditional linguistic measures, mPLM-Sim outperforms them in both qualitative and quantitative analyses. Furthermore, mPLM-Sim provides deep insights into the architecture of mPLMs by exploring language similarity across various model layers and comparing different mPLMs.

Notably, mPLM-Sim excels in selecting source languages for given target languages in zero-shot cross-lingual transfer, resulting in improved performance across a wide range of benchmarks. This capability allows mPLM-Sim to efficiently allocate resources, making it easier to adapt models to low-resource languages with better performance.

With the advent of large language models (LLMs), in-context learning has emerged as a critical technique. Through in-context learning, LLMs can directly perform new tasks by being provided with a few input-output examples alongside the input query (Brown et al., 2020). The success of in-context learning depends heavily on the careful selection of few-shot examples (Liu et al., 2022). Fine-tuned retrievers have been proven effective in retrieving relevant examples in English (Rubin et al., 2022), thus enhancing the performance of in-context learning. However, applying this approach to multilingual contexts encounters significant challenges, particularly in low-resource languages, due to the scarcity of labeled training data of low-resource languages.

Chapter 7 introduces XAMPLER (Lin et al., 2025b), an innovative approach to the challenge of selecting few-shot examples for input queries in low-resource languages. In the absence of labeled examples in these languages, XAMPLER leverages English samples as few-shot examples. To construct positive and negative pairs for contrastive learning during the fine-tuning of the example retriever, we leverage powerful large language models to label whether a candidate example is helpful or not for in-context learning of a given English query. A small massively multilingual language model serves as the foundation for fine-tuning the retriever. This entire process relies solely on annotated English data. By harnessing the cross-lingual transferability of the base multilingual language model, XAMPLER enables the fine-tuned retriever to identify relevant few-shot English examples for input queries across any language supported by the model. Experimental results show that XAMPLER achieves state-of-the-art performance in few-shot in-context learning across a wide range of languages, demonstrating its robustness and adaptability in low-resource settings.

Both mPLM-Sim and XAMPLER represent significant advancements in cross-lingual transfer, enhancing the use of both small- and large-scale language models for low-resource languages. mPLM-Sim refines source language selection for zero-shot transfer, while XAMPLER optimizes few-shot example selection using English data, enabling effective in-context learning for languages with no labeled data. Together, these methods boost performance in low-resource settings and help bridge the gap in language technology across a wide range of languages.

1.4 Outline

In this chapter, we provide the main topics covered in the publications throughout this thesis, along with the conclusions drawn from the research and suggestions for future work. Chapters 2, 3, and 4 focus on the development of massively multilingual language models, addressing the challenges associated with scaling these models to support a wide variety of languages. Chapter 2 introduces a massively multilingual corpus and leverages it to continue pre-training a small-scale multilingual language model, thereby creating a massively multilingual language model. Chapter 3 extends this approach to larger multilingual models, achieving strong performance in classification tasks via few-shot in-context learning. Chapter 4 advances the development of more sophisticated and powerful multilingual language models across a variety of tasks by incorporating improved data resources. Chapter 5 investigates the use of parallel data to enhance large language models, examining four key factors—quality and quantity of parallel corpora, training objectives, and model size—that impact the performance of multilingual models across a diverse set of languages and tasks. Chapter 6 introduces mPLM-Sim, a novel approach to measuring language similarity and improving cross-lingual transfer by optimizing source language selection. Finally, Chapter 7 presents XAMPLER, an advanced solution for retrieving relevant examples for in-context learning in low-resource languages.

1.5 Foundation

1.5.1 Transformer

The Transformer architecture, introduced by Vaswani et al. (2017), revolutionized natural language processing (NLP) by enabling more efficient scaling of language models compared to traditional recurrent architectures. Unlike recurrent neural networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which process input sequentially and struggle to capture long-range dependencies effectively, the Transformer leverages a self-attention mechanism that processes entire sequences in parallel. This parallelism not only significantly reduces training time but also enhances the model's capacity to capture complex relationships within the data.

As illustrated in Figure 1.3, the Transformer architecture is composed of two key components: the encoder and the decoder. Both components are built using layers of multi-head self-attention mechanisms and feed-forward networks, which are stacked multiple times. The encoder processes the input sequence through its layers to produce contextualized representations, capturing the relationships

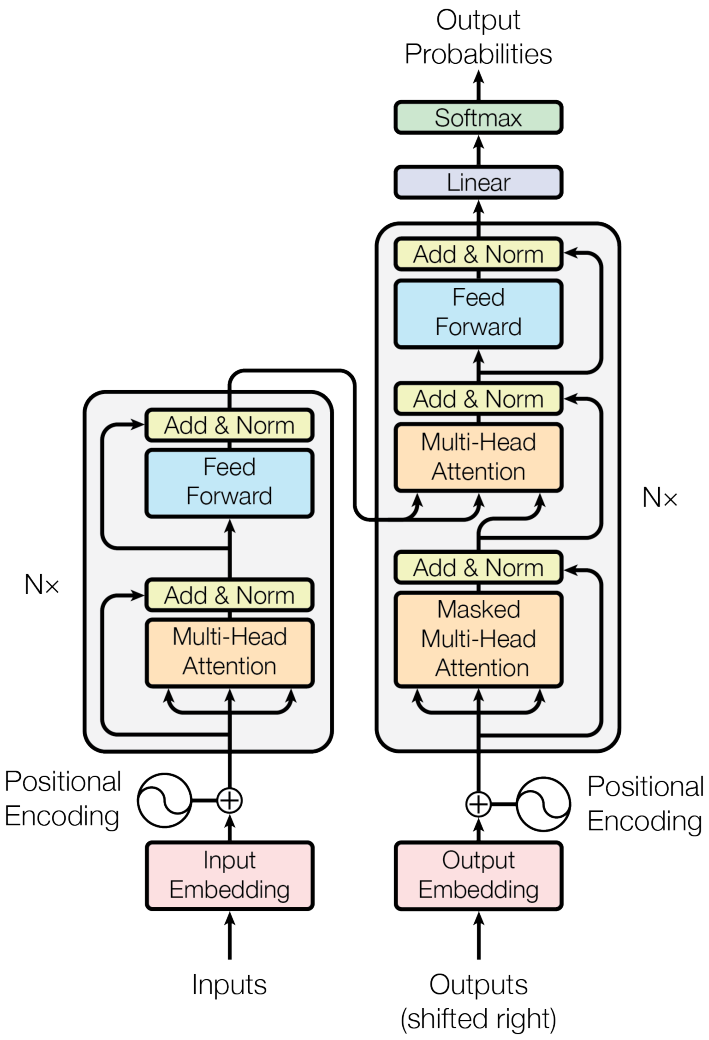


Figure 1.3: The Transformer architecture, as introduced by Vaswani et al. (2017).

between tokens within the input. The decoder generates the output sequence by attending to both the encoder's output and its own previously generated tokens.

One of the key innovations of the Transformer architecture is the multi-head attention mechanism, which enhances the model's ability to capture diverse contextual relationships within the input sequence. Unlike single-head attention, which provides a single representation for each token's interaction with the others, multi-head attention divides the attention computation into multiple heads, enabling the model to focus on different aspects of the sequence simultaneously.

As shown in Figure 1.4, the multi-head attention mechanism operates by projecting the input sequence into multiple subspaces, each corresponding to an attention head. Each head independently computes attention scores and generates a

attention.png

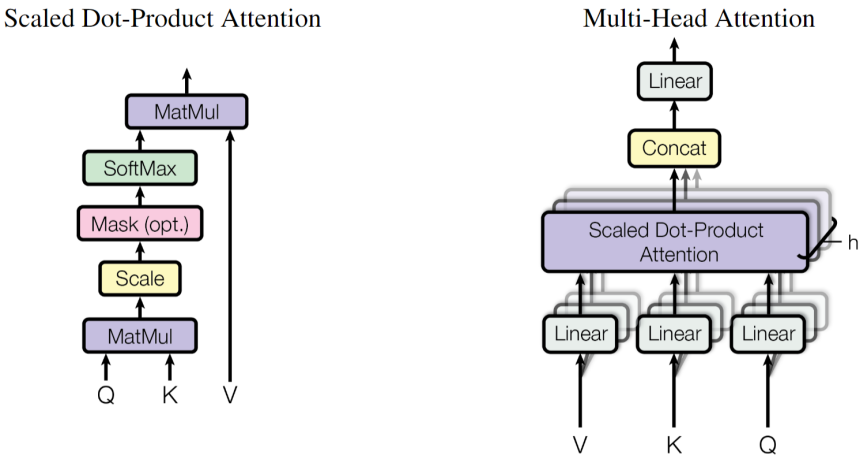


Figure 1.4: Scaled Dot-Product Attention (left) and Multi-Head Attention (right). Scaled Dot-Product Attention computes a weighted sum of values V based on the similarity between queries Q and keys K, normalized by the dimension of keys, d_k. Multi-Head Attention extends this by performing multiple parallel attention computations (heads) with independent parameterization, enabling the model to jointly attend to information from different representation subspaces. The figure was taken from Vaswani et al. (2017).

weighted representation of the sequence. These representations are then concatenated and linearly transformed to produce the final output.

Specifically, the process for a single attention head can be described as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1.1}$$

where Q, K, V are the query, key, and value matrices, derived from the input embeddings via learned linear transformations. d_k is the dimensionality of the key vectors, and $\sqrt{d_k}$ is used to scale the dot products, preventing them from growing too large. The softmax function ensures that the attention weights are normalized to sum to 1.

In multi-head attention, the mechanism is applied h times (where h is the number of heads), with separate projections for Q, K, and V for each head:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{1.2}$$

where W_i^Q, W_i^K, and W_i^V are learned weight matrices specific to the i-th head. The outputs of all heads are concatenated and linearly transformed:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{1.3}$$

where W^O is a learned output weight matrix.

The use of multiple heads allows the model to focus on different parts of the sequence simultaneously. For example, one head might focus on syntactic relationships, while another captures semantic relationships.

Unlike RNNs and their variants, such as LSTMs and GRUs, which process input tokens sequentially and inherently capture positional information through their architecture, Transformers lack a built-in mechanism to encode the order of tokens. Without positional encoding, a Transformer cannot differentiate between sequences with identical tokens but different orders. For instance, the sentences "The cat chased the mouse" and "The mouse chased the cat" would appear identical to the model, as it would treat both as containing the same set of tokens, despite their vastly different meanings.

To overcome this, Transformers incorporate positional encodings to inject information about token order. These encodings are added to the input embeddings at the base of the encoder and decoder stacks, enabling the model to distinguish between tokens at different positions within the sequence.

In the original Transformer architecture by Vaswani et al. (2017), position embeddings are generated using fixed, non-learnable functions. Each position p is encoded as a combination of sine and cosine functions with varying frequencies:

$$PE(p, 2i) = \sin\left(\frac{p}{10000^{2i/d}}\right) \tag{1.4}$$

$$PE(p, 2i+1) = \cos\left(\frac{p}{10000^{2i/d}}\right), \tag{1.5}$$

where d is the dimensionality of the embeddings, and i is the hidden dimension index.

Subsequent approaches, such as the learnable positional embeddings introduced by Devlin et al. (2019), replace these fixed encodings with trainable vectors. In this approach, each position in the sequence is assigned a unique vector, initialized randomly and optimized alongside the model's parameters during training. While this method offers greater flexibility and adapts more effectively to specific tasks, it is limited to a predefined maximum sequence length and lacks extrapolation capability to positions outside this range.

Recent advancements in positional encoding have addressed these limitations. Rotary Positional Embedding (RoPE) (Su et al., 2021) integrates learnable embeddings with rotational transformations, enabling better generalization and enhanced performance in language models. However, RoPE struggles with sequences exceeding the training length. To address this, methods such as Alibi (Press et al., 2022) and Lex (Sun et al., 2023) have been proposed, offering solutions that mitigate the extrapolation challenges and extend the model's ability to handle longer sequences effectively.

The Transformer architecture has since become the backbone of many cutting-edge models in natural language processing (NLP), including BERT (Devlin et al., 2019), which specializes in capturing bidirectional context for tasks like question answering and sentence classification, GPT (Radford et al., 2018), known for its autoregressive approach and impressive language generation capabilities, and T5 (Raffel et al., 2020), a versatile model designed to handle a wide range of NLP tasks using a unified text-to-text framework.

1.5.2 Pre-trained Language Representations

Pre-trained language representations have revolutionized natural language processing (NLP) by providing versatile, general-purpose embeddings that encapsulate rich linguistic and contextual knowledge. These representations serve as a robust foundation for a wide range of downstream tasks, including classification, translation, summarization, and question answering. Trained on massive corpora of unlabeled text using self-supervised learning objectives, they capture intricate patterns in syntax, semantics, and context. After pre-training, these representations can be fine-tuned for specific tasks or leveraged directly through prompting, dramatically reducing the dependency on labeled data and computational resources for task-specific applications. This paradigm has not only streamlined the development of NLP systems but also broadened their applicability across diverse languages and domains.

Static Representations

The emergence of static word embeddings, such as Word2Vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2017), marked a significant milestone in the evolution of natural language processing (NLP), paving the way for deep learning applications in the field. These embeddings provide dense vector representations of words, capturing semantic relationships and similarities based on their distributional properties in large text corpora. Unlike traditional one-hot or bag-of-words representations, static embeddings encode words into fixed-length vectors that reflect their contextual usage, enabling models to perform more effectively in a variety of NLP tasks, such as sentiment analysis, information retrieval, and machine translation.

Static embeddings are trained on large text corpora using techniques that leverage the distributional hypothesis: the idea that words occurring in similar contexts tend to have similar meanings. Word2Vec (Mikolov et al., 2013a), for example, employs neural network-based methods, including the Continuous Bag-of-Words (CBOW) and Skip-Gram models, as illustrated in Figure 1.5.

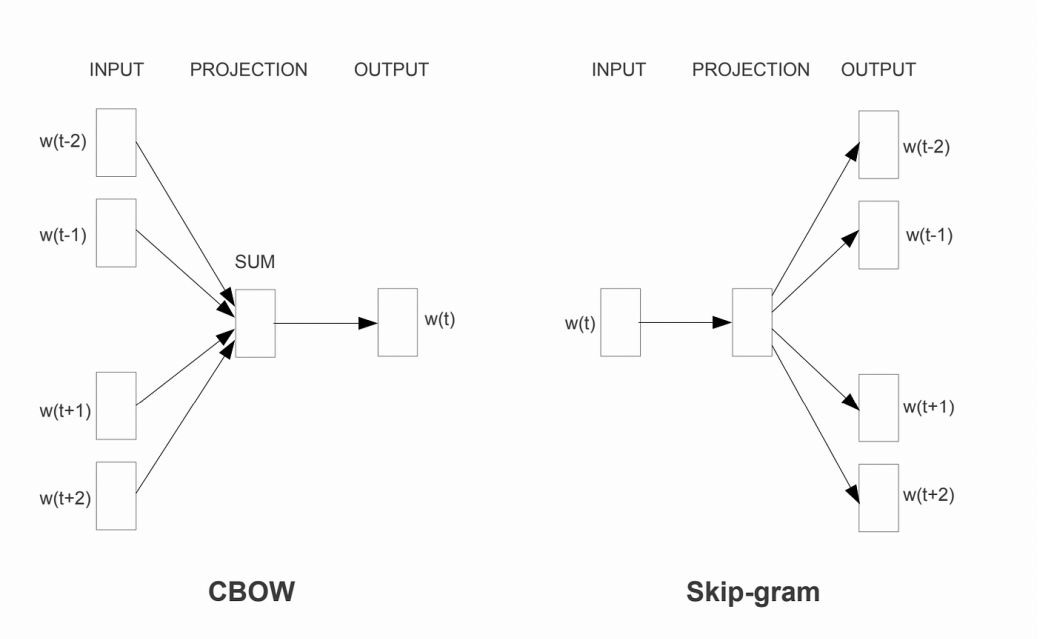


Figure 1.5: Two model architectures proposed by Mikolov et al. (2013a). The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word. The figure was taken from Mikolov et al. (2013a).

In the CBOW model, the objective is to predict a target word w_t given its surrounding context words $w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+k}$. The optimization goal is:

$$\mathcal{L}_{\text{CBOW}} = -\sum_{t=1}^{T} \log P(w_t \mid w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+k}), \tag{1.6}$$

where k is the size of the context window and T is the total number of words in the corpus.

In the Skip-Gram model, the task is reversed, aiming to predict the context words given a target word. The objective function is:

$$\mathcal{L}_{\text{Skip-Gram}} = -\sum_{t=1}^{T} \sum_{j=-k, j\neq 0}^{k} \log P(w_{t+j} \mid w_t). \tag{1.7}$$

Both architectures rely on the softmax function for estimating probabilities. During training, they optimize the embeddings such that semantically similar words appear closer in the resulting vector space. For example, words like “king" and “queen" or “cat" and “dog" tend to cluster together, reflecting their shared semantic properties. These embeddings have proven invaluable for numerous NLP tasks, forming a foundation for more advanced language representation models.

GloVe (Global Vectors for Word Representation) takes a different approach by constructing a co-occurrence matrix X that counts how frequently words i and j appear together in a fixed window of context. The embeddings are learned by minimizing the following loss function:

$$\mathcal{L}_{\text{GloVe}} = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^\top w_j + b_i + b_j - \log X_{ij} \right)^2, \tag{1.8}$$

where w_i and w_j are the word vectors for words i and j, b_i and b_j are their respective biases, X_{ij} is the co-occurrence count, and $f(X_{ij})$ is a weighting function to balance the influence of frequent and rare word pairs.

fastText, an extension of Word2Vec, introduces subword information by representing each word as a composition of character n-grams. For a word w, the embedding is computed as the sum of its subword n-gram embeddings:

$$\mathbf{v}_w = \sum_{g \in G(w)} \mathbf{v}_g, \tag{1.9}$$

where $G(w)$ represents the set of character n-grams in w, and $\mathbf{v}_g$ is the embedding for each n-gram g. This approach enables fastText to model morphological features effectively and handle rare or out-of-vocabulary (OOV) words.

While these methods revolutionized NLP by enabling pre-trained embeddings to transfer knowledge to various tasks, static embeddings have inherent limitations. Each word is assigned a single fixed vector, regardless of its meaning in different contexts. For example, the word “bank" is represented identically whether it refers to a financial institution or the side of a river. This lack of context-awareness spurred the development of dynamic, context-sensitive embeddings, as seen in modern transformer-based models. Nonetheless, the simplicity, efficiency, and effectiveness of static embeddings ensure their continued relevance in resource-constrained settings and as foundational tools in the history of NLP.

Contextualized Representations

The scalability of the Transformer architecture has revolutionized language modeling, transitioning from static word embeddings to advanced contextual language models. Pioneering models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), GPT (Radford et al., 2018), T5 (Raffel et al., 2020), and BART (Lewis et al., 2020) have significantly enhanced the ability to capture and represent nuanced linguistic information. These models mark a significant departure from earlier, static embedding techniques such as Word2Vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014), which assigned fixed representations to words, and instead generate dynamic embeddings based on context. The Transformer

architecture's self-attention mechanism allows it to model long-range dependencies, overcoming the limitations of previous recurrent models that struggled with capturing distant relationships between tokens.

These state-of-the-art language models are broadly categorized into three primary architectures: encoder-only models (e.g., BERT, RoBERTa) are optimized for understanding tasks, decoder-only models (e.g., GPT) are primarily used for generative tasks, and encoder-decoder models (e.g., T5, BART) are versatile in handling tasks that require both understanding and generation, such as translation or summarization.

The backbone of training these large-scale language models is self-supervised learning, which enables the models to learn linguistic representations directly from vast corpora of unlabeled text. Unlike supervised learning, which requires labeled data for training, self-supervised learning generates its own supervisory signal by constructing predictive tasks from raw text. Two widely adopted self-supervised objectives in this domain are Masked Language Modeling (MLM) and Causal Language Modeling (CLM).

Masked Language Modeling (MLM) aims to predict certain tokens in a sequence that are randomly masked. Given an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, a subset of tokens $\mathbf{m} \subset \mathbf{x}$ is replaced with a special `[MASK]` token, resulting in $\mathbf{x}_{\text{masked}}$. The model is trained to predict the masked tokens based on their context. The MLM objective is formally defined as:

$$\mathcal{L}_{\text{MLM}} = -\mathbb{E}_{\mathbf{x} \sim D} \sum_{i \in \mathcal{M}} \log P(x_i \mid \mathbf{x}_{\text{masked}}) \tag{1.10}$$

where $\mathcal{M}$ is the set of indices of the masked tokens, $\mathbf{x}_{\text{masked}}$ is the input sequence with masked tokens, $P(x_i \mid \mathbf{x}_{\text{masked}})$ is the predicted probability of the original token x_i given the masked sequence. The MLM objective enables the model to learn bidirectional contextual representations by considering both the left and right context of the masked tokens.

Causal Language Modeling (CLM), also referred to as autoregressive language modeling, predicts the next token in a sequence based on all previous tokens. Unlike MLM, CLM uses a unidirectional context, making it suitable for generative tasks. Given an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the CLM objective is defined as:

$$\mathcal{L}_{\text{CLM}} = -\mathbb{E}_{\mathbf{x} \sim D} \sum_{i=1}^{n} \log P(x_i \mid x_1, x_2, \ldots, x_{i-1}) \tag{1.11}$$

where $P(x_i \mid x_1, x_2, \ldots, x_{i-1})$ is the predicted probability of the i-th token given its preceding tokens. The CLM objective is commonly used in models designed for text generation, such as GPT-like architectures.

MLM provides bidirectional contextual understanding, making it well-suited for tasks requiring a holistic view of the sequence (e.g., BERT). CLM models the sequence in a unidirectional manner, which aligns with tasks like text generation (e.g., GPT). These self-supervised objectives form the foundation of modern language modeling, enabling models to generalize effectively across diverse downstream tasks.

Typically, these language models undergo a two-stage training process. First, they are pre-trained on massive corpora of unlabeled text to learn general-purpose language representations. During this pre-training phase, the models are exposed to diverse linguistic patterns, allowing them to develop a deep understanding of language structure and meaning. Once pre-training is completed, the models undergo a second phase, fine-tuning, where they are trained on labeled datasets for specific downstream tasks, such as sentiment analysis, question answering, or summarization. Fine-tuning allows the model to specialize and optimize its performance on particular tasks by adjusting the parameters learned during pre-training.

While the two-phase approach of pre-training and fine-tuning has proven effective, it can also introduce a mismatch between the distribution of data encountered during pre-training and the task-specific data encountered during fine-tuning. This mismatch may limit the model's ability to generalize across tasks, particularly when the downstream task is significantly different from the pre-training data. To address this challenge, recent advancements in prompt-based learning and few-shot learning have emerged, enabling models to perform well on new tasks with minimal task-specific training data.

Large Language Model Alignment

To address the challenges of task adaptability and data efficiency, newer models such as GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020), and GPT-3 (Brown et al., 2020) adopt innovative prompting techniques. As shown in Figure 1.6, these techniques exploit the model's pre-trained capabilities by framing tasks as text completions, enabling zero-shot or few-shot learning directly from task descriptions and a small number of examples. This eliminates the need for extensive task-specific fine-tuning or parameter updates, allowing the models to perform a wide range of tasks with minimal supervision. Such approaches not only mitigate the pre-training-fine-tuning mismatch but also significantly reduce the reliance on labeled data, making these models more adaptable to low-resource settings and diverse real-world applications.

Despite their impressive capabilities, PLMs can exhibit undesirable behaviors, such as generating fabricated information (hallucinations), producing biased or toxic outputs, or failing to adhere to user instructions (Bender et al., 2021;

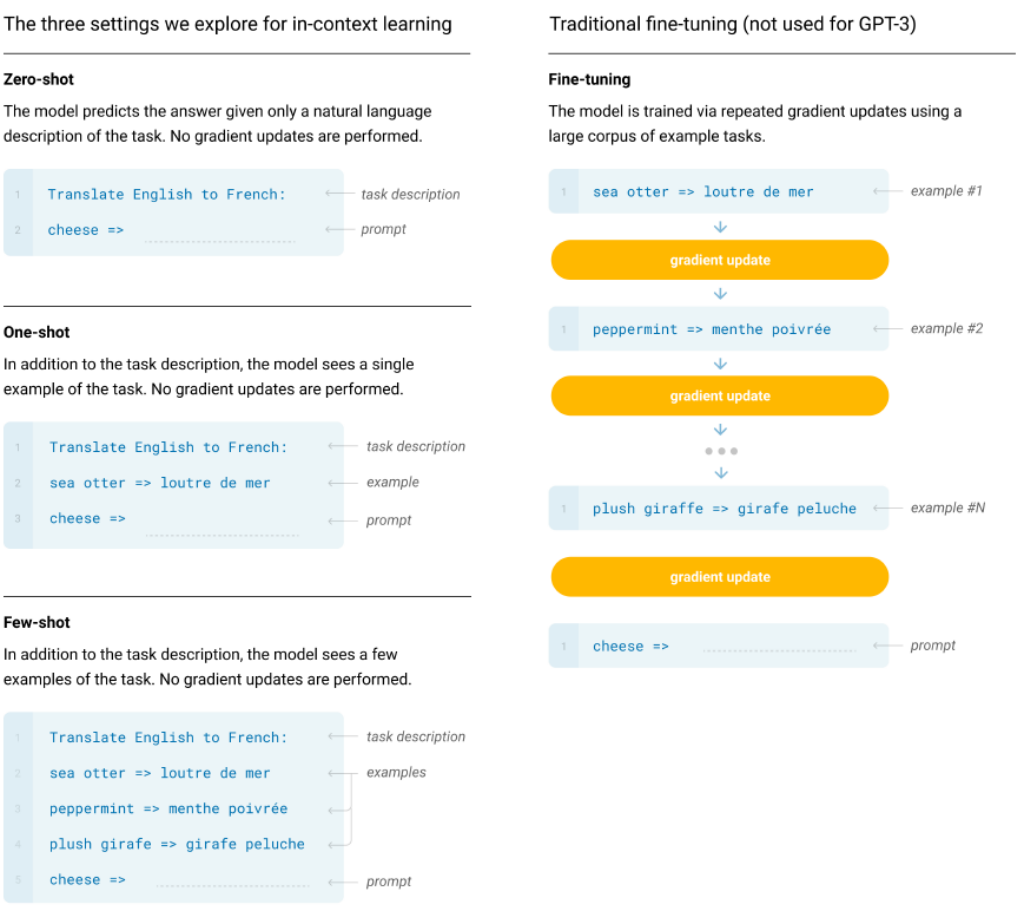


Figure 1.6: Zero-shot, one-shot and few-shot, contrasted with traditional fine-tuning. The panels above show four methods for performing a task with a language model – fine-tuning is the traditional method, whereas zero-, one-, and few-shot, which were proposed in Brown et al. (2020), require the model to perform the task with only forward passes at test time. The figure was taken from Brown et al. (2020).

Bommasani et al., 2021; Kenton et al., 2021). These issues present significant challenges, especially in high-stakes applications like healthcare, legal services, or education, where reliability, fairness, and ethical considerations are paramount. To mitigate these problems, instruction tuning (Wang et al., 2022b) has been introduced, helping models align more closely with human intentions and enabling them to better follow explicit instructions in task-oriented and conversational contexts. Instruction tuning enhances the models' ability to handle a broader range of queries while adhering to user-defined guidelines.

Concurrently, advancements in model architecture and scaling have led to the creation of even larger and more powerful models. Notable examples include Gopher (Rae et al., 2021) with 280 billion parameters and Megatron-Turing NLG

(Smith et al., 2022) with 530 billion parameters. These advancements have been complemented by research emphasizing the importance of scaling not only model size but also data size, as demonstrated by models like Chinchilla (Hoffmann et al., 2022), which optimizes both factors for superior performance (Kaplan et al., 2020; Hoffmann et al., 2022).

These innovations have spurred the development of a new generation of large language models (LLMs) (Zhao et al., 2023), exemplified by models such as PaLM (Chowdhery et al., 2022; Anil et al., 2023b), Gemini (Anil et al., 2023a), LLaMA (Touvron et al., 2023a,b; Dubey et al., 2024), Gemma (Rivière et al., 2024), Qwen (Bai et al., 2023), Mistral (Jiang et al., 2023), Claude 2,[3] and ChatGPT.[4] These models not only deliver exceptional performance across a wide range of tasks but also exhibit emergent abilities, such as understanding nuanced queries, reasoning across multiple steps, and generalizing to novel tasks with minimal supervision (Wei et al., 2022; Lu et al., 2023b). These emergent abilities mark a transformative shift in the capabilities of LLMs, making them increasingly integral to advancing research and applications in NLP.

PLMs represent a transformative leap in NLP, laying the foundation for unprecedented innovation in the field. By harnessing vast amounts of data and leveraging advanced architectures, PLMs have significantly improved the ability to understand and generate human language. This advancement not only enhances the performance of a wide range of NLP tasks—such as text classification, translation, and summarization—but also broadens accessibility across diverse languages and domains. Their adaptability allows for rapid fine-tuning and customization to meet the needs of various industries, opening up new possibilities for applications in healthcare, education, entertainment, and beyond.

1.5.3 Multilingual Representation

With over 7,000 languages spoken worldwide, supporting diverse languages in NLP is crucial to fostering inclusivity and ensuring accessibility for all. Multilingual representation in NLP aims to develop models that can understand, generate, and transfer knowledge across numerous languages. This capability is particularly important as most languages are significantly underrepresented in current NLP tools and resources. By building models that effectively manage multilingual contexts, NLP systems can democratize access to technology, break down language barriers, and promote seamless global communication and information exchange.

[3] https://web.archive.org/web/20230811143536/https://www.anthropic.com/index/claude-2

[4] https://openai.com/blog/chatgpt

mapping.png

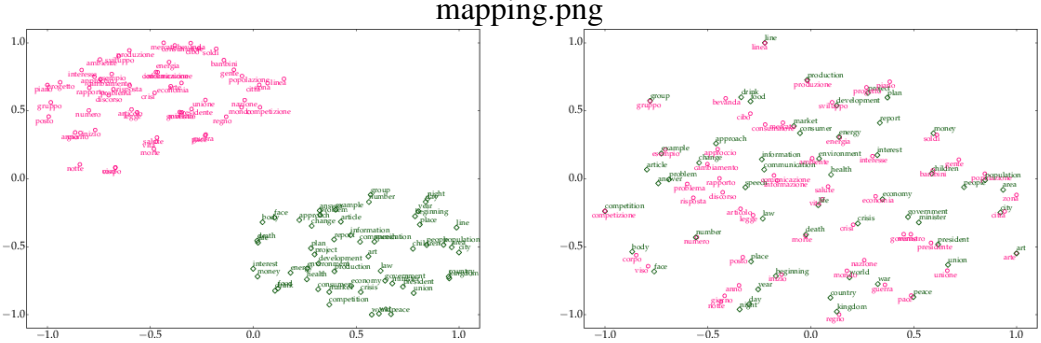


Figure 1.7: Cross-lingual mapping: Unaligned monolingual word embeddings (left) and word embeddings projected into a shared cross-lingual embedding space (right). Embeddings are visualized with t-SNE. The figure was taken from Ruder et al. (2019).

Cross-Lingual Mapping

As illustrated in Figure 1.7, cross-lingual mapping seeks to align the representations of words or sentences from different languages into a unified semantic space. This alignment is crucial for enabling seamless knowledge transfer across languages, supporting tasks such as cross-lingual information retrieval, machine translation, and low-resource language modeling. Approaches to cross-lingual mapping are generally divided into two main categories: supervised methods, which rely on parallel data for training, and unsupervised methods, which do not require explicit alignment between languages.

Supervised approaches rely on bilingual dictionaries or parallel corpora to explicitly learn mappings between source and target languages. One commonly employed technique is linear mapping (Mikolov et al., 2013b), where monolingual word embeddings from each language are aligned using a linear transformation. Let X and Y denote the embedding matrices of the source and target languages, respectively. The mapping is learned by solving:

$$W = \arg\min_{W} \|WX - Y\|_F, \tag{1.12}$$

where W is the transformation matrix and $\|\cdot\|_F$ represents the Frobenius norm. This optimization problem is a specific instance of the Procrustes problem, which aims to find the optimal linear transformation that aligns two embedding spaces. Mikolov et al. (2013b) employed a gradient descent method to solve this optimization. An extension of this approach constrains the transformation matrix W to be orthonormal, i.e., $W^T W = I$ (Xing et al., 2015).

Although supervised methods yield promising results when high-quality bilingual dictionaries are available, constructing these dictionaries can be a challenging task for many language pairs. To address this issue, unsupervised approaches

(Artetxe et al., 2018; Lample et al., 2018) eliminate the need for parallel data by exploiting structural similarities within monolingual embedding spaces. These methods frequently utilize adversarial training to align embeddings, minimizing domain divergence loss.

By aligning linguistic representations across languages, cross-lingual mapping helps bridge linguistic barriers, thereby extending the reach of NLP to diverse linguistic and cultural contexts. However, several challenges remain. The effectiveness of these mappings is often influenced by the degree of similarity between embedding spaces, which can vary significantly across languages with different scripts, syntactic structures, or linguistic families. Additionally, mappings tend to struggle with rare words, domain-specific terminology, and languages with limited training data. Addressing these challenges is crucial for improving the robustness and generalizability of cross-lingual mapping techniques.

Joint Training

Recently, joint training has emerged as the dominant approach in multilingual representation learning, demonstrated by both smaller multilingual language models, such as multilingual BERT (Devlin et al., 2019) supporting 104 languages and XLM-R (Conneau et al., 2020) covering 100 languages, as well as larger multilingual models like BLOOM (Scao et al., 2022) for 46 languages and Aya (Üstün et al., 2024) supporting 100 languages.

In addition, some multilingual language models are designed specifically for certain language groups. For African languages, models include AfriBERTa (Ogueji et al., 2021), KinyaBERT (Nzeyimana and Rubungo, 2022), AfroXLMR (Alabi et al., 2022a), SERENGETI (Adebara et al., 2022), Cheetah (Adebara et al., 2024), and AfriMT (Adelani et al., 2022). CINO (Yang et al., 2022) is designed for Chinese languages, while IndicBERT (Kakwani et al., 2020) focuses on Indian languages. For Indonesian languages, IndoBERT (Koto et al., 2020) and IndoGPT (Cahyawijaya et al., 2021) have been developed. Additionally, Tower (Alves et al., 2024) and EuroLLM (Martins et al., 2024) cater to European languages.

In this approach, raw text from multiple languages is aggregated during pretraining, enabling the models to learn shared cross-lingual representations. This method allows for the explicit alignment of representations across languages into a common semantic space, where similar concepts and linguistic structures are captured across diverse languages. By leveraging extensive multilingual corpora, these models can transfer knowledge across languages, effectively understanding and representing shared ideas, emotions, and expressions in a wide range of linguistic contexts.

Joint training allows language models to learn language-agnostic features while retaining language-specific nuances. Recent studies (Müller et al., 2021b; Chang

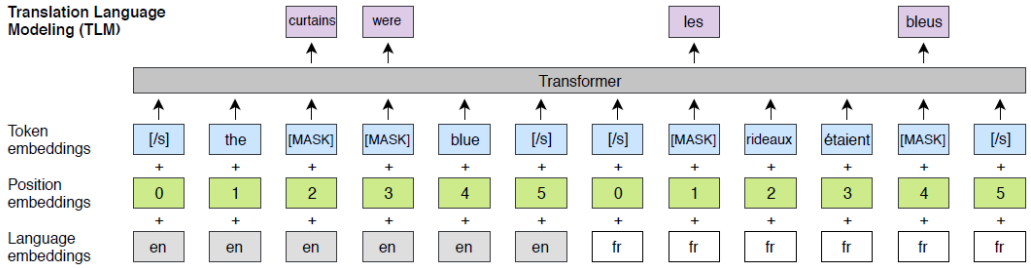


Figure 1.8: Translation Language Model (TLM). This model extends the traditional Masked Language Model (MLM) to parallel sentence pairs. For instance, when predicting a masked English or French token, the model attends to both the English sentence and its French translation, aligning embeddings between the two languages. Position embeddings in the target sentence are reset to facilitate this alignment. The figure was taken from (Conneau and Lample, 2019).

et al., 2022) show that multilingual models effectively encode both language-specific information (Gerz et al., 2018; Wu and Dredze, 2019; Rama et al., 2020; Choenni and Shutova, 2022; Liang et al., 2021; Choenni et al., 2023) and language-neutral knowledge, such as syntactic structures (Chi et al., 2020), token frequencies (Rajaee and Pilehvar, 2022), and representation shifts (Libovický et al., 2020; Pires et al., 2019). This enables multilingual language models to tackle a range of tasks, including machine translation, cross-lingual transfer, and multilingual content generation, laying the foundation for more inclusive and effective NLP systems. This approach has proven particularly beneficial for low-resource languages, allowing them to leverage data from high-resource languages to improve their performance.

Multilingual language models have a wide range of applications, including translation, information retrieval, sentiment analysis, and education. By advancing multilingual representation, NLP can help build a more inclusive technological ecosystem that bridges linguistic divides and meets the needs of a global, multilingual population.

Parallel Corpora Exploitation

Parallel corpora, which consist of aligned sentences or documents across multiple language pairs, are essential for advancing cross-lingual consistency and improving translation accuracy. Their use significantly enhances the performance of multilingual language models by providing critical resources for training on diverse language pairs (Conneau and Lample, 2019; Patra et al., 2022).

An effective approach for leveraging parallel corpora is the Translation Language Model (TLM), introduced by Conneau and Lample (2019). As illustrated in Figure 1.8, TLM extends the traditional Masked Language Model (MLM) by

incorporating parallel sentence pairs during pre-training. Specifically, given a pair of sentences in source and target languages, TLM concatenates both sentences and masks tokens within each sentence. The model then predicts the masked tokens using context from both sentences, encouraging the alignment of token embeddings across languages. This approach enables the model to learn shared cross-lingual representations by aligning linguistic structures between source and target languages.

Since parallel corpora are often scarce, especially for low-resource languages, bilingual lexicons covering a wide range of languages can be utilized to generate synthetic parallel data (Wang et al., 2022a). This strategy enhances multilingual models by allowing them to leverage available resources, boosting performance in languages with limited training data. Additionally, multilingual sentence embedding models, such as LASER (Artetxe and Schwenk, 2019), LABSE (Feng et al., 2022), and LASER3 (Heffernan et al., 2022), can be employed for bitext mining to further improve the alignment of cross-lingual representations.

By harnessing the power of parallel corpora, multilingual models can generalize more effectively across diverse languages. This ability is essential for enhancing the performance of various NLP tasks, including machine translation, cross-lingual information retrieval, and content generation. All of these tasks benefit from models that understand and represent multiple languages within a unified semantic framework.

Language Adaptation

Many languages have limited data or are underrepresented during the pre-training phase of multilingual language models, which hinders their performance in these languages. Additionally, the curse of multilinguality (Conneau et al., 2020; Wang et al., 2020b; Chang et al., 2023) can further degrade the model's performance, as adding more languages to the training data often leads to a deterioration in the accuracy of the model on individual languages. To overcome these challenges, researchers have explored various approaches to adapt existing multilingual models for better performance on specific languages.

A widely studied technique for adapting pre-trained multilingual models to specific tasks or languages is the use of adapters (Pfeiffer et al., 2020; Üstün et al., 2020; Nguyen et al., 2021; Faisal and Anastasopoulos, 2022; Yong et al., 2022; Alves et al., 2023). Adapters are lightweight modules inserted into the layers of a pre-trained model, allowing it to adapt to new tasks or languages with minimal retraining. This method captures language-specific nuances, improves performance on low-resource languages, and reduces the computational cost of adaptation. The modular nature of adapters is particularly appealing for adapting large, complex models to diverse linguistic contexts.

A prominent adapter technique for efficient model adaptation is Low-Rank Adaptation (LoRA) (Hu et al., 2022). LoRA offers an efficient way to adapt large pre-trained language models, especially in low-resource settings. Rather than adjusting all parameters of the model, LoRA introduces low-rank matrices into each layer, fine-tuning these matrices while keeping the remaining model parameters fixed. This significantly reduces the number of parameters that need to be updated, making the adaptation process computationally more efficient.

The key idea behind LoRA is to factorize the weight matrix W of a neural network layer into two low-rank matrices A and B:

$$W' = W + \Delta W = W + A \cdot B \tag{1.13}$$

where W' represents the adapted weight matrix, W is the original pre-trained weight matrix, A and B are low-rank matrices fine-tuned during adaptation. By introducing A and B, LoRA ensures that only a small number of parameters are updated, thus reducing the computational cost associated with model adaptation. The rank of A and B is typically much smaller than W, making the adaptation process even more efficient.

Another approach to adapting multilingual language models to low-resource languages is vocabulary extension or substitution (Chau et al., 2020; Wang et al., 2020a; Müller et al., 2020, 2021a; Pfeiffer et al., 2021; Chen et al., 2023; Downey et al., 2023). This approach involves modifying or expanding the model's vocabulary to better accommodate the linguistic features of low-resource languages. Moreover, Alabi et al. (2022b) propose removing vocabulary tokens from the embedding layer that correspond to non-African writing scripts before applying adapters. This approach reduces the model size, making it more efficient for adapting existing multilingual language models to African languages.

Vocabulary extension involves adding new tokens to the model's vocabulary, which helps address out-of-vocabulary (OOV) issues and ensures better representation of previously unseen words or concepts. This process enables the model to effectively handle new tokens and improve its performance on low-resource languages.

On the other hand, vocabulary substitution focuses on replacing existing tokens in the model's vocabulary with those from low-resource languages, often through a process of cross-lingual token alignment. This method ensures smoother adaptation by aligning the model's vocabulary with the linguistic characteristics of the target language, improving its capacity to generate and understand language-specific tokens.

In the era of large language models, in-context learning has emerged as a powerful approach, enabling models to efficiently adapt to low-resource languages. By incorporating key contextual information, models can perform tasks more ef-

fectively in these languages. One such piece of information is the use of few-shot examples in the target language, which helps guide the model's predictions. Translation-based evaluation leverages pre-trained translation models (Shi et al., 2022; Ahuja et al., 2023) or self-translation techniques (Huang et al., 2023; Etxaniz et al., 2023; Zhang et al., 2023) to provide translations between high-resource and low-resource languages. Additionally, bilingual lexicons and grammar books offer valuable linguistic resources, further enhancing model performance in low-resource settings (Lu et al., 2023a; Tanzer et al., 2024; Zhang et al., 2024b,a).

Overall, these strategies help mitigate the challenges posed by the curse of multilinguality, where the performance of multilingual models declines as more languages are added to the training set. By focusing on techniques that adapt, extend, and augment models for low-resource languages, these approaches show great promise in advancing multilingual NLP applications. Ultimately, they contribute to the development of more equitable and inclusive models that can better support a diverse range of languages.

1.6 Conclusion and Future Work

1.6.1 Conclusion

This thesis presents significant advancements in both the modeling and adaptation of massively multilingual language models, with a particular emphasis on reducing the performance gap between high- and low-resource languages. Its contributions span the entire development pipeline—from data collection and model architecture to adaptation and evaluation—culminating in a cohesive framework for building more equitable and linguistically inclusive multilingual systems.

Through the Glot500, MaLA500, and EMMA500 series, this work introduces comprehensive multilingual corpora and progressively scaled models that achieve state-of-the-art performance across diverse linguistic settings. An in-depth analysis of parallel corpora reveals how the interaction among data quality, data quantity, and training objectives fundamentally determines the effectiveness of multilingual language models.

In the area of task adaptation, the mPLM-Sim and XAMPLER studies propose complementary strategies to strengthen cross-lingual generalization. mPLM-Sim introduces a model-based language similarity metric to identify optimal source languages for transfer learning, enabling more effective adaptation to underrepresented languages. In parallel, XAMPLER fine-tunes a cross-lingual retriever to supply high-quality in-context examples, substantially enhancing in-context learning performance in low-resource scenarios.

Overall, this thesis demonstrates that the integration of high-quality multilin-

gual data, principled transfer mechanisms, and data-driven adaptation techniques can markedly reduce disparities between high- and low-resource languages. These advances pave the way for the next generation of multilingual language models—systems that not only achieve linguistic parity but also embody a deeper understanding of global cultural diversity, fostering more inclusive and representative language technologies.

1.6.2 Future Work

To reach the next milestone—developing large language models with capabilities for low-resource languages comparable to those for English—several critical challenges must be addressed.

Model Architecture and Scalability Designing an effective architecture is crucial for developing multilingual language models that can support both high- and low-resource languages. A well-balanced architecture must distribute model capacity equitably across languages, enabling effective cross-lingual knowledge transfer while mitigating overfitting to high-resource data. Integrating interpretability tools further enhances transparency, allowing researchers to analyze internal representations, track transfer dynamics, and evaluate language-specific performance.

Certain architectural approaches are particularly well-suited for multilingual settings. The Mixture-of-Experts (MoE) framework offers a scalable and efficient solution by dynamically routing inputs through specialized expert networks, thereby allocating computational capacity where it is most beneficial. Meanwhile, tokenizer-free models provide a flexible alternative for handling low-resource languages, removing the constraints of fixed vocabularies and facilitating seamless extension to previously unseen linguistic systems.

High-Quality Multilingual Data High-quality pretraining for multilingual language models requires datasets that balance quantity and quality, particularly for low-resource languages. Large-scale web-crawled corpora often underrepresent these languages, and noisy or unverified data can amplify biases and errors. To address this, a combination of human curation, LLM-based data generation, and structured linguistic resources is essential.

Human-curated data remains the most reliable source, ensuring linguistic authenticity and cultural fidelity. Community-driven collection and validation, especially involving native speakers, can expand coverage while preserving language-specific nuances. At the same time, LLM-generated data can supplement scarce resources: models can produce candidate text, which is then verified or refined

by humans. Additionally, seed resources such as lexicons, grammar rules, and parallel corpora can guide both human and LLM-based synthesis, ensuring the resulting datasets are linguistically grounded and culturally appropriate.

By strategically combining these approaches—human expertise, LLM distillation, and structured linguistic resources—researchers can create comprehensive, high-quality multilingual datasets that support robust pretraining and effective cross-lingual transfer.

Cross-Lingual Transfer Cross-lingual transfer can be divided into semantic-level transfer and solution-level transfer. Semantic-level transfer involves sharing universal knowledge—such as general world facts, common reasoning patterns, and syntactic structures—that can be applied across multiple languages. In contrast, solution-level transfer focuses on adapting knowledge that is culturally, regionally, or legally specific, reflecting the unique characteristics of each language and context.

To facilitate semantic-level transfer, techniques like cross-lingual retrieval and machine translation are critical for bridging data gaps, particularly for low-resource languages. For solution-level transfer, leveraging reinforcement learning on high-quality curated data allows models to learn how to solve tasks effectively in one context and generalize that capability across languages.

Multimodal Grounding Incorporating multimodal data—text, speech, images, and video—provides richer contextual grounding, particularly for languages with limited textual resources. Speech data supports languages lacking standardized orthography, while visual and environmental cues enhance semantic understanding and cross-lingual reasoning.

Multimodal integration strengthens learning signals for low-resource languages and offers a universal medium to bridge linguistic divides, supporting more equitable multilingual capabilities without redundancy in cultural or localization considerations.

By addressing these challenges—scalable architectures, high-quality multilingual data, cross-lingual transfer, and multimodal grounding—massively multilingual language models can achieve more equitable support across languages, fostering a globally inclusive and accessible technological ecosystem.

Chapter 2

Glot500: Scaling Multilingual Corpora and Language Models to 500 Languages

Glot500: Scaling Multilingual Corpora and Language Models to 500 Languages

Ayyoob Imani[*1,2], **Peiqin Lin**[*1,2], **Amir Hossein Kargaran**[1,2], **Silvia Severini**[1], **Masoud Jalili Sabet**[1], **Nora Kassner**[1,2], **Chunlan Ma**[1,2], **Helmut Schmid**[1], **André F. T. Martins**[3,4,5], **François Yvon**[6] **and Hinrich Schütze**[1,2]
[1]CIS, LMU Munich, Germany [2]Munich Center for Machine Learning (MCML), Germany
[3]Instituto Superior Técnico (Lisbon ELLIS Unit) [4]Instituto de Telecomunicações
[5]Unbabel [6]Sorbonne Université, CNRS, ISIR, France
{ayyoob, linpq, amir, silvia}@cis.lmu.de

Abstract

The NLP community has mainly focused on scaling Large Language Models (LLMs) *vertically*, i.e., making them better for about 100 languages. We instead scale LLMs *horizontally*: we create, through continued pretraining, Glot500-m, an LLM that covers 511 predominantly low-resource languages. An important part of this effort is to collect and clean Glot500-c, a corpus that covers these 511 languages and allows us to train Glot500-m. We evaluate Glot500-m on five diverse tasks across these languages. We observe large improvements for both high-resource and low-resource languages compared to an XLM-R baseline. Our analysis shows that no single factor explains the quality of multilingual LLM representations. Rather, a combination of factors determines quality including corpus size, script, "help" from related languages and the total capacity of the model. Our work addresses an important goal of NLP research: we should not limit NLP to a small fraction of the world's languages and instead strive to support as many languages as possible to bring the benefits of NLP technology to all languages and cultures. Code, data and models are available at https://github.com/cisnlp/Glot500.

1 Introduction

The NLP community has mainly focused on scaling Large Language Models (LLMs) *vertically*, i.e., deepening their understanding of high-resource languages by scaling up parameters and training data. While this approach has revolutionized NLP, the achievements are largely limited to high-resource languages. Examples of "vertical" LLMs are GPT3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022) and Bloom (BigScience et al., 2022). In this paper, we create Glot500-m, a model that instead focuses on scaling multilingual LLMs *horizontally*, i.e., scaling to a large number of languages the great majority of which is low-resource. As LLMs are essential for progress in NLP, lack of LLMs supporting low-resource languages is a serious impediment to bringing NLP to all of the world's languages and cultures. Our goal is to address this need with the creation of Glot500-m.[1]

Existing multilingual LLMs support only about 100 (Conneau et al., 2020) out of the 7000 languages of the world. These supported languages are the ones for which large amounts of training data are available through projects such as Oscar (Suárez et al., 2019) and the Wikipedia dumps.[2] Following Siddhant et al. (2022), we refer to the 100 languages covered by XLM-R (Conneau et al., 2020) as **head languages** and to the remaining languages as **tail languages**. This terminology is motivated by the skewed distribution of available data per language: for the best-resourced languages there are huge corpora available, but for the long tail of languages, only small corpora exist. This is a key problem we address: the availability of data for tail languages is limited compared to head languages. As a result, tail languages have often been ignored by language technologies (Joshi et al., 2020).

Although there exists some work on machine translation for a large number of tail languages (Costa-jussà et al., 2022; Bapna et al., 2022), existing LLMs for tail languages are limited to a relatively small number of languages (Wang et al., 2019; Alabi et al., 2022; Wang et al., 2022). In this paper, we address this gap. Our work has three parts. (i) **Corpus collection.** We collect Glot2000-c, a corpus covering thousands of tail languages. (ii) **Model training.** Using Glot500-c, a subset of Glot2000-c, we train Glot500-m, an LLM covering 511 languages. (iii) **Validation.** We conduct an extensive evaluation of the quality of Glot500-m's

[*]Equal contribution.

[1]In concurrent work, Adebara et al. (2022) train a multilingual model for 517 African languages on a 42 gigabyte corpus, but without making the model available.

[2]https://dumps.wikimedia.org/

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 1082–1117

representations of tail languages on a diverse suite of tasks.

In more detail, **corpus collection** considers three major sources: websites that are known to publish content in specific languages, corpora with classified multilingual content and datasets published in specific tail languages. The resulting dataset Glot2000-c comprises 700GB in 2266 languages collected from ≈150 sources. After cleaning and deduplication, we create the subset Glot500-c, consisting of 511 languages and 534 *language-scripts* (where we define a language-script as a combination of ISO 639-3[3] and script) to train Glot500-m. Our criterion for including a language-script in Glot500-c is that it includes more than 30,000 sentences.

Model training. To train Glot500-m, we employ vocabulary extension and continued pretraining. XLM-R's vocabulary is extended with new tokens trained on Glot500-c. We then perform continued pretraining of XLM-R with the MLM objective (Devlin et al., 2019).

Validation. We comprehensively evaluate Glot500-m on a diverse suite of natural language understanding, sequence labeling and multilingual tasks for hundreds of languages. The results demonstrate that Glot500-m performs better than XLM-R-B (XLM-R-base) for tail languages by a large margin while performing comparably (or better) for head languages.

Previous work on multilinguality has been hindered by the lack of LLMs supporting a large number of languages. This limitation has led to studies being conducted in settings dissimilar from real-world scenarios. For example, Dufter and Schütze (2020) use synthetic language data. And the curse of multilinguality has been primarily studied for a set of high-resource languages (Conneau et al., 2020). By creating Glot500-m, we can investigate these issues in a more realistic setting. We make code, data and trained models available to foster research by the community on how to include hundreds of languages that are currently ill-served by NLP technology.

Contributions. (i) We train the multilingual model Glot500-m on a 600GB corpus, covering more than 500 diverse languages, and make it publicly available at `https://github.com/cisnlp/Glot500`. (ii) We collect and clean Glot500-c, a corpus that covers these diverse languages and allows us to train Glot500-m, and will make as much of it publicly available as possible. (iii) We evaluate Glot500-m on pseudoperplexity and on five diverse tasks across these languages. We observe large improvements for low-resource languages compared to an XLM-R baseline. (iv) Our extensive analysis shows that no single factor explains the quality of multilingual LLM representations. Rather, a combination of factors determines quality including corpus size, script, "help" from related languages and the total capacity of the model. (v) Our work addresses an important goal of NLP research: we should not limit NLP to a relatively small number of high-resource languages and instead strive to support as many languages as possible to bring the benefits of NLP to all languages and cultures.

2 Related Work

Training multilingual LLMs using the masked language modeling (MLM) objective is effective to achieve cross-lingual representations (Devlin et al., 2019; Conneau et al., 2020). These models can be further improved by incorporating techniques such as discriminative pre-training (Chi et al., 2022) and the use of parallel data (Yang et al., 2020; Chi et al., 2021). However, this primarily benefits a limited set of languages with large corpora.

Recent research has attempted to extend existing LLMs to languages with limited resources. Wang et al. (2019) propose vocabulary extension; Ebrahimi and Kann (2021) investigate adaptation methods, including MLM and Translation Language Model (TLM) objectives and adapters; Alabi et al. (2022) adapt XLM-R to 17 African languages; Wang et al. (2022) expand language models to low-resource languages using bilingual lexicons.

Alternatively, parameter-efficient fine-tuning adapts pre-trained models to new languages by training a small set of weights effectively (Zhao et al., 2020; Pfeiffer et al., 2021; Ansell et al., 2022). Pfeiffer et al. (2022) address the "curse of multilinguality" by sharing a part of the model among all languages and having separate modules for each language. We show that the common perception that multilinguality increases as we add more languages, until, from some point, it starts decreasing, is naive. The amount of available data per language and the similarity between languages also play important roles (§6.8).

Another approach trains LLMs from scratch for a limited number of tail languages; e.g., AfriBERTa

[3] `https://iso639-3.sil.org/code_tables/639`

(Ogueji et al., 2021a) and IndicNLPSuite (Kakwani et al., 2020) are LLMs for 11 African languages and 11 Indic languages. In concurrent work, Adebara et al. (2022) train a multilingual model for 517 African languages on a 42 GB corpus, but without making the model available and with an evaluation on a smaller number of languages than ours.

Closely related to our work on corpus creation, Bapna et al. (2022) and Costa-jussà et al. (2022) also create NLP resources for a large number of tail languages. They train a language identifier model and extract textual data for tail languages from large-scale web crawls. This approach is effective, but it requires significant computational resources and native speakers for all tail languages. This is hard to do outside of large corporations. Bapna et al. (2022) have not made their data available. Costa-jussà et al. (2022) have only released a portion of their data in around 200 languages.

A key benefit of "horizontally" scaled multilingual LLMs is transfer from high- to low-resource languages. Our evaluation suggests that Glot500-m excels at this, but this is not the main focus of our paper. There is a large body of work on crosslingual transfer: (Artetxe and Schwenk, 2019; Imani-Googhari et al., 2022; Lauscher et al., 2020; Conneau et al., 2020; Turc et al., 2021; Fan et al., 2021; Severini et al., 2022; Choenni and Shutova, 2022; Wang et al., 2023), inter alia.

3 Glot2000-c

3.1 Data Collection

One of the major challenges in developing NLP technologies for tail languages is the scarcity of high-quality training data. In this work, we propose a lightweight methodology that is easily replicable for academic labs. We identify tail language data previously published by researchers, publishers and translators and then crawl or download them. By crawling a few websites and compiling data from around 150 different datasets, we amass more than 700GB of text in 2266 languages. We will refer to these sources of data as *data sources*. Our data covers many domains, including religious texts, news articles and scientific papers. Some of the data sources are high-quality, verified by native speakers, translators and linguists. Others are less reliable such as web crawls and Wikipedia dumps. It is therefore necessary to clean the data. For a list of data sources, see §C.

3.2 Language-Scripts

Some languages are written in multiple scripts; e.g., Tajik is written in both Cyrillic and Arabic scripts. Some data sources indicate the script, but others either do not or provide mixed text in multiple scripts. We detect the script for each sentence and treat each language-script as a separate entity.

3.3 Ngram LMs and Language Divergence

We train a 3-gram character-level language model M_i for each language-script L_i, using KenLM (Heafield, 2011). We refer to the perplexity calculated for the corpus of language L_i using language model M_j as $\mathcal{PP}(M_j, L_i)$. Similar to Gamallo et al. (2017), we define a perplexity-based divergence measure of languages L_i and L_j as:

$$\mathcal{D}_{L_i,L_j} = \max\left(\mathcal{PP}(M_j, L_i), \mathcal{PP}(M_i, L_j)\right)$$

We use $\mathcal{D}$ to filter out noisy data in §3.4 and study the effect of similar languages in LLM training in §6.7 and §6.8. For more details, see §A.

3.4 Data Cleaning

To remove noise, we use chunk-level and corpus-level filters.

While some sources are sentence-split, others provide multiple sentences (e.g., a paragraph) as one chunk. Chunk-level filters process each chunk of text from a data source as a unit, without sentence-splitting. Some chunk-level filters are based on the notion of word: we use white space tokenization when possible and otherwise resort to sentencePiece (Kudo and Richardson, 2018) trained by Costa-jussà et al. (2022).

As chunk-level filters, we employ the **sentence-level filters** SF1–SF5 from BigScience ROOTS (Laurençon et al., 2022).

SF1 Character repetition. If the ratio of repeated characters is too high, it is likely that the sentence has not enough textual content.

SF2 Word repetition. A high ratio of repeated words indicates non-useful repetitive content.

SF3 Special characters. Sentences with a high ratio of special characters are likely to be crawling artifacts or computer code.

SF4 Insufficient number of words. Since training language models requires enough context, very small chunks of text are not useful.

SF5 Deduplication. If two sentences are identical after eliminating punctuation and white space, one is removed.

	langs	*scripts*	*sents'*	*median s'*
Glot2000-c	2266	35	2.3B	8K
Glot500-c	511	30	1.5B	120K
Costa-jussà et al. (2022)	134	-	2.4B	3.3M
Bapna et al. (2022)	1503	-	1.7B	25K

Table 1: Statistics for Glot2000-c, Glot500-c and existing multilingual datasets: number of languages, scripts, sentences' and median number of sentences' per language-script.

In the rest of the paper, we refer to a chunk as a **sentence'**. A sentence' can consist of a short segment, a complete sentence or a chunk (i.e., several sentences).

Corpus-level filters detect if the corpus of a language-script is noisy; e.g., the corpus is in another language or consists of non-meaningful content such as tabular data. We employ filters CF1 and CF2.

CF1 In case of **mismatch between language and script**, the corpus is removed; e.g., Chinese written in Arabic is unlikely to be Chinese.

CF2 Perplexity mismatch. For each language-script L1, we find its closest language-script L2: the language-script with the lowest perplexity divergence (§3.3). If L1 and L2 are not in the same typological family, we check L1/L2 manually and take appropriate action such as removing the corpus (e.g., if it is actually English) or correcting the ISO code assigned to the corpus.

3.5 Training Data: Glot500-c

Among the 2000+ language-scripts that we collected data for, after cleaning, most have too little data for pretraining LLMs. It is difficult to quantify the minimum amount needed for pretraining. Therefore, we pick a relatively high "safe" threshold, 30,000 sentences', for inclusion of language-scripts in model training. This allows us to train the model effectively and cover many low-resource languages. Table 1 gives Glot500-c statistics. See §B for a list of language-scripts. We train Glot500-m on Glot500-c; note that while Glot500-c focuses on tail languages, it contains some data in head languages which we include in Glot500-m training to prevent catastrophic forgetting.

We divide the corpus for each language into train/dev/test, reserving 1000 sentences' each for dev and test and using the rest for train. We pick 1000 parallel verses if we have a Bible translation and add 500 each to test and dev. These parallel verses convey identical meanings and facilitate crosslingual evaluation. We pretrain the model using only the training data.

	XLM-R-B	XLM-R-L	Glot500-m
Model Size	278M	560M	395M
Vocab Size	250K	250K	401K
Transformer Size	86M	303M	86M

Table 2: Model sizes. Glot500-m and XLM-R-B have the same transformer size, but Glot500-m has a larger vocabulary, resulting in an overall larger model.

4 Glot500-m

4.1 Vocabulary Extension

To extend XLM-R's vocabulary, we use SentencePiece (Kudo and Richardson, 2018) with a unigram language model (Kudo, 2018) to train a tokenizer with a vocabulary size of 250K on Glot500-c. We sample data from different language-scripts according to a multinomial distribution, with α=.3. The amount we sample for head languages is the same as tail languages with the lowest amount; this favors tail languages – head languages are already well learned by XLM-R. We merge the obtained tokens with XLM-R's vocabulary. About 100K new tokens were in fact old tokens, i.e., already part of XLM-R's vocabulary. We take the probabilities of the (genuinely) new tokens directly from SentencePiece. After adding the 151K new tokens to XLM-R's vocabulary (which has size 250K), the vocabulary size of Glot500-m is 401K.

We could also calculate probabilities of existing and new tokens over a mixture of original XLM-R training corpus and Glot500-c (Chung et al., 2020). For head languages, the percentage of changed tokens using the new tokenizer compared to the original tokenizer ranges from 0.2% to 50%. However, we found no relationship between percentage of changed tokens and change in performance on downstream tasks. Thus, there was little effect of tokenization in our experiments.

4.2 Continued Pretraining

We create Glot500-m by continued pretraining of XLM-R-B with the MLM objective. The optimizer used is Adam with betas (0.9, 0.999). Initial learning rate: 5e-5. Each training step contains a batch of 384 training samples randomly picked from all language-scripts. The sampling strategy across language-scripts is the same as for vocabu-

	|head|	|tail|	measure (%)
Sentence Retrieval Tatoeba	70	28	Top10 Acc.
Sentence Retrieval Bible	94	275	Top10 Acc.
Text Classification	90	264	F1
NER	89	75	F1
POS	63	28	F1
Roundtrip Alignment	85	288	Accuracy

Table 3: Evaluation tasks and measures. |head|/|tail|: number of head/tail language-scripts

lary extension (§4.1). We save checkpoints every 10K steps and select the checkpoint with the best average performance on downstream tasks by early stopping. Table 2 lists the sizes of XLM-R-B, XLM-R-L and Glot500-m. Except for a larger vocabulary (§4.1), Glot500-m has the same size as XLM-R-B. We train Glot500-m on a server with eight NVIDIA RTX A6000 GPUs for two weeks.

Similar to XLM-R, we concatenate sentences' of a language-script and feed them as a stream to the tokenizer. The resulting output is then divided into chunks of 512 tokens and fed to the model.

5 Experimental Setup

For most tail languages, there are no manually labeled evaluation data. We therefore adopt a mixed evaluation strategy: based partly on human labels, partly on evaluation methods that are applicable to many languages without requiring gold data. Table 3 lists all our evaluation tasks.

Perplexity Following Salazar et al. (2020), we calculate pseudoperplexity (PPPL) over the held-out test set. PPPL is based on masking tokens one-by-one (not left to right). Salazar et al. (2020) give evidence that PPPL is a better measure of linguistic acceptability compared to standard left-to-right perplexity.

Roundtrip Alignment For assessing the quality of multilingual representations for a broad range of tail languages without human gold data, we adopt roundtrip evaluation (Dufter et al., 2018). We first word-align sentences' in a parallel corpus based on the multilingual representations of an LLM. We then start from a word w in a sentence' in language-script L1, follow the alignment links to its translations in language-script L2, then the alignment links from L2 to L3 and so on, until in the end we follow alignment links back to L1. If this "roundtrip" gets us back to w, then it indicates that the LLM has similar representations for the meaning of w in language-scripts L1, L2, L3, etc. In other words, the cross-lingual quality of representations is high. Vice versa, failure to get back to w is a sign of poor multilingual representations.

We use SimAlign (Jalili Sabet et al., 2020) and align on the sub-word level on the Bible part of test, based on the representations of the LLM computed by transformer layer 8 as suggested in the original paper. We use intersection symmetrization: each word in a sentence' is aligned to at most one word in the other sentence'.

As evaluation measure we compute the percentage of roundtrips that were successes, i.e., the roundtrip starts at w in L1 and returns back to w. For each language-script in test, we randomly select three language-scripts as intermediate points L2, L3, L4. Since the intermediate points influence the results, we run the experiment five times with different intermediate points and report the average. All models are evaluated with the same five sets of three intermediate language-scripts.

Sequence Labeling We consider two sequence labeling tasks: Named Entity Recognition (NER) and Part-Of-Speech (POS) tagging. We use the WikiANN dataset (Pan et al., 2017) for NER and version v2.11 of Universal Dependencies (UD) (de Marneffe et al., 2021) for POS. Since training data does not exist for some languages, we finetune on English (with early stopping based on dev) and evaluate zero-shot transfer on all languages covered by WikiANN/UD. We set the learning rate to 2e-5 with Adam.

Sentence Retrieval Following (Hu et al., 2020), we use up to 1000 English-aligned sentences' from Tatoeba (Artetxe and Schwenk, 2019) to evaluate SentRetr (sentence retrieval). We also use 500 English-aligned sentences' from the Bible part of test. We find nearest neighbors using cosine similarity based on the average word embeddings in layer $l = 8$ – following Jalili Sabet et al. (2020) – and compute top10 accuracy. For fair comparison and because the architectures are the same, we do not optimize the hyperparameter l for Glot500-m and XLM-R-B.

Text Classification We evaluate on Taxi1500 (Ma et al., 2023). It provides gold data for text classification with six classes in a large number of language-scripts of which Glot500-m supports 354. We finetune on English (with early stopping on dev) and evaluate zero-shot on test of the target language-script. Learning rate: 2e-5, batch size:

16 (following Ma et al. (2023)).

6 Experiments

In this section, we discuss aggregate results. For detailed results, see §D and §E.

6.1 Results

Table 4 gives results. Glot500-m outperforms XLM-R-B on all tasks for both head and tail language-scripts, except for POS on head. That Glot500-m outperforms XLM-R-B is expected for tail language-scripts (i.e., those not covered by XLM-R). For these language-scripts the improvement margin is large. Outperformance may seem counterintuitive for head language-scripts (those covered by XLM-R) since Glot500-m has the same number of (non-embedding) parameters as XLM-R-B. Since the number of covered languages has greatly increased, leaving less capacity per language, we might expect underperformance. There are a few possible explanations. First, XLM-R may be undertrained, and the inclusion of more head language training data may improve their representations. Second, having more languages may improve multilinguality by allowing languages to synergize and enhance each other's representations and cross-lingual transfer. Third, there are languages similar to head languages among the tail languages, which in turn aids head languages.

The gap between Glot500-m and the baselines for tail language-scripts in sequence labeling is smaller. These tasks do not require as deep an understanding of language and thus transfer from head to tail language-scripts is easier through shared tokens.

Glot500-m also outperforms XLM-R-L for tail language-scripts (all tasks) and head language-scripts (3 tasks). This suggests that scaling up size is not the only way for improvements. We can also improve the quality of multilingual LLM representations by increasing the number of languages.

6.2 Language Coverage

Table 5 compares Glot500-m vs. XLM-R-B on pseudoperplexity. For fair comparison we use word-level normalization. For 69 head language-scripts, Glot500-m underperforms XLM-R-B. This is expected as Glot500-m's training data is small for these language-scripts. Glot500-m outperforms XLM-R-B for 420 tail language-scripts.

There are eight tail language-scripts for which

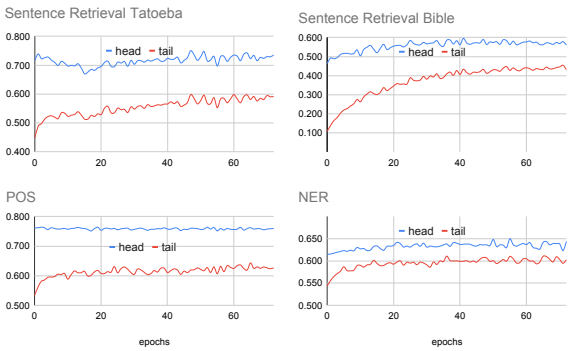


Figure 1: Progression of training for sentence retrieval and sequence labeling. x-axis: epochs/10K. The improvement is fast in the beginning for tail languages, then gets slower and and reaches a plateau. This pattern is partially observed for head languages.

Glot500-m performs worse than XLM-R-B. Five are tail languages with a similar head language where the two share a macro-language: ekk/Standard Estonian (est/Estonian), aln/Gheg Albanian (sqi/Albanian), nob/Norwegian Bokmal (nor/Norwegian), hbs/Serbo-Croatian (srp/Serbian), lvs/Standard Latvian (lav/Latvian). Since XLM-R-B's pretraining corpus is large for the five head languages, its performance is good for the close tail languages.

The other three languages all have a unique script: sat/Santali (Ol Chiki script), div/Dhivehi (Thaana script), iku/Inuktitut (Inuktitut syllabics). For these languages, XLM-R-B's tokenizer returns many UNK tokens since it is not trained on these scripts, resulting in an unreasonably optimistic estimate of pseudoperplexity by our implementation.

Glot500-m's token-level normalized pseudoperplexity ranges from 1.95 for lhu/Lahu to 94.4 for tok/Toki Pona. The average is 13.5, the median 10.6. We analyze the five language-scripts with the highest pseudoperplexity: tok_Latn, luo_Latn, acm_Arab, ach_Latn, and teo_Latn.

tok/Toki Pona is a constructed language. According to Wikipedia: "Essentially identical concepts can be described by different words as the choice relies on the speaker's perception and experience." This property can result in higher variability and higher perplexity.

acm/Mesopotamian Arabic contains a large number of tweets in raw form. This may result in difficult-to-predict tokens in test.

luo/Luo, ach/Acoli and teo/Teso are related Nilotic languages spoken in Kenya, Tanzania, Uganda and South Sudan. Their high perplex-

	tail			head			all		
	XLM-R-B	XLM-R-L	Glot500-m	XLM-R-B	XLM-R-L	Glot500-m	XLM-R-B	XLM-R-L	Glot500-m
Pseudoperplexity	304.2	168.6	**12.2**	12.5	**8.4**	11.8	247.8	136.4	**11.6**
Sentence Retrieval Tatoeba	32.6	33.6	**59.8**	66.2	71.1	**75.0**	56.6	60.4	**70.7**
Sentence Retrieval Bible	7.4	7.1	**43.2**	54.2	58.3	**59.0**	19.3	20.1	**47.3**
Text Classification	13.7	13.9	**46.6**	51.3	**60.5**	54.7	23.3	25.8	**48.7**
NER	47.5	51.8	**60.7**	61.8	**66.0**	63.9	55.3	59.5	**62.4**
POS	41.7	43.5	**62.3**	76.4	**78.4**	76.0	65.8	67.7	**71.8**
Roundtrip Alignment	2.6	3.1	**4.5**	3.4	4.1	**5.5**	2.8	3.3	**4.7**

Table 4: Evaluation of XLM-R base and large (XLM-R-B and XLM-R-L) and Glot500-m on pseudoperplexity and six multilingual tasks across 5 seeds. Each number is an average over head, tail and all language-scripts. See §D, §E for results per task and language-script. Glot500-m outperforms XLM-R-B in all tasks for head (except for POS) and tail language-scripts and XLM-R-L for tail language-scripts. Best result per row/column group in bold.

	head	tail
Glot500-m is better	37	420
XLM-R-B is better	69	8

Table 5: Pseudoperplexity Glot500-m vs XLM-R-B. Glot500-m's worse performance on head can be attributed to smaller training corpora and the relative difficulty of learning five times more languages with the same number of (non-embedding) parameters. Glot500-m performs better on almost all tail language-scripts. §6.2 discusses the eight exceptions.

ity could be related to the fact that they are tonal languages, but the tones are not orthographically indicated. Another possible explanation is that the training data is dominated by one subcorpus (Jehova's Witnesses) whereas the test data are dominated by PBC. There are orthographic differences between the two, e.g., "dong" (JW) vs. "doŋ" (PBC) for Acoli. These three languages are also spoken over a large area in countries with different standard languages, which could increase variability.

Our analysis is not conclusive. We note however that the gap between the three languages and the next most difficult languages in terms of pseudoperplexity is not large. So maybe Luo, Acoli and Teso are simply (for reasons still to be determined) languages that have higher perplexity than others.

6.3 Training Progression

To analyze the training process, we evaluate Glot500-m on sequence labeling and SentRetr at 10,000-step intervals. Figure 1 shows that performance improves rapidly at the onset of training, but then the rate of improvement slows down. This trend is particularly pronounced for tail languages in SentRetr. In comparison, sequence labeling is relatively straightforward, with the baseline (XLM-R-B, epoch 0) achieving high performance by correctly transferring prevalent classes such as *verb* and *noun* through shared vocabulary, resulting in a smaller improvement of Glot500-m vs. XLM-R-B.

For SentRetr, we observe larger improvements for the Bible than for Tatoeba. This is likely due to the higher proportion of religious data in Glot500-c, compared to XLM-R's training data (i.e., CC100).

The average performance on downstream tasks peaks at 480K steps. We have taken a snapshot of Glot500-m at this stage and released it.

6.4 Analysis across Language-Scripts

To analyze the effect of language-scripts, we select five tail language-scripts each with the largest and smallest gain when comparing Glot500-m vs. XLM-R-B for SentRetr and sequence labeling.

Table 6 shows that Glot500-m improves languages with scripts not covered by XLM-R (e.g., div/Dhivehi, Thaana script, see §6.2) by a large margin since XLM-R simply regards the uncovered scripts as unknown tokens and cannot compute meaningful representations for the input. The large amount of data we collected in Glot500-c also contributes to the improvement for tail languages, e.g., for tat_Cyrl (Tatar) in SentRetr Tatoeba and mlt_Latn (Maltese) in POS. See §6.7 for a detailed analysis of the effect of corpus size.

On the other hand, Glot500-m achieves just comparable or even worse results for some language-scripts. We see at least three explanations. (i) As discussed in §6.2, some tail languages (e.g., nob/Norwegian Bokmal) are close to a head language (e.g., nor/Norwegian), so Glot500-m has no advantage over XLM-R-B. (ii) A language is at the low end of our corpus size range (i.e., 30,000 sentences'). Example: xav_Latn, Xavánte. (iii) Some languages are completely distinct from all other languages in Glot500-c, thus without support from any similar language. An example is mau_Latn, Huautla Mazatec. Glot500-m has a much harder

		language-script	XLMR	Glot500	gain		language-script	XLMR	Glot500	gain
high end	SentRetr Tatoeba	tat C Tatar	10.3	70.3	60.0	SentRetr Bible	uzn C Northern Uzbek	5.4	87.0	81.6
		nds L Low German	28.8	77.1	48.3		crs L Seselwa Creole	7.4	80.6	73.2
		tuk L Turkmen	16.3	63.5	47.3		srn L Sranan Tongo	6.8	79.8	73.0
		ile L Interlingue	34.6	75.6	41.0		uzb C Uzbek	6.2	78.8	72.6
		uzb C Uzbek	25.2	64.5	39.3		bcl L Central Bikol	10.2	79.8	69.6
low end		dtp L Kadazan Dusun	5.6	21.1	15.5		xav L Xavánte	2.2	5.0	2.8
		kab L Kabyle	3.7	16.4	12.7		mau L Huautla Mazatec	2.4	3.6	1.2
		pam L Pampanga	4.8	11.0	6.2		ahk L Akha	3.0	3.2	0.2
		lvs L Standard Latvian	73.4	76.9	3.5		aln L Gheg Albanian	67.8	67.6	-0.2
		nob L Bokmål	93.5	95.7	2.2		nob L Bokmål	82.8	79.2	-3.6
high end	NER	div T Dhivehi	0.0	50.9	50.9	POS	mlt L Maltese	21.3	80.3	59.0
		che C Chechen	15.3	61.2	45.9		sah C Yakut	21.9	76.9	55.0
		mri L Maori	16.0	58.9	42.9		sme L Northern Sami	29.6	73.6	44.1
		nan L Min Nan	42.3	84.9	42.6		yor L Yoruba	22.8	64.2	41.4
		tgk C Tajik	26.3	66.4	40.0		quc L K'iche'	28.5	64.1	35.6
low end		zea L Zeeuws	68.1	67.3	-0.8		lzh H Literary Chinese	11.7	18.4	6.7
		vol L Volapük	60.0	59.0	-1.0		nap L Neapolitan	47.1	50.0	2.9
		min L Minangkabau	42.3	40.4	-1.8		hyw A Western Armenian	79.1	81.1	2.0
		wuu H Wu Chinese	28.9	23.9	-5.0		kmr L Northern Kurdish	73.5	75.2	1.7
		lzh H Literary Chinese	15.7	10.3	-5.4		aln L Gheg Albanian	54.7	51.2	-3.5

Table 6: Results for five tail language-scripts each with the largest (high end) and smallest (low end) gain Glot500-m vs. XLM-R-B for four tasks. Glot500-m's gain over XLM-R-B is large at the high end and small or slightly negative at the low end. L = Latin, C = Cyrillic, H = Hani, A = Armenian, T = Thaana

lang-script		XLM-R-B	Glot500-m	gain
uig_Arab	head	45.8	56.2	10.4
uig_Latn	tail	9.8	62.8	53.0
hin_Deva	head	67.0	76.6	9.6
hin_Latn	tail	13.6	43.2	29.6
uzb_Latn	head	54.8	67.6	12.8
uzb_Cyrl	tail	6.2	78.8	72.6
kaa_Cyrl	tail	17.6	73.8	56.2
kaa_Latn	tail	9.2	43.4	34.2
kmr_Cyrl	tail	4.0	42.4	38.4
kmr_Latn	tail	35.8	63.0	27.2
tuk_Cyrl	tail	13.6	65.0	51.4
tuk_Latn	tail	9.6	66.2	56.6

Table 7: Sentence Retrieval Bible performance of Glot500-m and XLM-R-B for six languages with two scripts: Uighur (uig), Hindi (hin), Uzbek (uzb), Kara-Kalpak (kaa), Northern Kurdish (kmr), Turkmen (tuk). Glot500-m clearly outperforms XLM-R-B with large differences for tail language-scripts.

time learning good representations in these cases.

6.5 Languages with Multiple Scripts

Table 7 compares SentRetr performance XLM-R-B vs. Glot500-m for six languages with two scripts. Unsurprisingly, XLM-R performs much better for a language-script it was pretrained on ("head") than on one that it was not ("tail"). We can improve the performance of a language, even surpassing the language-script covered by XLM-R, if we collect enough data for its script not covered by XLM-R. For languages with two scripts not covered by XLM-R, the performance is better for the script for which we collect a larger corpus. For example, kaa_Cyrl (Kara-Kalpak) has about three times as much data as kaa_Latn. This explains why kaa_Cyrl outperforms kaa_Latn by 30%.

Dufter and Schütze (2020) found that, after training a multilingual model with two scripts for English (natural English and "fake English"), the model performed well at zero-shot transfer if the capacity of the model was of the right size (i.e., not too small, not too large). Our experiments with real data show the complexity of the issue: even if there is a "right" size for an LLM that supports both full acquisition of languages and multilingual transfer, this size is difficult to determine and it may be different for different language pairs in a large horizontally scaled model like Glot500-m.

6.6 Analysis across Language Families

Table 8 compares SentRetr performance Glot500-m vs. XLM-R-B for seven language families that have ten or more language-scripts in Glot500-c. We assign languages to families based on Glottolog.[4] Generally, XLM-R has better performance the more language-scripts from a language family are represented in its training data; e.g., performance is better for indo1319 and worse for maya1287. The results suggest that Glot500-m's improvement over

[4] http://glottolog.org/glottolog/family

family	$\|L_G\|$	$\|L_X\|$	XLM-R-B	Glot500-m	gain
indo1319	91	50	41.5	61.4	19.9
atla1278	69	2	5.5	45.2	39.6
aust1307	53	6	13.7	47.0	33.2
turk1311	22	7	20.1	62.9	42.8
sino1245	22	2	7.6	38.9	31.3
maya1287	15	0	3.8	20.3	16.4
afro1255	12	5	13.0	34.3	21.4

Table 8: Average Sentence Retrieval Bible performance of Glot500-m and XLM-R-B for seven language families. The difference in coverage of a family by Glot500-m vs. XLM-R-B is partially predictive of the performance difference. $|L_G|/|L_X|$: number of language-scripts from family covered by Glot500-m/XLM-R.

lang-script	Glot+1	Glot500-m
rug_Latn, Roviana	**51.0**	49.0
yan_Latn, Mayangna/Sumo	**46.4**	31.8
wbm_Latn, Wa/Va	**49.6**	46.4
ctd_Latn, Tedim Chin	47.4	**59.4**
quh_Latn, Southern Quechua	33.4	**56.2**
tat_Cyrl, Tatar	58.8	**67.2**

Table 9: Performance on Sentence Retrieval Bible of continued pretraining on just one language-script (Glot+1) vs. on Glot500-c (Glot500-m). Glot500-m underperforms on the top three and outperforms on the bottom three. Our explanation is that the second group is supported by closely related languages in Glot500-c; e.g., for Southern Quechua (quh), Glot500-m also covers closely related Cuzco Quechua (quz). For the first group this is not the case; e.g., the Wa language (wbm) has no close relative in Glot500-c.

XLM-R is the larger, the better our training corpus Glot500-c's coverage is of a family.

6.7 Effect of Amount of Training Data

We examine correlation between pretraining corpus size and Glot500-m zero-shot performance. We focus on SentRetr Bible (§5) since it supports the most head and tail languages. We find that Pearson's $r = .34$, i.e., corpus size and performance are moderately, but clearly correlated. We suspect that the correlation is not larger because, in addition to corpus size of language l itself, corpus size of languages closely related to l is also an important factor (see §6.4 for a similar finding for Norwegian). We therefore also compute Pearson's r between (i) performance of language l on SentRetr Bible and (ii) joint corpus size of l and its k nearest neighbors (according to perplexity divergence, §3.3). In this case, Pearson's $r = .44$ (for both $k = 3$ and $k = 4$), indicating that the corpus size of nearest neighbor languages does play a role.

6.8 Support through Related Languages

Building on §6.7, there is another way we can investigate the positive effect of closely related languages on performance: We can compare performance (again on SentRetr Bible) of continued pretraining on just one language (we refer to this model as Glot+1) vs. on all 511 languages represented in Glot500-c (i.e., Glot500-m). Table 9 presents results for six language-scripts selected from various language families and suggests that some languages do not receive support from related languages (top three). In that case, Glot+1 can fully concentrate on learning the isolated language and does better than Glot500-c. Other languages (bottom three) do receive support from related languages. For example, Southern Quechua (quh) seems to receive support in Glot500-m from closely related Cuzco Quechua (quz), resulting in Glot500-m outperforming Glot+1.

7 Conclusion and Future Work

We collect and data-clean Glot500-c, a large corpus of hundreds of usually neglected tail (i.e., long-tail) languages and create Glot500-m, an LLM that is trained on Glot500-c and covers these languages. We evaluate Glot500-m on six tasks that allow us to evaluate almost all languages. We observe large improvements for both head and tail languages compared to XLM-R. Our analysis shows that no single factor fully explains the quality of the representation of a language in a multilingual model. Rather, a combination of factors is important, including corpus size, script, "help" from related languages and the total capacity of the model.

This work is the first to create a language model on a dataset of several hundreds of gigabytes and to make it publicly available for such a large and diverse number of low-resource languages. In future research, we would like to train larger models to further investigate the effect of model size, distill highly multilingual models for resource-efficient deployment, explore alternatives to continued pretraining and use models for more tail language downstream tasks.

Limitations

(1) We did not perform any comprehensive hyperparameter search, which would have further consolidated our results. This decision was made due to the high cost of training multiple models. (2) Compared to current very large models, Glot500-m

is comparatively small. (3) Although we have tried to minimize the amount of noise in our data, some noise is still present.

Ethics Statement

There are two issues worth mentioning in regards to this project. First, it was not feasible for us to thoroughly examine the content of the data for all languages, thus we cannot confirm the absence of discrimination based on factors such as race or sexuality. The data was solely utilized as a textual corpus, and the content should not be interpreted as an endorsement by our team. If the model is subsequently utilized for generation, it is possible that the training data may be reflected in the generated output. However, addressing potential biases within the data is an area for future research. Second, it is important to note that while the data sources utilized in this study do not explicitly prohibit the reuse of data for research purposes, some sources do have copyright statements indicating that such use is permissible while others do not. Additionally, certain sources prohibit the redistribution of data. As such, data from these sources is omitted from the published version of Glot2000-c.

Acknowledgements

We would like to thank Renhao Pei, Yihong Liu, Verena Blaschke, and the anonymous reviewers. This work was funded by the European Research Council (grants #740516 and #758969) and EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631).

References

Solomon Teferra Abate, Michael Melese, Martha Yifiru Tachbelie, Million Meshesha, Solomon Atinafu, Wondwossen Mulugeta, Yaregal Assabie, Hafte Abera, Binyam Ephrem, Tewodros Abebe, Wondimagegnhue Tsegaye, Amanuel Lemma, Tsegaye Andargie, and Seifedin Shifaw. 2018. Parallel corpora for bi-lingual English-Ethiopian languages statistical machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3102–3111, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 1–10, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. Shami: A corpus of Levantine Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2022. SERENGETI: Massively multilingual language models for Africa. *arXiv preprint arXiv:2212.10785*.

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Adelani, Dana Ruiter, Jesujoba Alabi, Damilola Adebonojo, Adesina Ayeni, Mofe Adeyemi, Ayodele Esther Awokoya, and Cristina España-Bonet. 2021. The effect of domain and diacritics in Yoruba–English neural machine translation. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 61–75, Virtual. Association for Machine Translation in the Americas.

Rodrigo Agerri, Xavier Gómez Guinovart, German Rigau, and Miguel Anxo Solla Portela. 2018. Developing new linguistic resources and tools for the Galician language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. DART: A large dataset of dialectal Arabic tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, Dublin, Ireland. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Niyati Bafna. 2022. Empirical models for an indic language continuum.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Marta Bañón, Miquel Esplà-Gomis, Mikel L. Forcada, Cristian García-Romero, Taja Kuzman, Nikola Ljubesic, Rik van Noord, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Peter Rupnik, Vít Suchomel, Antonio Toral, Tobias van der Werff, and Jaume Zaragoza. 2022. Macocu: Massive collection and curation of monolingual and bilingual data: focus on under-resourced languages. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation, EAMT 2022, Ghent, Belgium, June 1-3, 2022*, pages 301–302. European Association for Machine Translation.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. *arXiv preprint arXiv:2205.03983*.

Workshop BigScience, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von

Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. BLOOM: a 176b-parameter open-access multilingual language model.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

José Camacho-Collados, Claudio Delli Bovi, Alessandro Raganato, and Roberto Navigli. 2016. A large-scale multilingual disambiguation of glosses. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1701–1708, Portorož, Slovenia. European Language Resources Association (ELRA).

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. XLM-E: Cross-lingual language model pre-training via ELECTRA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6170–6182, Dublin, Ireland. Association for Computational Linguistics.

Rochelle Choenni and Ekaterina Shutova. 2022. Investigating language relationships in multilingual sentence encoders through the lens of linguistic typology. *Computational Linguistics*, 48(3):635–672.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised

cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational Linguistics*, 47(2):255–308.

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.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1520–1530, Melbourne, Australia. Association for Computational Linguistics.

Jonathan Dunn. 2020. Mapping languages: the corpus of global language use. *Lang. Resour. Evaluation*, 54(4):999–1018.

Eberhard, David M., Gary F. Simons, and Charles D. Fennig (eds.). 2022. Ethnologue: Languages of the world. twenty-fifth edition.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. 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 4555–4567, Online. Association for Computational Linguistics.

Mahmoud El-Haj. 2020. Habibi - a multi dialect multi national Arabic song lyrics corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.

Mahmoud El-Haj, Paul Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48.

Pablo Gamallo, Jose Ramom Pichel, and Iñaki Alegria. 2017. A perplexity-based method for similar languages discrimination. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 109–114, Valencia, Spain. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 759–765. European Language Resources Association (ELRA).

Santiago Góngora, Nicolás Giossa, and Luis Chiruzzo. 2021. Experiments on a Guarani corpus of news and social media. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 153–158, Online. Association for Computational Linguistics.

Santiago Góngora, Nicolás Giossa, and Luis Chiruzzo. 2022. Can we use word embeddings for enhancing Guarani-Spanish machine translation? In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 127–132, Dublin, Ireland. Association for Computational Linguistics.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 306–316, Online. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Samin Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 4693–4703. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Ayyoob ImaniGooghari, Silvia Severini, Masoud Jalili Sabet, François Yvon, and Hinrich Schütze. 2022. Graph-based multilingual label propagation for low-resource part-of-speech tagging. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1577–1589, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

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

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

Fajri Koto and Ikhwan Koto. 2020. Towards computational linguistics in Minangkabau language: Studies on sentiment analysis and machine translation. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 138–148, Hanoi, Vietnam. Association for Computational Linguistics.

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.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for 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.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The BigScience ROOTS Corpus: A 1.6 TB Composite Multilingual Dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Colin Leong, Joshua Nemecek, Jacob Mansdorfer, Anna Filighera, Abraham Owodunni, and Daniel Whitenack. 2022. Bloom library: Multimodal datasets in 300+ languages for a variety of downstream tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8608–8621. Association for Computational Linguistics.

Chunlan Ma, Ayyoob ImaniGooghari, Haotian Ye, Ehsaneddin Asgari, and Hinrich Schütze. 2023. Taxi1500: A multilingual dataset for text classification in 1500 languages.

Martin Majliš. 2011. W2C – web to corpus – corpora. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jamshidbek Mirzakhalov, Anoop Babu, Duygu Ataman, Sherzod Kariev, Francis Tyers, Otabek Abduraufov, Mammad Hajili, Sardana Ivanova, Abror Khaytbaev, Antonio Laverghetta Jr., Bekhzodbek Moydinboyev, Esra Onal, Shaxnoza Pulatova, Ahsan Wahab, Orhan Firat, and Sriram Chellappan. 2021. A large-scale study of machine translation in Turkic languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5876–5890, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Steven Moran, Christian Bentz, Ximena Gutierrez-Vasques, Olga Pelloni, and Tanja Samardzic. 2022. TeDDi sample: Text data diversity sample for language comparison and multilingual NLP. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1150–1158, Marseille, France. European Language Resources Association.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3603–3609, Marseille, France. European Language Resources Association.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Raj Dabre, Shohei Higashiyama, Shantipriya Parida, Anoop Kunchukuttan, Makoto Morishita, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2022. Overview of the 9th workshop on Asian translation. In *Proceedings of the 9th Workshop on Asian Translation*, pages 1–36, Gyeongju, Republic of Korea. International Conference on Computational Linguistics.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 1–45, Online. Association for Computational Linguistics.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021a. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021b. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126.

Chester Palen-Michel, June Kim, and Constantine Lignos. 2022. Multilingual open text release 1: Public domain news in 44 languages. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2080–2089, Marseille, France. European Language Resources Association.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring.

In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Silvia Severini, Ayyoob Imani, Philipp Dufter, and Hinrich Schütze. 2022. Towards a broad coverage named entity resource: A data-efficient approach for many diverse languages. *arXiv preprint arXiv:2201.12219*.

Aditya Siddhant, Ankur Bapna, Orhan Firat, Yuan Cao, Mia Xu Chen, Isaac Caswell, and Xavier Garcia. 2022. Towards the next 1000 languages in multilingual machine translation: Exploring the synergy between supervised and self-supervised learning. *arXiv preprint arXiv:2201.03110*.

Anil Kumar Singh. 2008. Named entity recognition for south and south East Asian languages: Taking stock. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.

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

Iulia Turc, Kenton Lee, Jacob Eisenstein, Ming-Wei Chang, and Kristina Toutanova. 2021. Revisiting the primacy of english in zero-shot cross-lingual transfer. *CoRR*, abs/2106.16171.

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, and Dong Yu. 2019. Improving pre-trained multilingual model with vocabulary expansion. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 316–327, Hong Kong, China. Association for Computational Linguistics.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2023. NLNDE at semeval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis. *CoRR*, abs/2305.00090.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020a. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4003–4012. European Language Resources Association.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020b. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

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.

Jian Yang, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9386–9393.

Rodolfo Zevallos, John Ortega, William Chen, Richard Castro, Núria Bel, Cesar Toshio, Renzo Venturas, Hilario Aradiel, and Nelsi Melgarejo. 2022. Introducing QuBERT: A large monolingual corpus and BERT model for Southern Quechua. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 1–13, Hybrid. Association for Computational Linguistics.

Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. 2020. Masking as an efficient alternative to finetuning for pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2226–2241, Online. Association for Computational Linguistics.

A N-grams LMs and Language Divergence

Perplexity and Language Divergence. Perplexity measures how well a model predicts a sample test data. Assuming a test data contains sequences of

characters $S = ch_1, ch_2, \cdots, ch_T$, perplexity ($\mathcal{PP}$) of S given an n-gram character level language model M is computed as follows:

$$\mathcal{PP}(S, M) = \sqrt[T]{\prod_{t=1}^{T} \frac{1}{\mathbb{P}\left(ch_t \mid ch_1^{t-1}\right)}} \quad (1)$$

where $\mathbb{P}\left(ch_t \mid ch_1^{t-1}\right)$ is computed as by dividing the observed frequency (C) of $ch_1^{t-1}ch_i$ by the observed frequency of ch_1^{t-1} in M training data:

$$\mathbb{P}\left(ch_t \mid ch_1^{t-1}\right) = \frac{C\left(ch_1^{t-1}ch_t\right)}{C\left(ch_1^{t-1}\right)} \quad (2)$$

Given the definition of perplexity, we can determine how well a trained language model on language L_1 predicts the test text of language L_2 and vice-versa. The divergence between two languages is computed with the maximum of the perplexity values in both directions. Two reasons lead to the use of max: first, a symmetrical divergence is required, and second, languages differ in their complexity, so one direction of computing perplexity may result in a much lower perplexity than another. Thus, comparing perplexity results becomes difficult. As an example, the Kuanua language (ksd_Latn) has short words and a simple structure, which results in 3−gram models getting lower perplexity on its text compared to other languages. The lower the perplexity the smaller the divergence between languages. The divergence ($\mathcal{D}$) between language L_i and L_j with trained language models of M_{L_z} and test texts of S_{L_z}, where L_z is the corresponding language, computed as follows:

$$\mathcal{D}_{L_i,L_j} = \max\left(\mathcal{PP}(S_{L_i}, M_{L_j}), \mathcal{PP}(S_{L_j}, M_{L_i})\right) \quad (3)$$

Runs and Data. The data used to train and test the character level n-gram models is the same data used for the training and testing of the Glot500-m. The training of the models was limited to 100,000 sentences' per language-script. We use KenLM library (Heafield, 2011) to build n-gram models. This library uses an interpolated modified Kneser-Ney smoothing for estimating the unseen n-grams. Our evaluation has been performed over 7 n-gram models ($3 \leq n \leq 9$).

Baseline and Evaluation. Language family trees were used as a baseline for evaluating the divergence measures of the proposed approach. We obtained language family tree data from Ethnologue online version (Eberhard et al., 2022). For each language, the family tree follows the general order from largest typological language family group to smallest. There is only one family tree for each language in the baseline data. Nodes in the family tree represent typological language family groups. Each node only has one parent, so if a node is common in the family tree of two languages, its parent is also common. We evaluate our perplexity method on the following binary classification task: Do the majority of a language L_z's k nearest neighbors belong to the same typological language family group as L_z? Assuming languages L_i and L_j, with the following family trees:

T_{L_i} : ❶ → ❷ → ❸ → ❹ → ❺ → ❻
T_{L_j} : ❶ → ❷ → ❼ → ❽

These 2 languages belong to the same typological family group with family tree levels of $l \in \{1, 2\}$, but not with family tree levels of $l = 3$ and higher.

Result. When it comes to language families, the majority of studies only refer to the largest typological language family group (level $l = 1$). Here, we also assess our methodology for other levels. The results of classification accuracy for 3−gram model, $k \in \{1, 3, 7, 13, 21\}$ and $l \in \{1, 2, 3, \max\}$ are shown in Table 10. In cases where the maximum level of a tree is less than the l parameter, the maximum level for that language is used. Languages without a family or no other family member in our data are excluded. We only report the 3−gram model results as it gets the best results in most configurations among other n-gram models. With increasing l, the accuracy decreases, since more languages fall outside the same typological family. As k increases, the accuracy decreases, because languages with faraway neighbors are being included but the number of languages in the language typological group family will remain the same. There are times when languages have a lot of loan words from other languages because of geological proximity or historical reasons (e.g, colonization), which makes them similar to the languages they borrowed words from in our method. However they are different when it comes to their typological families and our method fails in these cases. Aymara (Macrolanguage: aym_Latn) and Quechua (Macrolanguage: que_Latn), for example, had a great deal of contact and influence on each other, but they do not belong to the same typological group. As well, some of the typological families are not that large, which makes our results worse when k increases. This is

the case, for instance, of the Tarascan typological family which only has two members.

model	l	k	accuracy (%)
3-gram	1	1	84.45
3-gram	1	3	75.77
3-gram	1	7	69.08
3-gram	1	13	62.75
3-gram	1	21	55.33
3-gram	2	1	79.75
3-gram	2	3	67.63
3-gram	2	7	59.49
3-gram	2	13	51.36
3-gram	2	21	42.68
3-gram	3	1	75.05
3-gram	3	3	60.22
3-gram	3	7	49.55
3-gram	3	13	38.34
3-gram	3	21	29.84
3-gram	max	1	59.31
3-gram	max	3	36.89
3-gram	max	7	18.81
3-gram	max	13	6.87
3-gram	max	21	2.89

Table 10: Detecting the typological relatedness of language with n-gram divergence: (Eq. 3); l: level of typological language family group; k: number of nearest language neighbors.

B Languages

The list of languages used to train Glot500-m with the amount of available data for each language is available in Tables 11, 12 and 13.

On Macrolanguages The presence of language codes that are supersets of other language codes within datasets is not uncommon (Kreutzer et al., 2022). This issue becomes more prevalent in extensive collections. Within the ISO 639-3 standard, these languages are referred to as macrolanguages. When confronted with macrolanguages, if it is not feasible to ascertain the specific individual language contained within a dataset, the macrolanguage code is retained. Consequently, it is possible that in Glot2000-c and Glot500-c both the corpora for the macrolanguage and its individual languages have been included.

C List of data sources

The datasets and repositories used in this project involve: AI4Bharat,[5] AIFORTHAI-LotusCorpus,[6] Add (El-Haj et al., 2018), AfriBERTa (Ogueji et al., 2021b), AfroMAFT (Adelani et al., 2022; Xue et al., 2021), Anuvaad,[7] AraBench (Sajjad et al., 2020), AUTSHUMATO,[8] Bloom (Leong et al., 2022), CC100 (Conneau et al., 2020; Wenzek et al., 2020a), CCNet (Wenzek et al., 2020b), CMU_Haitian_Creole,[9] CORP.NCHLT,[10] Clarin,[11] DART (Alsarsour et al., 2018), Earthlings (Dunn, 2020), FFR,[12] Flores200 (Costa-jussà et al., 2022), GiossaMedia (Góngora et al., 2022, 2021), Glosses (Camacho-Collados et al., 2016), Habibi (El-Haj, 2020), HinDialect (Bafna, 2022), HornMT,[13] IITB (Kunchukuttan et al., 2018), IndicNLP (Nakazawa et al., 2021), Indiccorp (Kakwani et al., 2020), isiZulu,[14] JParaCrawl (Morishita et al., 2020), KinyaSMT,[15] LeipzigData (Goldhahn et al., 2012), Lindat,[16] Lingala_Song_Lyrics,[17] Lyrics,[18] MC4 (Raffel et al., 2020), MTData (Gowda et al., 2021), MaCoCu (Bañón et al., 2022), Makerere MT Corpus,[19] Masakhane community,[20] Mburisano_Covid,[21] Menyo20K (Adelani et al., 2021), Minangkabau corpora (Koto and Koto, 2020), MoT (Palen-Michel et al., 2022), NLLB_seed (Costa-jussà et al., 2022), Nart/abkhaz,[22] OPUS (Tiedemann, 2012), OSCAR (Suárez et al., 2019), ParaCrawl (Bañón et al., 2020), Parallel Corpora for Ethiopian Lan-

[5] https://ai4bharat.org/

[6] https://github.com/korakot/corpus/releases/download/v1.0/AIFORTHAI-LotusCorpus.zip

[7] https://github.com/project-anuvaad/anuvaad-parallel-corpus

[8] https://autshumato.sourceforge.net/

[9] http://www.speech.cs.cmu.edu/haitian/text/

[10] https://repo.sadilar.org/handle/20.500.12185/7

[11] https://www.clarin.si/

[12] https://github.com/bonaventuredossou/ffr-v1/tree/master/FFR-Dataset

[13] https://github.com/asmelashteka/HornMT

[14] https://zenodo.org/record/5035171

[15] https://github.com/pniyongabo/kinyarwandaSMT

[16] https://lindat.cz/faq-repository

[17] https://github.com/espoirMur/songs_lyrics_webscrap

[18] https://lyricstranslate.com/

[19] https://zenodo.org/record/5089560

[20] https://github.com/masakhane-io/masakhane-community

[21] https://repo.sadilar.org/handle/20.500.12185/536

[22] https://huggingface.co/datasets/Nart/abkhaz_text

Language-Script	|Sent|	Family	Head	Language-Script	|Sent|	Family	Head	Language-Script	|Sent|	Family	Head
hbs_Latn	63411156	indo1319		vec_Latn	514240	indo1319		swh_Latn	95776	atla1278	yes
mal_Mlym	48098273	drav1251	yes	jpn_Jpan	510722	japo1237	yes	alt_Cyrl	95148	turk1311	
aze_Latn	46300705		yes	lus_Latn	509250	sino1245		rmn_Grek	94533	indo1319	
guj_Gujr	45738685	indo1319	yes	crs_Latn	508755	indo1319		miq_Latn	94343	misu1242	
ben_Beng	43514870	indo1319	yes	kqn_Latn	507913	atla1278		kaa_Cyrl	88815	turk1311	
kan_Knda	41836495	drav1251	yes	ndo_Latn	496613	atla1278		kos_Latn	88603	aust1307	
tel_Telu	41580525	drav1251	yes	snd_Arab	488730	indo1319	yes	grn_Latn	87568		
mlt_Latn	40654838	afro1255		yue_Hani	484700	sino1245		lhu_Latn	87255	sino1245	
fra_Latn	39197581	indo1319	yes	tiv_Latn	483064	atla1278		lzh_Hani	86035	sino1245	
spa_Latn	37286756	indo1319	yes	kua_Latn	473535	atla1278		ajp_Arab	83297	afro1255	
eng_Latn	36122761	indo1319	yes	kwy_Latn	473274	atla1278		cmn_Hani	80745	sino1245	yes
fil_Latn	33493255	aust1307	yes	hin_Latn	466175	indo1319		gcf_Latn	80737	indo1319	
nob_Latn	32869205	indo1319		iku_Cans	465011			rmn_Cyrl	79925	indo1319	
rus_Cyrl	31787973	indo1319	yes	kal_Latn	462430	eski1264		kjh_Cyrl	79262	turk1311	
deu_Latn	31015993	indo1319	yes	tdt_Latn	459818	aust1307		rng_Latn	78177	atla1278	
tur_Latn	29184662	turk1311	yes	gsw_Latn	449240	indo1319		mgh_Latn	78117	atla1278	
pan_Guru	29052537	indo1319	yes	mfe_Latn	447435	indo1319		xmv_Latn	77896	aust1307	
mar_Deva	28748897	indo1319	yes	swc_Latn	446378	atla1278		ige_Latn	77114	atla1278	
por_Latn	27824391	indo1319	yes	mon_Latn	437950	mong1349		rmy_Latn	76991	indo1319	
nld_Latn	25061426	indo1319	yes	mos_Latn	437666	atla1278		srm_Latn	76884	indo1319	
ara_Arab	24524122		yes	kik_Latn	437228	atla1278		bak_Latn	76809	turk1311	
zho_Hani	24143786		yes	cnh_Latn	436667	sino1245		gur_Latn	76151	atla1278	
ita_Latn	23539857	indo1319	yes	gil_Latn	434529	aust1307		idu_Latn	75106	atla1278	
ind_Latn	23018106	aust1307	yes	pon_Latn	434522	aust1307		yom_Latn	74818	atla1278	
ell_Grek	22033282	indo1319	yes	umb_Latn	431589	atla1278		tdx_Latn	74430	aust1307	
bul_Cyrl	21823004	indo1319	yes	lvs_Latn	422952	indo1319		mzn_Arab	73719	indo1319	
swe_Latn	20725883	indo1319	yes	sco_Latn	411591	indo1319		cfm_Latn	70227	sino1245	
ces_Latn	20376340	indo1319	yes	ori_Orya	410827		yes	zpa_Latn	69237	otom1299	
isl_Latn	19547941	indo1319	yes	arg_Latn	410683	indo1319		kbd_Cyrl	67914	abkh1242	
pol_Latn	19339945	indo1319	yes	kur_Latn	407169	indo1319	yes	lao_Laoo	66966	taik1256	yes
ron_Latn	19190217	indo1319	yes	dhv_Latn	405711	aust1307		nap_Latn	65826	indo1319	
dan_Latn	19174573	indo1319	yes	luo_Latn	398974	nilo1247		qub_Latn	64973	quec1387	
hun_Latn	18800025	ural1272	yes	lun_Latn	395764	atla1278		oke_Latn	64508	atla1278	
tgk_Cyrl	18659517	indo1319		nzi_Latn	394247	atla1278		ote_Latn	64224	otom1299	
srp_Latn	18371769	indo1319	yes	gug_Latn	392227	tupi1275		bsb_Latn	63634	aust1307	
fas_Arab	18277593		yes	bar_Latn	387070	indo1319		ogo_Latn	61901	atla1278	
ceb_Latn	18149215	aust1307		bci_Latn	384059	atla1278		abn_Latn	61830	atla1278	
heb_Hebr	18128962	afro1255	yes	chk_Latn	380596	aust1307		ldi_Latn	61827	atla1278	
hrv_Latn	17882932	indo1319	yes	roh_Latn	377067	indo1319		ayr_Latn	61570	ayma1253	
glg_Latn	17852274	indo1319	yes	aym_Latn	373329	ayma1253		gom_Deva	61140	indo1319	
fin_Latn	16730388	ural1272	yes	yap_Latn	358929	aust1307		bba_Latn	61123	atla1278	
slv_Latn	15719210	indo1319	yes	ssw_Latn	356561	atla1278		aln_Latn	60989	indo1319	
vie_Latn	15697827	aust1305	yes	quz_Latn	354781	quec1387		leh_Latn	59944	atla1278	
mkd_Cyrl	14717004	indo1319	yes	sah_Cyrl	352697	turk1311		ban_Latn	59805	aust1307	
slk_Latn	14633631	indo1319	yes	tsn_Latn	350954	atla1278		ace_Latn	59333	aust1307	
nor_Latn	14576191	indo1319	yes	lmo_Latn	348135	indo1319		pes_Arab	57511	indo1319	yes
est_Latn	13600579		yes	ido_Latn	331239	arti1236		skg_Latn	57228	aust1307	
ltz_Latn	12997242	indo1319		abk_Cyrl	321578	abkh1242		ary_Arab	56933	afro1255	
eus_Latn	12775959		yes	zne_Latn	318871	atla1278		hus_Latn	56176	maya1287	
lit_Latn	12479626	indo1319	yes	quy_Latn	311040	quec1387		glv_Latn	55641	indo1319	
kaz_Cyrl	12378727	turk1311	yes	kam_Latn	310659	atla1278		fat_Latn	55609	atla1278	
lav_Latn	12143980	indo1319	yes	bbc_Latn	310420	aust1307		frr_Latn	55254	indo1319	
bos_Latn	11014744	indo1319	yes	vol_Latn	310399	arti1236		mwn_Latn	54805	atla1278	
epo_Latn	8737198	arti1236	yes	wal_Latn	309873	gong1255		mai_Deva	54687	indo1319	
cat_Latn	8648271	indo1319	yes	uig_Arab	307302	turk1311	yes	dua_Latn	53392	atla1278	
tha_Thai	7735209	taik1256	yes	vmw_Latn	306899	atla1278		dzo_Tibt	52732	sino1245	
ukr_Cyrl	7462046	indo1319	yes	kwn_Latn	305362	atla1278		ctd_Latn	52135	sino1245	
tgl_Latn	7411064	aust1307	yes	pam_Latn	303737	aust1307		nnb_Latn	52041	atla1278	
sin_Sinh	7293178	indo1319	yes	seh_Latn	300243	atla1278		sxn_Latn	51749	aust1307	
gle_Latn	7225513	indo1319	yes	tsc_Latn	298442	atla1278		mps_Latn	50645	tebe1251	
hin_Deva	7046700	indo1319	yes	nyk_Latn	297976	atla1278		mny_Latn	50581	atla1278	
kor_Hang	6468444	kore1284	yes	kmb_Latn	296269	atla1278		gkp_Latn	50549	mand1469	
ory_Orya	6266475	indo1319		zai_Latn	277632	otom1299		kat_Latn	50424	kart1248	
urd_Arab	6009594	indo1319	yes	gym_Latn	274512	chib1249		bjn_Latn	49068	aust1307	
swa_Latn	5989369		yes	bod_Tibt	273489	sino1245		acr_Latn	48886	maya1287	
sqi_Latn	5526836	indo1319	yes	nde_Latn	269931	atla1278		dtp_Latn	48468	aust1307	
bel_Cyrl	5319675	indo1319	yes	fon_Latn	268566	atla1278		lam_Latn	46853	atla1278	
afr_Latn	5157787	indo1319	yes	ber_Latn	264426			bik_Latn	46561		
nno_Latn	4899103	indo1319		nbl_Latn	259158	atla1278		poh_Latn	46454	maya1287	
tat_Cyrl	4708088	turk1311		kmr_Latn	256677	indo1319		phm_Latn	45862	atla1278	

Table 11: List of languages used to train Glot500-m (Part I).

Language-Script	|Sent|	Family	Head	Language-Script	|Sent|	Family	Head	Language-Script	|Sent|	Family	Head
ast_Latn	4683554	indo1319		guc_Latn	249044	araw1281		hrx_Latn	45716	indo1319	
mon_Cyrl	4616960	mong1349	yes	mam_Latn	248348	maya1287		quh_Latn	45566	quec1387	
hbs_Cyrl	4598073	indo1319		nia_Latn	247406	aust1307		hyw_Cyrl	45379	indo1319	
hau_Latn	4368483	afro1255	yes	nyn_Latn	241992	atla1278		rue_Cyrl	45369	indo1319	
sna_Latn	4019596	atla1278		cab_Latn	240101	araw1281		eml_Latn	44630	indo1319	
msa_Latn	3929084		yes	top_Latn	239232	toto1251		acm_Arab	44505	afro1255	
som_Latn	3916769	afro1255	yes	tog_Latn	231969	atla1278		tob_Latn	44473	guai1249	
srp_Cyrl	3864091	indo1319	yes	mco_Latn	231209	mixe1284		ach_Latn	43974	nilo1247	
mlg_Latn	3715802		yes	tzh_Latn	230706	maya1287		vep_Latn	43076	ural1272	
zul_Latn	3580113	atla1278		pms_Latn	227748	indo1319		npi_Deva	43072	indo1319	
arz_Arab	3488224	afro1255		wuu_Hani	224088	sino1245		tok_Latn	42820	arti1236	
nya_Latn	3409030	atla1278		plt_Latn	220413	aust1307		sgs_Latn	42467	indo1319	
tam_Taml	3388255	drav1251	yes	yid_Hebr	220214	indo1319	yes	lij_Latn	42447	indo1319	
hat_Latn	3226932	indo1319		ada_Latn	219427	atla1278		myv_Cyrl	42147	ural1272	
uzb_Latn	3223485	turk1311	yes	iba_Latn	213615	aust1307		tih_Latn	41873	aust1307	
sot_Latn	3205510	atla1278		kek_Latn	209932	maya1287		tat_Latn	41640	turk1311	
uzb_Cyrl	3029947	turk1311		koo_Latn	209375	atla1278		lfn_Latn	41632	arti1236	
cos_Latn	3015055	indo1319		sop_Latn	206501	atla1278		cgg_Latn	41196	atla1278	
als_Latn	2954874	indo1319		kac_Latn	205542	sino1245		ful_Latn	41188	atla1278	
amh_Ethi	2862985	afro1255	yes	qvi_Latn	205447	quec1387		gor_Latn	41174	aust1307	
sun_Latn	2586011	aust1307	yes	cak_Latn	204472	maya1287		ile_Latn	40984	arti1236	
war_Latn	2584810	aust1307		kbp_Latn	202877	atla1278		ium_Latn	40683	hmon1336	
div_Thaa	2418687	indo1319		ctu_Latn	201662	maya1287		teo_Latn	40203	nilo1247	
yor_Latn	2392359	atla1278		kri_Latn	201087	indo1319		kia_Latn	40035	atla1278	
fao_Latn	2365271	indo1319		mau_Latn	199134	otom1299		crh_Cyrl	39985	turk1311	
uzn_Cyrl	2293672	turk1311		scn_Latn	199068	indo1319		crh_Latn	39896	turk1311	
smo_Latn	2290439	aust1307		tyv_Cyrl	198649	turk1311		enm_Latn	39809	indo1319	
bak_Cyrl	2264196	turk1311		ina_Latn	197315	arti1236		sat_Olck	39614	aust1305	
ilo_Latn	2106531	aust1307		btx_Latn	193701	aust1307		mad_Latn	38993	aust1307	
tso_Latn	2100708	atla1278		nch_Latn	193129	utoa1244		cac_Latn	38812	maya1287	
mri_Latn	2046850	aust1307		ncj_Latn	192962	utoa1244		hnj_Latn	38611	hmon1336	
hmn_Latn	1903898			pau_Latn	190529	aust1307		ksh_Latn	38130	indo1319	
asm_Beng	1882353	indo1319	yes	toj_Latn	189651	maya1287		ikk_Latn	38071	atla1278	
hil_Latn	1798875	aust1307		pcm_Latn	187594	indo1319		sba_Latn	38040	cent2225	
nso_Latn	1619354	atla1278		dyu_Latn	186367	mand1469		zom_Latn	37013	sino1245	
ibo_Latn	1543820	atla1278		kss_Latn	185868	atla1278		bqc_Latn	36881	mand1469	
kin_Latn	1521612	atla1278		afb_Arab	183694	afro1255		bim_Latn	36835	atla1278	
hye_Armn	1463123	indo1319	yes	urh_Latn	182214	atla1278		mdy_Ethi	36370	gong1255	
oci_Latn	1449128	indo1319		quc_Latn	181559	maya1287		bts_Latn	36216	aust1307	
lin_Latn	1408460	atla1278		new_Deva	181427	sino1245		gya_Latn	35902	atla1278	
tpi_Latn	1401844	indo1319		yao_Latn	179965	atla1278		ajg_Latn	35631	atla1278	
twi_Latn	1400979	atla1278		ngl_Latn	178498	atla1278		agw_Latn	35585	aust1307	
kir_Cyrl	1397566	turk1311	yes	nyu_Latn	177483	atla1278		kom_Cyrl	35249	ural1272	
pap_Latn	1360138	indo1319		kab_Latn	176015	afro1255		knv_Latn	35196		
nep_Deva	1317291	indo1319	yes	tuk_Cyrl	175769	turk1311		giz_Latn	35040	afro1255	
azj_Latn	1315834	turk1311		xmf_Geor	174994	kart1248		hui_Latn	34926	nucl1709	
bcl_Latn	1284493	aust1307		ndc_Latn	174305	atla1278		kpg_Latn	34900	aust1307	
xho_Latn	1262364	atla1278	yes	san_Deva	165616	indo1319	yes	zea_Latn	34426	indo1319	
cym_Latn	1244783	indo1319	yes	nba_Latn	163485	atla1278		aoj_Latn	34349	nucl1708	
gaa_Latn	1222307	atla1278		bpy_Beng	162838	indo1319		csy_Latn	34126	sino1245	
ton_Latn	1216118	aust1307		ncx_Latn	162558	utoa1244		azb_Arab	33758	turk1311	yes
tah_Latn	1190747	aust1307		qug_Latn	162500	quec1387		csb_Latn	33743	indo1319	
lat_Latn	1179913	indo1319	yes	rmn_Latn	162069	indo1319		tpm_Latn	33517	atla1278	
srn_Latn	1172349	indo1319		cjk_Latn	160645	atla1278		quw_Latn	33449	quec1387	
ewe_Latn	1161605	atla1278		arb_Arab	159884	afro1255	yes	rmy_Cyrl	33351	indo1319	
bem_Latn	1111969	atla1278		kea_Latn	158047	indo1319		ixl_Latn	33289	maya1287	
efi_Latn	1082621	atla1278		mck_Latn	157521	atla1278		mbb_Latn	33240	aust1307	
bis_Latn	1070170	indo1319		arn_Latn	155882	arau1255		pfl_Latn	33148	indo1319	
orm_Latn	1067699		yes	pdt_Latn	155485	indo1319		pcd_Latn	32867	indo1319	
haw_Latn	1062491	aust1307		her_Latn	154827	atla1278		tlh_Latn	32863	arti1236	
hmo_Latn	1033636	pidg1258		gla_Latn	152563	indo1319	yes	suz_Deva	32811	sino1245	
kat_Geor	1004297	kart1248	yes	kmr_Cyrl	151728	indo1319		gcr_Latn	32676	indo1319	
pag_Latn	983637	aust1307		mwl_Latn	150054	indo1319		jbo_Latn	32619	arti1236	
loz_Latn	964418	atla1278		nav_Latn	147702	atha1245		tbz_Latn	32264	atla1278	
fry_Latn	957422	indo1319	yes	ksw_Mymr	147674	sino1245		bam_Latn	32150	mand1469	
mya_Mymr	945180	sino1245	yes	mxv_Latn	147591	otom1299		prk_Latn	32085	aust1305	
nds_Latn	944715	indo1319		hif_Latn	147261	indo1319		jam_Latn	32048	indo1319	
run_Latn	943828	atla1278		wol_Latn	146992	atla1278		twx_Latn	32028	atla1278	

Table 12: List of languages used to train Glot500-m (Part II).

Language-Script	|Sent|	Family	Head	Language-Script	|Sent|	Family	Head	Language-Script	|Sent|	Family	Head
pnb_Arab	899895	indo1319		sme_Latn	146803	ural1272		nmf_Latn	31997	sino1245	
rar_Latn	894515	aust1307		gom_Latn	143937	indo1319		caq_Latn	31903	aust1305	
fij_Latn	887134	aust1307		bum_Latn	141673	atla1278		rop_Latn	31889	indo1319	
wls_Latn	882167	aust1307		mgr_Latn	138953	atla1278		tca_Latn	31852	ticu1244	
ckb_Arab	874441	indo1319		ahk_Latn	135068	sino1245		yan_Latn	31775	misu1242	
ven_Latn	860249	atla1278		kur_Arab	134160	indo1319		xav_Latn	31765	nucl1710	
zsm_Latn	859947	aust1307	yes	bas_Latn	133436	atla1278		bih_Deva	31658		
chv_Cyrl	859863	turk1311		bin_Latn	133256	atla1278		cuk_Latn	31612	chib1249	
lua_Latn	854359	atla1278		tsz_Latn	133251	tara1323		kjb_Latn	31471	maya1287	
que_Latn	838486			sid_Latn	130406	afro1255		hne_Deva	31465	indo1319	
sag_Latn	771048	atla1278		diq_Latn	128908	indo1319		wbm_Latn	31394	aust1305	
guw_Latn	767918	atla1278		srd_Latn	127064			zlm_Latn	31345	aust1307	
bre_Latn	748954	indo1319	yes	tcf_Latn	126050	otom1299		tui_Latn	31161	atla1278	
toi_Latn	745385	atla1278		bzj_Latn	124958	indo1319		ifb_Latn	30980	aust1307	
pus_Arab	731992	indo1319	yes	udm_Cyrl	121705	ural1272		izz_Latn	30894	atla1278	
che_Cyrl	728201	nakh1245		cce_Latn	120636	atla1278		rug_Latn	30857	aust1307	
pis_Latn	714783	indo1319		meu_Latn	120273	aust1307		aka_Latn	30704	atla1278	
kon_Latn	685194			chw_Latn	119751	atla1278		pxm_Latn	30698	book1242	
oss_Cyrl	683517	indo1319		cbk_Latn	118789	indo1319		kmm_Latn	30671	sino1245	
hyw_Armn	679819	indo1319		ibg_Latn	118733	aust1307		mcn_Latn	30666	afro1255	
iso_Latn	658789	atla1278		bhw_Latn	117381	aust1307		ifa_Latn	30621	aust1307	
nan_Latn	656389	sino1245		ngu_Latn	116851	utoa1244		dln_Latn	30620	sino1245	
lub_Latn	654390	atla1278		nyy_Latn	115914	atla1278		ext_Latn	30605	indo1319	
lim_Latn	652078	indo1319		szl_Latn	112496	indo1319		ksd_Latn	30550	aust1307	
tuk_Latn	649411	turk1311		ish_Latn	111814	atla1278		mzh_Latn	30517	mata1289	
tir_Ethi	649117	afro1255		naq_Latn	109747	khoe1240		llb_Latn	30480	atla1278	
tgk_Latn	636541	indo1319		toh_Latn	107583	atla1278		hra_Latn	30472	sino1245	
yua_Latn	610052	maya1287		ttj_Latn	106925	atla1278		mwm_Latn	30432	cent2225	
min_Latn	609065	aust1307		nse_Latn	105189	atla1278		krc_Cyrl	30353	turk1311	
lue_Latn	599429	atla1278		hsb_Latn	104802	indo1319		tuc_Latn	30349	aust1307	
khm_Khmr	590429	aust1305	yes	ami_Latn	104559	aust1307		mrw_Latn	30304	aust1307	
tum_Latn	589857	atla1278		alz_Latn	104392	nilo1247		pls_Latn	30136	otom1299	
tll_Latn	586530	atla1278		apc_Arab	102392	afro1255		rap_Latn	30102	aust1307	
ekk_Latn	582595	ural1272		vls_Latn	101900	indo1319		fur_Latn	30052	indo1319	
lug_Latn	566948	atla1278		mhr_Cyrl	100474	ural1272		kaa_Latn	30031	turk1311	
niu_Latn	566715	aust1307		djk_Latn	99234	indo1319		prs_Arab	26823	indo1319	yes
tzo_Latn	540262	maya1287		wes_Latn	98492	indo1319		san_Latn	25742	indo1319	yes
mah_Latn	534614	aust1307		gkn_Latn	97041	atla1278		som_Arab	14199	afro1255	yes
tvl_Latn	521556	aust1307		grc_Grek	96986	indo1319		uig_Latn	9637	turk1311	yes
jav_Latn	516833	aust1307	yes	hbo_Hebr	96484	afro1255		hau_Arab	9593	afro1255	yes

Table 13: List of languages used to train Glot500-m (Part III).

guages (Abate et al., 2018), Phontron (Neubig, 2011), QADI (Abdelali et al., 2021), Quechua-IIC (Zevallos et al., 2022), SLI_GalWeb.1.0 (Agerri et al., 2018), Shami (Abu Kwaik et al., 2018), Stanford NLP,[23] StatMT,[24] TICO (Anastasopoulos et al., 2020), TIL (Mirzakhalov et al., 2021), Tatoeba,[25] TeDDi (Moran et al., 2022), Tilde (Rozis and Skadiņš, 2017), W2C (Majliš, 2011), WAT (Nakazawa et al., 2022), WikiMatrix (Schwenk et al., 2021), Wikipedia,[26] Workshop on NER for South and South East Asian Languages (Singh, 2008), XLSum (Hasan et al., 2021).

D Results for Each Task and Language

We report the detailed results for all tasks and languages in Table 14 (Sentence Retrieval Tatoeba), 15, 16 (Sentence Retrieval Bible), 17 (NER), and 18 (POS), 19, 20 (Text Classification), 21, 22 (Round Trip Alignment).

E Perplexity Results for all Languages

Perplexity number for all languages is presented in Table 23, Table 24, and Table 25.

[23] https://nlp.stanford.edu/
[24] https://statmt.org/
[25] https://tatoeba.org/en/
[26] https://huggingface.co/datasets/wikipedia

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
afr_Latn	71.9	76.5	**81.1**	heb_Hebr	76.3	**84.1**	76.0	pam_Latn	4.8	5.6	**11.0**
amh_Ethi	35.1	37.5	**44.6**	hin_Deva	73.8	**88.8**	85.6	pes_Arab	83.3	86.6	**87.6**
ara_Arab	59.2	**66.8**	64.2	hrv_Latn	79.6	85.6	**89.8**	pms_Latn	16.6	12.6	**54.5**
arz_Arab	32.5	47.8	**63.5**	hsb_Latn	21.5	23.0	**53.6**	pol_Latn	82.6	**89.6**	82.4
ast_Latn	59.8	59.8	**87.4**	hun_Latn	76.1	**81.8**	69.2	por_Latn	91.0	**92.1**	90.1
aze_Latn	62.6	78.3	**79.9**	hye_Armn	64.6	40.0	**83.2**	ron_Latn	86.0	**89.1**	82.8
bel_Cyrl	70.0	80.5	**81.4**	ido_Latn	25.7	28.8	**57.6**	rus_Cyrl	89.6	**91.6**	91.5
ben_Beng	54.1	68.2	**69.4**	ile_Latn	34.6	41.9	**75.6**	slk_Latn	73.2	**80.6**	75.9
bos_Latn	78.5	82.2	**92.4**	ina_Latn	62.7	66.2	**91.4**	slv_Latn	72.1	**78.0**	77.0
bre_Latn	10.3	10.9	**19.9**	ind_Latn	84.3	**90.2**	88.8	spa_Latn	85.5	**89.0**	88.9
bul_Cyrl	84.4	**88.3**	86.7	isl_Latn	78.7	**84.5**	84.0	sqi_Latn	72.2	81.4	**84.7**
cat_Latn	72.8	73.9	**78.7**	ita_Latn	81.3	84.7	**86.4**	srp_Latn	78.1	85.0	**90.0**
cbk_Latn	33.2	36.0	**49.4**	jpn_Jpan	74.4	**80.8**	72.6	swe_Latn	90.4	**92.4**	89.7
ceb_Latn	15.2	15.0	**41.3**	kab_Latn	3.7	3.0	**16.4**	swh_Latn	30.3	34.6	**44.1**
ces_Latn	71.1	**81.3**	75.1	kat_Geor	61.1	**79.1**	67.7	tam_Taml	46.9	42.3	**66.4**
cmn_Hani	79.5	84.8	**85.6**	kaz_Cyrl	60.3	69.9	**72.3**	tat_Cyrl	10.3	10.3	**70.3**
csb_Latn	21.3	20.2	**40.3**	khm_Khmr	41.1	45.0	**52.5**	tel_Telu	58.5	50.4	**67.9**
cym_Latn	45.7	45.7	**55.7**	kor_Hang	73.4	**84.3**	78.0	tgl_Latn	47.6	54.2	**77.1**
dan_Latn	91.9	**93.9**	91.5	kur_Latn	24.1	28.5	**54.1**	tha_Thai	56.8	39.4	**78.1**
deu_Latn	**95.9**	94.7	95.0	lat_Latn	33.6	**48.0**	42.8	tuk_Latn	16.3	14.8	**63.5**
dtp_Latn	5.6	4.7	**21.1**	lfn_Latn	32.5	35.9	**59.3**	tur_Latn	77.9	**85.4**	78.4
ell_Grek	76.2	**84.1**	80.2	lit_Latn	73.4	**76.8**	65.6	uig_Arab	38.8	58.3	**62.6**
epo_Latn	64.9	68.5	**74.3**	lvs_Latn	73.4	**78.9**	76.9	ukr_Cyrl	77.1	**88.3**	83.7
est_Latn	63.9	68.6	**69.1**	mal_Mlym	80.1	**84.4**	83.8	urd_Arab	54.4	34.3	**80.9**
eus_Latn	45.9	**54.4**	52.7	mar_Deva	63.5	**81.2**	77.9	uzb_Cyrl	25.2	32.2	**64.5**
fao_Latn	45.0	42.7	**82.4**	mhr_Cyrl	6.5	5.8	**34.9**	vie_Latn	85.4	**87.9**	87.0
fin_Latn	81.9	**85.8**	72.3	mkd_Cyrl	70.5	**83.9**	81.4	war_Latn	8.0	6.5	**26.2**
fra_Latn	85.7	85.8	**86.0**	mon_Cyrl	60.9	**77.3**	77.0	wuu_Hani	56.1	47.4	**79.7**
fry_Latn	60.1	62.4	**75.1**	nds_Latn	28.8	29.0	**77.1**	xho_Latn	28.9	31.7	**56.3**
gla_Latn	21.0	21.2	**41.9**	nld_Latn	90.3	**91.8**	**91.8**	yid_Hebr	37.3	51.8	**74.4**
gle_Latn	32.0	36.9	**50.8**	nno_Latn	70.7	77.8	**87.8**	yue_Hani	50.3	42.3	**76.3**
glg_Latn	72.6	75.8	**77.5**	nob_Latn	93.5	**96.5**	95.7	zsm_Latn	81.4	87.4	**91.8**
gsw_Latn	36.8	31.6	**69.2**	oci_Latn	22.9	23.2	**46.9**				

Table 14: Top10 accuracy of XLM-R-B, XLM-R-L, and Glot500-m on Sentence Retrieval Tatoeba.

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
ace_Latn	4.4	4.6	**53.4**	iba_Latn	14.4	13.6	**66.0**	pan_Guru	43.2	**59.4**	48.8
ach_Latn	4.4	3.2	**40.0**	ibo_Latn	5.0	3.0	**30.4**	pap_Latn	12.4	9.2	**72.4**
acr_Latn	2.6	3.4	**25.4**	ifa_Latn	4.4	4.4	**39.2**	pau_Latn	4.4	4.0	**29.8**
afr_Latn	76.8	**77.2**	69.4	ifb_Latn	4.8	3.6	**36.6**	pcm_Latn	13.6	10.4	**66.8**
agw_Latn	5.8	3.0	**36.0**	ikk_Latn	3.0	3.2	**50.6**	pdt_Latn	9.2	8.6	**68.6**
ahk_Latn	3.0	2.6	**3.2**	ilo_Latn	6.2	3.6	**55.0**	pes_Arab	69.4	72.2	**80.8**
aka_Latn	5.0	4.2	**57.0**	ind_Latn	**82.6**	80.4	72.2	pis_Latn	6.4	5.0	**57.2**
aln_Latn	67.8	**72.4**	67.6	isl_Latn	62.6	**73.6**	66.0	pls_Latn	5.0	4.0	**34.4**
als_Latn	51.4	48.0	**55.8**	ita_Latn	**75.4**	73.6	70.0	plt_Latn	26.6	28.0	**59.8**
alt_Cyrl	12.6	9.0	**50.8**	ium_Latn	3.2	3.0	**24.8**	poh_Latn	3.4	2.4	**15.2**
alz_Latn	4.6	3.8	**34.6**	ixl_Latn	4.0	3.0	**18.4**	pol_Latn	79.2	**79.8**	63.8
amh_Ethi	35.4	43.2	**52.8**	izz_Latn	2.8	2.8	**25.6**	pon_Latn	5.6	4.4	**21.6**
aoj_Latn	5.0	3.0	**20.4**	jam_Latn	6.6	4.4	**67.8**	por_Latn	**81.6**	79.8	76.6
arb_Arab	7.0	7.8	**14.6**	jav_Latn	25.4	33.2	**47.4**	prk_Latn	3.6	2.2	**49.8**
arn_Latn	4.8	4.0	**28.4**	jpn_Jpan	65.0	**71.8**	64.2	prs_Arab	79.4	78.6	**88.8**
ary_Arab	2.8	4.0	**15.2**	kaa_Cyrl	17.6	24.8	**73.8**	pxm_Latn	3.2	3.2	**24.0**
arz_Arab	5.4	4.8	**24.8**	kaa_Latn	9.2	9.8	**43.4**	qub_Latn	4.6	3.6	**43.4**
asm_Beng	26.2	40.6	**66.6**	kab_Latn	3.4	2.4	**20.6**	quc_Latn	3.6	2.8	**24.8**
ayr_Latn	4.8	4.8	**52.8**	kac_Latn	3.6	3.2	**26.4**	qug_Latn	4.8	3.6	**50.8**
azb_Arab	7.4	6.8	**72.4**	kal_Latn	3.4	3.6	**23.2**	quh_Latn	4.6	4.4	**56.2**
aze_Latn	71.0	**78.6**	73.0	kan_Knda	51.2	**67.6**	50.2	quw_Latn	6.2	4.6	**49.2**
bak_Cyrl	5.4	6.4	**65.2**	kat_Geor	54.2	**61.4**	51.4	quy_Latn	4.6	4.6	**61.4**
bam_Latn	3.4	3.6	**60.2**	kaz_Cyrl	61.4	**73.0**	56.8	quz_Latn	4.8	4.2	**68.0**
ban_Latn	9.0	9.8	**33.0**	kbp_Latn	2.6	2.6	**36.0**	qvi_Latn	4.4	3.4	**46.8**
bar_Latn	13.4	12.8	**40.8**	kek_Latn	5.0	3.4	**26.4**	rap_Latn	3.2	3.2	**25.6**
bba_Latn	3.8	3.4	**36.8**	khm_Khmr	28.4	42.6	**47.6**	rar_Latn	3.2	3.0	**26.6**
bbc_Latn	7.8	7.4	**57.2**	kia_Latn	4.0	5.6	**33.2**	rmy_Latn	6.8	5.8	**34.6**
bci_Latn	4.4	3.6	**13.2**	kik_Latn	3.2	2.8	**53.4**	ron_Latn	**72.2**	69.6	66.6
bcl_Latn	10.2	11.2	**79.8**	kin_Latn	5.0	5.0	**59.4**	rop_Latn	4.6	3.4	**46.0**
bel_Cyrl	67.2	**72.8**	55.8	kir_Cyrl	54.8	**70.2**	66.6	rug_Latn	3.6	3.4	**49.0**
bem_Latn	6.6	5.4	**58.2**	kjb_Latn	4.0	3.8	**29.6**	run_Latn	5.4	6.4	**54.6**
ben_Beng	46.4	52.8	**53.4**	kjh_Cyrl	11.0	7.8	**53.8**	rus_Cyrl	**75.8**	74.6	71.2
bhw_Latn	4.4	6.0	**47.8**	kmm_Latn	4.8	3.8	**42.6**	sag_Latn	6.0	4.4	**52.4**
bim_Latn	4.2	2.8	**52.2**	kmr_Cyrl	4.0	4.2	**42.4**	sah_Cyrl	6.2	4.6	**45.8**
bis_Latn	7.0	4.6	**48.6**	kmr_Latn	35.8	40.4	**63.0**	san_Deva	13.8	14.2	**27.2**
bod_Tibt	2.0	1.8	**33.2**	knv_Latn	2.8	2.2	**9.0**	san_Latn	4.6	3.8	**9.8**
bqc_Latn	3.4	3.0	**39.2**	kor_Hang	64.0	**71.6**	61.2	sba_Latn	2.8	2.8	**37.6**
bre_Latn	17.6	23.4	**32.8**	kpg_Latn	5.2	3.8	**51.8**	seh_Latn	6.4	4.8	**74.6**
bts_Latn	6.0	5.0	**56.4**	krc_Cyrl	9.2	10.2	**63.0**	sin_Sinh	44.8	**56.6**	45.0
btx_Latn	11.0	9.0	**59.6**	kri_Latn	2.8	2.8	**62.8**	slk_Latn	**75.2**	72.8	63.6
bul_Cyrl	**81.2**	78.0	76.4	ksd_Latn	7.0	5.4	**42.6**	slv_Latn	63.6	**64.6**	51.8
bum_Latn	4.8	3.6	**38.0**	kss_Latn	2.2	2.4	**6.0**	sme_Latn	6.8	6.2	**47.8**
bzj_Latn	7.8	4.0	**75.0**	ksw_Mymr	1.6	2.0	**31.8**	smo_Latn	4.4	3.4	**36.0**
cab_Latn	5.8	4.6	**17.4**	kua_Latn	4.8	5.4	**43.8**	sna_Latn	7.0	3.6	**43.0**
cac_Latn	3.6	3.0	**14.8**	lam_Latn	4.6	3.6	**27.4**	snd_Arab	52.2	64.6	**66.6**
cak_Latn	3.4	3.4	**21.4**	lao_Laoo	31.4	**52.8**	49.6	som_Latn	22.2	29.0	**33.0**
caq_Latn	3.2	4.4	**30.2**	lat_Latn	52.2	**57.8**	49.6	sop_Latn	5.2	4.2	**31.2**
cat_Latn	**86.6**	81.0	76.4	lav_Latn	74.2	**78.0**	58.8	sot_Latn	6.0	4.8	**52.2**
cbk_Latn	31.8	35.6	**54.6**	ldi_Latn	5.4	4.4	**25.2**	spa_Latn	**81.2**	78.8	80.0
cce_Latn	5.2	4.6	**51.8**	leh_Latn	5.6	4.0	**58.2**	sqi_Latn	58.2	58.2	**63.4**
ceb_Latn	14.2	12.6	**68.0**	lhu_Latn	2.0	2.0	**5.0**	srm_Latn	4.0	3.2	**32.4**
ces_Latn	75.2	**75.8**	58.0	lin_Latn	6.6	5.4	**65.4**	srn_Latn	6.8	5.2	**79.8**
cfm_Latn	4.6	4.0	**46.8**	lit_Latn	**74.4**	71.6	62.4	srp_Cyrl	83.0	**87.0**	81.2
che_Cyrl	3.4	3.4	**14.0**	loz_Latn	6.8	4.6	**49.2**	srp_Latn	85.0	**87.2**	81.2
chk_Latn	5.4	4.2	**41.2**	ltz_Latn	9.8	10.0	**73.8**	ssw_Latn	4.8	8.4	**47.0**
chv_Cyrl	4.6	4.2	**56.0**	lug_Latn	4.6	4.0	**49.4**	sun_Latn	22.4	25.4	**43.0**
ckb_Arab	4.0	4.8	**47.2**	luo_Latn	6.4	4.4	**40.8**	suz_Deva	3.6	3.4	**34.2**
cmn_Hani	39.2	40.8	**41.8**	lus_Latn	3.8	3.8	**54.4**	swe_Latn	**79.8**	**79.8**	78.0
cnh_Latn	4.8	4.2	**55.6**	lzh_Hani	25.0	31.4	**63.4**	swh_Latn	47.8	48.8	**66.4**
crh_Cyrl	8.8	11.2	**75.2**	mad_Latn	7.6	4.4	**44.4**	sxn_Latn	4.8	4.8	**25.8**
crs_Latn	7.4	5.2	**80.6**	mah_Latn	4.8	4.2	**35.6**	tam_Taml	42.8	**56.8**	52.0
csy_Latn	3.8	5.0	**50.0**	mai_Deva	6.4	9.6	**59.2**	tat_Cyrl	8.2	6.2	**67.2**
ctd_Latn	4.2	5.4	**59.4**	mal_Mlym	49.4	**62.6**	56.8	tbz_Latn	2.6	2.6	**28.0**
ctu_Latn	2.8	2.8	**21.6**	mam_Latn	3.8	3.2	**12.8**	tca_Latn	2.4	3.2	**15.4**
cuk_Latn	5.0	3.4	**22.2**	mar_Deva	66.2	69.0	**74.8**	tdt_Latn	6.2	5.0	**62.2**
cym_Latn	38.8	**46.0**	42.4	mau_Latn	2.4	2.4	**3.6**	tel_Telu	44.4	**57.2**	42.6
dan_Latn	71.6	**73.2**	63.2	mbb_Latn	3.0	3.4	**33.6**	teo_Latn	5.8	3.4	**26.0**
deu_Latn	78.8	**80.6**	66.6	mck_Latn	5.2	3.6	**57.4**	tgk_Cyrl	4.6	4.2	**71.2**
djk_Latn	4.6	4.0	**40.4**	mcn_Latn	6.0	4.2	**39.2**	tgl_Latn	61.0	60.6	**78.6**
dln_Latn	5.2	4.8	**66.4**	mco_Latn	2.6	2.6	**7.0**	tha_Thai	30.0	37.0	**45.4**

Table 15: Top10 accuracy of XLM-R-B, XLM-R-L, and Glot500-m on Sentence Retrieval Bible (Part I).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
dtp_Latn	5.4	4.2	**24.2**	mdy_Ethi	2.8	2.4	**31.6**	tih_Latn	5.2	4.4	**51.6**
dyu_Latn	4.2	2.4	**50.2**	meu_Latn	5.6	4.4	**52.0**	tir_Ethi	7.4	6.2	**43.4**
dzo_Tibt	2.2	2.0	**36.4**	mfe_Latn	9.0	6.8	**78.6**	tlh_Latn	7.8	6.4	**72.4**
efi_Latn	4.4	4.2	**54.0**	mgh_Latn	5.2	3.4	**23.6**	tob_Latn	2.2	3.0	**16.8**
ell_Grek	52.6	**53.8**	48.6	mgr_Latn	4.0	4.4	**57.6**	toh_Latn	4.0	4.0	**47.2**
enm_Latn	39.8	39.2	**66.0**	mhr_Cyrl	6.6	5.4	**48.0**	toi_Latn	4.2	4.4	**47.4**
epo_Latn	**64.6**	59.8	56.2	min_Latn	9.4	6.2	**29.0**	toj_Latn	4.2	4.0	**15.6**
est_Latn	72.0	**75.6**	56.4	miq_Latn	4.4	4.4	**47.4**	ton_Latn	4.2	3.8	**22.4**
eus_Latn	26.2	**28.4**	23.0	mkd_Cyrl	**76.6**	72.6	74.8	top_Latn	3.4	3.6	**8.0**
ewe_Latn	4.6	3.0	**49.0**	mlg_Latn	29.0	28.4	**66.0**	tpi_Latn	5.8	4.4	**58.0**
fao_Latn	24.0	28.4	**73.4**	mlt_Latn	5.8	5.2	**50.4**	tpm_Latn	3.6	3.0	**39.6**
fas_Arab	78.2	80.4	**89.2**	mos_Latn	4.2	3.6	**42.8**	tsn_Latn	5.4	3.6	**41.8**
fij_Latn	3.8	3.0	**36.4**	mps_Latn	3.2	3.2	**21.6**	tso_Latn	5.6	5.0	**50.8**
fil_Latn	60.4	64.4	**72.0**	mri_Latn	4.2	3.8	**48.4**	tsz_Latn	5.6	3.2	**27.0**
fin_Latn	**75.6**	75.0	53.8	mrw_Latn	6.0	4.4	**52.2**	tuc_Latn	2.6	2.6	**31.4**
fon_Latn	2.6	2.0	**33.4**	msa_Latn	40.0	40.2	**40.6**	tui_Latn	3.6	3.2	**38.0**
fra_Latn	**88.6**	86.8	79.2	mwm_Latn	2.6	2.6	**35.8**	tuk_Cyrl	13.6	15.8	**65.0**
fry_Latn	27.8	27.4	**44.0**	mxv_Latn	3.0	3.4	**8.8**	tuk_Latn	9.6	9.6	**66.2**
gaa_Latn	3.8	3.4	**47.0**	mya_Mymr	20.2	27.8	**29.4**	tum_Latn	5.2	4.6	**66.2**
gil_Latn	5.6	3.6	**36.8**	myv_Cyrl	4.6	4.0	**35.0**	tur_Latn	74.4	**74.8**	63.2
giz_Latn	6.2	4.0	**41.0**	mzh_Latn	4.6	3.2	**36.2**	twi_Latn	3.8	3.0	**50.0**
gkn_Latn	4.0	3.4	**32.2**	nan_Latn	3.2	3.2	**13.6**	tyv_Cyrl	6.8	7.0	**46.6**
gkp_Latn	3.0	3.2	**20.4**	naq_Latn	3.0	2.2	**25.0**	tzh_Latn	6.0	5.2	**25.8**
gla_Latn	25.2	26.6	**43.0**	nav_Latn	2.4	2.8	**11.2**	tzo_Latn	3.8	3.8	**16.6**
gle_Latn	35.0	38.6	**40.0**	nbl_Latn	9.2	11.8	**53.8**	udm_Cyrl	6.0	5.0	**55.2**
glv_Latn	5.8	3.6	**47.4**	nch_Latn	4.4	3.0	**21.4**	uig_Arab	45.8	**63.6**	56.2
gom_Latn	6.0	4.6	**42.8**	ncj_Latn	4.6	3.0	**25.2**	uig_Latn	9.8	11.0	**62.8**
gor_Latn	3.8	3.0	**26.0**	ndc_Latn	5.2	4.6	**40.0**	ukr_Cyrl	**66.0**	63.4	57.0
grc_Grek	17.4	23.8	**54.8**	nde_Latn	13.0	15.2	**53.8**	urd_Arab	47.6	47.0	**65.0**
guc_Latn	3.4	2.6	**13.0**	ndo_Latn	5.2	4.0	**48.2**	uzb_Cyrl	6.2	7.4	**78.8**
gug_Latn	4.6	3.2	**36.0**	nds_Latn	9.6	8.4	**43.0**	uzb_Latn	54.8	60.8	**67.6**
guj_Gujr	53.8	71.2	**71.4**	nep_Deva	35.6	50.6	**58.6**	uzn_Cyrl	5.4	5.4	**87.0**
gur_Latn	3.8	2.8	**27.0**	ngu_Latn	4.6	3.4	**27.6**	ven_Latn	4.8	4.2	**47.2**
guw_Latn	4.0	3.4	**59.4**	nia_Latn	4.6	3.2	**29.4**	vie_Latn	**72.8**	71.0	57.8
gya_Latn	3.6	3.0	**41.0**	nld_Latn	**78.0**	75.8	71.8	wal_Latn	4.2	5.4	**51.4**
gym_Latn	3.6	3.8	**18.0**	nmf_Latn	4.6	4.6	**36.6**	war_Latn	9.8	6.6	**43.4**
hat_Latn	6.0	4.2	**68.2**	nnb_Latn	3.6	3.2	**42.0**	wbm_Latn	3.8	2.4	**46.4**
hau_Latn	28.8	36.0	**54.8**	nno_Latn	58.4	67.2	**72.6**	wol_Latn	4.6	4.4	**35.8**
haw_Latn	4.2	3.4	**38.8**	nob_Latn	82.8	**85.2**	79.2	xav_Latn	2.2	2.4	**5.0**
heb_Hebr	25.0	**26.0**	21.8	nor_Latn	81.2	84.2	**86.2**	xho_Latn	10.4	16.2	**40.8**
hif_Latn	12.2	16.4	**39.0**	npi_Deva	50.6	70.8	**76.6**	yan_Latn	4.2	3.4	**31.8**
hil_Latn	11.0	10.8	**76.2**	nse_Latn	5.2	5.0	**54.8**	yao_Latn	4.4	3.8	**55.2**
hin_Deva	67.0	72.8	**76.6**	nso_Latn	6.0	4.2	**57.0**	yap_Latn	4.0	4.0	**24.0**
hin_Latn	13.6	16.0	**43.2**	nya_Latn	4.0	4.6	**60.2**	yom_Latn	4.8	3.6	**42.2**
hmo_Latn	6.4	4.4	**48.2**	nyn_Latn	4.4	4.2	**51.8**	yor_Latn	3.4	3.6	**37.4**
hne_Deva	13.4	14.8	**75.0**	nyy_Latn	3.0	3.0	**25.6**	yua_Latn	3.8	3.4	**18.2**
hnj_Latn	2.8	2.8	**54.2**	nzi_Latn	3.2	3.0	**47.2**	yue_Hani	17.2	14.0	**24.0**
hra_Latn	5.2	4.6	**52.2**	ori_Orya	42.6	**62.0**	57.0	zai_Latn	6.2	4.2	**38.0**
hrv_Latn	79.8	**81.8**	72.6	ory_Orya	31.4	47.0	**55.2**	zho_Hani	40.4	40.2	**44.4**
hui_Latn	3.8	3.0	**28.0**	oss_Cyrl	4.2	3.6	**54.8**	zlm_Latn	83.4	78.4	**87.0**
hun_Latn	76.4	**78.2**	56.2	ote_Latn	3.6	2.4	**18.0**	zom_Latn	3.6	3.4	**50.2**
hus_Latn	3.6	3.2	**17.6**	pag_Latn	8.0	5.0	**61.2**	zsm_Latn	90.2	**91.0**	83.0
hye_Armn	30.8	33.0	**75.2**	pam_Latn	8.2	7.0	**49.8**	zul_Latn	11.0	16.0	**49.0**

Table 16: Top10 accuracy of XLM-R-B, XLM-R-L, and Glot500-m on Sentence Retrieval Bible (Part II).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
ace_Latn	33.4	38.9	**44.2**	heb_Hebr	51.5	**56.5**	49.0	ori_Orya	**31.4**	27.6	31.0
afr_Latn	75.6	**78.3**	76.7	hin_Deva	67.0	**71.1**	69.4	oss_Cyrl	33.7	39.2	**52.1**
als_Latn	60.7	61.4	**80.0**	hrv_Latn	77.2	**78.9**	77.3	pan_Guru	50.0	**50.5**	48.1
amh_Ethi	42.2	40.9	**45.4**	hsb_Latn	64.0	69.0	**71.2**	pms_Latn	71.2	74.9	**75.9**
ara_Arab	44.7	48.7	**56.1**	hun_Latn	76.2	**79.8**	75.9	pnb_Arab	57.0	64.6	**65.8**
arg_Latn	73.6	74.6	**77.2**	hye_Armn	50.8	**61.7**	54.8	pol_Latn	77.5	**81.2**	78.1
arz_Arab	48.3	52.5	**57.4**	ibo_Latn	40.8	42.8	**58.6**	por_Latn	77.8	**81.2**	78.6
asm_Beng	53.2	**64.4**	64.2	ido_Latn	61.6	**78.6**	77.8	pus_Arab	37.4	39.9	**41.4**
ast_Latn	78.1	82.8	**84.5**	ilo_Latn	55.3	65.3	**77.1**	que_Latn	59.1	55.2	**66.8**
aym_Latn	40.8	38.7	**47.1**	ina_Latn	54.7	**63.4**	58.0	roh_Latn	52.6	55.7	**60.3**
aze_Latn	62.4	**69.2**	66.1	ind_Latn	49.0	54.1	**56.6**	ron_Latn	74.8	**79.9**	74.2
bak_Cyrl	35.1	49.3	**59.4**	isl_Latn	69.1	**77.2**	72.1	rus_Cyrl	63.8	**70.0**	67.6
bar_Latn	55.2	58.6	**68.4**	ita_Latn	77.3	**81.2**	78.7	sah_Cyrl	47.3	49.7	**74.2**
bel_Cyrl	74.2	**78.7**	74.3	jav_Latn	58.4	**61.2**	55.8	san_Deva	36.9	**37.3**	35.8
ben_Beng	65.3	**75.8**	71.6	jbo_Latn	18.0	26.3	**27.8**	scn_Latn	49.9	54.8	**65.8**
bih_Deva	50.7	57.1	**58.7**	jpn_Jpan	19.7	**20.6**	17.2	sco_Latn	80.9	81.8	**85.6**
bod_Tibt	2.5	3.0	**31.6**	kan_Knda	56.9	**60.8**	58.4	sgs_Latn	42.5	47.4	**62.7**
bos_Latn	74.0	**74.3**	74.2	kat_Geor	65.5	**69.5**	68.3	sin_Sinh	52.2	57.0	**57.8**
bre_Latn	59.1	**63.9**	63.3	kaz_Cyrl	43.7	**52.7**	50.0	slk_Latn	75.0	**81.7**	78.5
bul_Cyrl	76.8	**81.6**	77.2	khm_Khmr	43.3	**46.2**	40.6	slv_Latn	79.4	**82.2**	80.1
cat_Latn	82.2	**85.4**	83.7	kin_Latn	60.5	58.4	**67.1**	snd_Arab	41.2	**46.6**	41.8
cbk_Latn	**54.6**	54.0	54.1	kir_Cyrl	44.2	**46.9**	46.7	som_Latn	55.8	55.5	**58.2**
ceb_Latn	55.1	**57.8**	53.8	kor_Hang	49.1	**58.5**	50.9	spa_Latn	72.8	**73.3**	72.8
ces_Latn	77.6	**80.8**	78.3	ksh_Latn	41.3	48.3	**58.7**	sqi_Latn	74.0	74.4	**76.6**
che_Cyrl	15.4	24.6	**60.9**	kur_Latn	58.8	65.0	**69.6**	srp_Cyrl	59.7	**71.4**	66.4
chv_Cyrl	52.9	51.6	**75.9**	lat_Latn	70.7	**79.2**	73.8	sun_Latn	42.0	49.7	**57.7**
ckb_Arab	33.1	42.6	**75.5**	lav_Latn	73.4	**77.1**	74.0	swa_Latn	65.6	69.0	**69.6**
cos_Latn	54.3	**56.4**	56.0	lij_Latn	36.9	41.6	**46.6**	swe_Latn	71.8	**75.9**	69.7
crh_Latn	44.3	52.4	**54.7**	lim_Latn	59.9	64.7	**71.8**	szl_Latn	58.2	56.7	**67.6**
csb_Latn	55.1	54.2	**61.2**	lin_Latn	37.4	41.3	**54.0**	tam_Taml	55.0	**57.9**	55.2
cym_Latn	57.9	**60.1**	59.7	lit_Latn	73.4	**77.0**	73.5	tat_Cyrl	40.7	47.7	**68.0**
dan_Latn	81.5	**84.2**	81.7	lmo_Latn	68.8	68.4	**71.3**	tel_Telu	47.4	**52.5**	46.0
deu_Latn	74.3	**78.6**	75.7	ltz_Latn	47.4	55.8	**69.1**	tgk_Cyrl	24.7	38.3	**68.5**
diq_Latn	37.8	43.3	**53.1**	lzh_Hani	15.6	**21.6**	11.8	tgl_Latn	71.0	74.7	**75.1**
div_Thaa	0.0	0.0	**51.1**	mal_Mlym	61.0	**63.3**	61.3	tha_Thai	**4.2**	1.6	3.2
ell_Grek	73.7	**78.6**	72.8	mar_Deva	60.2	**63.4**	60.7	tuk_Latn	45.6	50.7	**59.7**
eml_Latn	32.9	36.1	**40.8**	mhr_Cyrl	44.3	48.3	**63.1**	tur_Latn	74.9	**79.3**	76.1
eng_Latn	82.7	**84.5**	83.3	min_Latn	42.9	**46.2**	41.8	uig_Arab	44.0	**50.9**	48.0
epo_Latn	63.8	**71.8**	68.0	mkd_Cyrl	74.5	**80.4**	73.3	ukr_Cyrl	75.2	**76.3**	74.2
est_Latn	72.2	**78.5**	73.5	mlg_Latn	54.9	54.3	**57.9**	urd_Arab	51.2	57.8	**74.5**
eus_Latn	59.0	**62.0**	58.0	mlt_Latn	43.2	48.3	**73.3**	uzb_Latn	70.6	**76.2**	75.1
ext_Latn	36.9	**47.1**	46.1	mon_Cyrl	72.4	**74.3**	66.9	vec_Latn	59.0	63.3	**66.4**
fao_Latn	61.1	70.8	**72.4**	mri_Latn	14.2	18.3	**53.5**	vep_Latn	59.8	59.3	**71.3**
fas_Arab	44.6	**58.0**	51.2	msa_Latn	62.3	**70.4**	65.8	vie_Latn	68.5	**77.8**	71.3
fin_Latn	75.5	**79.1**	75.2	mwl_Latn	42.6	**47.5**	45.3	vls_Latn	68.1	73.6	**73.7**
fra_Latn	77.2	**79.8**	76.0	mya_Mymr	51.3	53.4	**55.5**	vol_Latn	59.2	55.6	**59.2**
frr_Latn	45.4	46.8	**54.8**	mzn_Arab	36.4	43.1	**44.9**	war_Latn	61.9	61.4	**66.1**
fry_Latn	74.3	**79.0**	77.5	nan_Latn	46.2	51.4	**82.1**	wuu_Hani	29.4	**54.0**	25.1
fur_Latn	44.9	50.1	**56.4**	nap_Latn	53.0	53.9	**55.7**	xmf_Geor	40.2	40.0	**62.6**
gla_Latn	55.5	61.4	**63.5**	nds_Latn	62.4	66.7	**77.1**	yid_Hebr	47.6	**52.5**	50.3
gle_Latn	70.8	**74.6**	72.2	nep_Deva	63.2	**66.4**	62.7	yor_Latn	42.2	40.1	**63.1**
glg_Latn	80.2	**81.1**	79.4	nld_Latn	80.1	**83.6**	80.8	yue_Hani	24.8	**30.3**	22.6
grn_Latn	40.0	42.3	**54.7**	nno_Latn	76.6	**80.4**	78.0	zea_Latn	65.2	67.4	**68.6**
guj_Gujr	61.0	**61.9**	59.8	nor_Latn	76.5	**80.1**	76.7	zho_Hani	24.2	**28.8**	23.4
hbs_Latn	61.1	57.2	**61.5**	oci_Latn	65.3	67.8	**70.1**				

Table 17: F1 of XLM-R-B, XLM-R-L, and Glot500-m on NER.

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
afr_Latn	88.7	**89.3**	87.5	hbo_Hebr	38.9	45.7	**54.2**	pol_Latn	84.7	**85.4**	82.4
ajp_Arab	62.9	67.3	**69.7**	heb_Hebr	68.0	**69.2**	67.2	por_Latn	88.6	**89.8**	88.2
aln_Latn	53.5	**60.4**	52.3	hin_Deva	71.3	**75.3**	70.3	quc_Latn	28.9	29.3	**62.4**
amh_Ethi	64.5	**66.2**	66.1	hrv_Latn	85.9	**86.2**	85.5	ron_Latn	83.9	**85.7**	80.6
ara_Arab	68.5	**69.7**	65.4	hsb_Latn	71.5	74.4	**83.6**	rus_Cyrl	89.1	**89.7**	88.7
bam_Latn	25.4	23.5	**40.8**	hun_Latn	82.6	**82.7**	81.2	sah_Cyrl	20.3	22.8	**76.8**
bel_Cyrl	86.2	**86.2**	86.0	hye_Armn	85.2	**86.5**	84.0	san_Deva	18.3	**28.6**	26.1
ben_Beng	82.8	**83.8**	83.8	hyw_Armn	78.5	**82.5**	80.4	sin_Sinh	57.7	**60.1**	54.7
bre_Latn	61.6	**66.6**	60.7	ind_Latn	83.5	**84.1**	82.7	slk_Latn	85.6	**85.8**	84.4
bul_Cyrl	**89.1**	88.9	88.1	isl_Latn	84.2	**85.1**	82.8	slv_Latn	78.5	**79.1**	75.9
cat_Latn	86.7	**87.9**	86.3	ita_Latn	88.3	**89.6**	87.3	sme_Latn	29.8	31.5	**73.7**
ceb_Latn	49.3	49.5	**66.4**	jav_Latn	73.2	**76.7**	74.1	spa_Latn	88.5	**89.0**	88.0
ces_Latn	85.0	**85.4**	84.4	jpn_Jpan	17.3	**32.2**	31.7	sqi_Latn	81.4	**82.9**	77.9
cym_Latn	65.5	**67.0**	64.4	kaz_Cyrl	77.3	**79.1**	75.9	srp_Latn	86.1	**86.6**	85.3
dan_Latn	90.7	**91.0**	90.2	kmr_Latn	73.1	**78.2**	75.5	swe_Latn	93.5	**93.7**	92.1
deu_Latn	**88.4**	88.4	87.9	kor_Hang	**53.7**	53.4	53.1	tam_Taml	76.1	**76.9**	75.0
ell_Grek	**87.3**	87.0	85.4	lat_Latn	75.0	**80.3**	72.4	tat_Cyrl	45.0	48.8	**70.1**
eng_Latn	96.3	**96.5**	96.0	lav_Latn	86.0	**86.3**	83.5	tel_Telu	**85.0**	85.0	82.2
est_Latn	86.1	**86.4**	83.1	lij_Latn	48.1	48.6	**76.8**	tgl_Latn	72.7	**74.8**	74.7
eus_Latn	71.3	**73.7**	61.8	lit_Latn	84.1	**84.6**	81.1	tha_Thai	46.0	54.7	**56.7**
fao_Latn	77.0	80.6	**89.2**	lzh_Hani	14.1	**23.1**	23.0	tur_Latn	72.9	**74.0**	70.7
fas_Arab	71.8	**74.2**	71.5	mal_Mlym	**86.9**	86.7	84.4	uig_Arab	68.2	**70.2**	68.9
fin_Latn	85.2	**85.7**	80.8	mar_Deva	83.0	**85.2**	80.8	ukr_Cyrl	85.9	**86.3**	84.8
fra_Latn	86.7	**87.3**	85.4	mlt_Latn	21.0	21.9	**79.5**	urd_Arab	61.0	**68.2**	62.0
gla_Latn	57.4	**61.8**	60.2	myv_Cyrl	39.7	38.6	**65.7**	vie_Latn	70.9	**72.2**	67.1
gle_Latn	65.5	**68.7**	64.4	nap_Latn	52.8	17.0	**63.6**	wol_Latn	25.6	25.5	**61.6**
glg_Latn	83.7	**86.4**	82.6	nds_Latn	58.0	67.3	**77.2**	xav_Latn	8.4	5.3	**14.0**
glv_Latn	27.5	29.5	**52.7**	nld_Latn	88.5	**88.8**	88.2	yor_Latn	21.7	21.4	**63.9**
grc_Grek	62.0	68.1	**73.1**	nor_Latn	88.1	**88.9**	88.0	yue_Hani	31.5	**42.0**	40.9
grn_Latn	8.9	7.8	**19.8**	pcm_Latn	47.3	50.1	**57.1**	zho_Hani	28.6	42.4	**43.1**
gsw_Latn	48.7	55.9	**80.3**								

Table 18: F1 of XLM-R-B, XLM-R-L, and Glot500-m on POS.

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
ace_Latn	15	25	**60**	iba_Latn	30	35	**56**	ote_Latn	6	5	**36**
ace_Latn	15	25	**60**	iba_Latn	30	35	**56**	ote_Latn	6	5	**36**
ach_Latn	9	8	**34**	ibo_Latn	8	6	**51**	pag_Latn	22	21	**52**
acr_Latn	10	8	**46**	ifa_Latn	12	12	**47**	pam_Latn	20	18	**41**
afr_Latn	54	**64**	57	ifb_Latn	14	11	**48**	pan_Guru	53	**65**	59
agw_Latn	11	13	**54**	ikk_Latn	11	7	**47**	pap_Latn	31	36	**55**
ahk_Latn	5	5	**24**	ilo_Latn	15	13	**52**	pau_Latn	12	10	**41**
aka_Latn	11	7	**48**	ind_Latn	62	**66**	63	pcm_Latn	25	28	**46**
aln_Latn	44	**51**	49	isl_Latn	50	**60**	49	pdt_Latn	17	20	**53**
als_Latn	45	**51**	50	ita_Latn	57	**68**	61	pes_Arab	60	**70**	64
alt_Cyrl	25	23	**54**	ium_Latn	6	7	**53**	pis_Latn	13	13	**57**
alz_Latn	13	11	**34**	ixl_Latn	10	7	**33**	pls_Latn	6	7	**41**
amh_Ethi	42	**49**	43	izz_Latn	9	6	**41**	plt_Latn	30	**51**	50
aoj_Latn	12	9	**41**	jam_Latn	15	14	**55**	poh_Latn	16	8	**48**
arb_Arab	27	**55**	45	jav_Latn	44	**54**	49	pol_Latn	53	**63**	47
arn_Latn	9	8	**46**	jpn_Jpan	56	**66**	56	pon_Latn	10	8	**50**
ary_Arab	16	27	**40**	kaa_Cyrl	35	49	**59**	por_Latn	61	**67**	57
arz_Arab	28	**49**	39	kab_Latn	8	7	**30**	prk_Latn	6	6	**51**
asm_Beng	44	**53**	**53**	kac_Latn	7	8	**44**	prs_Arab	62	**67**	65
ayr_Latn	11	9	**53**	kal_Latn	9	7	**33**	pxm_Latn	9	9	**43**
azb_Arab	19	17	**55**	kan_Knda	53	**63**	59	qub_Latn	13	10	**55**
aze_Latn	56	**64**	61	kat_Geor	55	**60**	57	quc_Latn	9	7	**45**
bak_Cyrl	17	19	**57**	kaz_Cyrl	53	**64**	56	qug_Latn	13	8	**59**
bam_Latn	7	7	**46**	kbp_Latn	5	5	**35**	quh_Latn	11	10	**56**
ban_Latn	21	24	**46**	kek_Latn	6	9	**45**	quw_Latn	13	10	**48**
bar_Latn	31	42	**45**	khm_Khmr	51	**64**	59	quy_Latn	12	11	**57**
bba_Latn	6	6	**42**	kia_Latn	7	7	**39**	quz_Latn	11	8	**56**
bci_Latn	9	8	**28**	kik_Latn	7	6	**40**	qvi_Latn	9	8	**59**
bcl_Latn	28	27	**51**	kin_Latn	17	9	**50**	rap_Latn	8	7	**50**
bel_Cyrl	56	**67**	54	kir_Cyrl	55	**63**	60	rar_Latn	8	9	**48**
bem_Latn	13	14	**43**	kjb_Latn	7	9	**48**	rmy_Latn	16	12	**47**
ben_Beng	53	**65**	60	kjh_Cyrl	15	19	**50**	ron_Latn	60	**70**	60
bhw_Latn	11	11	**47**	kmm_Latn	8	6	**46**	rop_Latn	10	10	**50**
bim_Latn	7	7	**47**	kmr_Cyrl	8	8	**44**	rug_Latn	7	7	**55**
bis_Latn	13	12	**57**	knv_Latn	7	6	**44**	run_Latn	16	9	**49**
bqc_Latn	7	7	**36**	kor_Hang	59	**70**	60	rus_Cyrl	60	**66**	61
bre_Latn	30	**49**	36	kpg_Latn	9	10	**57**	sag_Latn	9	11	**42**
bts_Latn	18	17	**56**	krc_Cyrl	25	22	**56**	sah_Cyrl	10	9	**52**
btx_Latn	23	26	**53**	kri_Latn	7	9	**52**	sba_Latn	7	6	**41**
bul_Cyrl	61	**70**	57	ksd_Latn	10	11	**53**	seh_Latn	11	8	**47**
bum_Latn	9	9	**43**	kss_Latn	5	5	**23**	sin_Sinh	54	**66**	59
bzj_Latn	18	14	**56**	ksw_Mymr	5	5	**53**	slk_Latn	56	**63**	56
cab_Latn	9	8	**41**	kua_Latn	12	12	**45**	slv_Latn	59	**66**	61
cac_Latn	10	10	**47**	lam_Latn	5	8	**28**	sme_Latn	10	12	**43**
cak_Latn	7	8	**53**	lao_Laoo	56	**66**	64	smo_Latn	8	7	**51**
caq_Latn	7	7	**47**	lat_Latn	56	**64**	50	sna_Latn	13	11	**42**
cat_Latn	53	**64**	48	lav_Latn	54	**66**	55	snd_Arab	54	**64**	57
cbk_Latn	43	47	**57**	ldi_Latn	8	9	**28**	som_Latn	32	**45**	33
cce_Latn	13	9	**47**	leh_Latn	13	10	**44**	sop_Latn	12	8	**32**
ceb_Latn	28	30	**49**	lhu_Latn	6	6	**30**	sot_Latn	11	8	**45**
ces_Latn	50	**65**	53	lin_Latn	10	7	**49**	spa_Latn	61	**69**	60
cfm_Latn	8	8	**55**	lit_Latn	54	**66**	53	sqi_Latn	57	**68**	60
che_Cyrl	11	6	**20**	loz_Latn	10	10	**48**	srm_Latn	10	9	**53**
chv_Cyrl	8	7	**52**	ltz_Latn	22	30	**52**	srn_Latn	10	9	**53**
cmn_Hani	53	**62**	56	lug_Latn	16	9	**45**	srp_Latn	55	**67**	56
cnh_Latn	7	8	**56**	luo_Latn	12	10	**39**	ssw_Latn	14	17	**40**
crh_Cyrl	22	31	**57**	lus_Latn	11	7	**52**	sun_Latn	40	**47**	**47**
crs_Latn	14	17	**61**	lzh_Hani	46	**55**	**55**	suz_Deva	15	13	**53**
csy_Latn	9	7	**52**	mad_Latn	23	28	**56**	swe_Latn	60	**66**	56
ctd_Latn	9	8	**56**	mah_Latn	6	6	**42**	swh_Latn	47	**59**	56
ctu_Latn	15	14	**51**	mai_Deva	34	39	**59**	sxn_Latn	11	8	**46**
cuk_Latn	15	7	**44**	mal_Mlym	56	**64**	60	tam_Taml	56	**61**	60
cym_Latn	46	**51**	48	mam_Latn	10	6	**31**	tat_Cyrl	21	28	**64**
dan_Latn	51	**62**	50	mar_Deva	55	**63**	60	tbz_Latn	6	6	**43**
deu_Latn	56	**65**	53	mau_Latn	5	5	**6**	tca_Latn	5	5	**47**
djk_Latn	12	10	**46**	mbb_Latn	11	7	**48**	tdt_Latn	16	13	**56**
dln_Latn	10	5	**52**	mck_Latn	15	10	**41**	tel_Telu	55	**65**	60
dtp_Latn	9	8	**39**	mcn_Latn	13	9	**43**	teo_Latn	12	8	**26**
dyu_Latn	6	8	**52**	mco_Latn	6	7	**28**	tgk_Cyrl	10	7	**55**
dzo_Tibt	6	5	**55**	mdy_Ethi	6	7	**47**	tgl_Latn	48	**60**	56

Table 19: F1 of XLM-R-B, XLM-R-L, and Glot500-m on Text Classification (Part I).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
efi_Latn	10	9	**50**	meu_Latn	15	11	**52**	tha_Thai	56	**67**	61
ell_Grek	37	47	**54**	mfe_Latn	16	14	**61**	tih_Latn	11	11	**56**
eng_Latn	74	**75**	68	mgh_Latn	10	6	**35**	tir_Ethi	23	27	**48**
enm_Latn	46	56	**65**	mgr_Latn	14	12	**46**	tlh_Latn	30	26	**59**
epo_Latn	53	**63**	53	mhr_Cyrl	14	10	**43**	tob_Latn	6	9	**52**
est_Latn	62	**68**	53	min_Latn	27	37	**50**	toh_Latn	11	8	**41**
eus_Latn	28	**33**	22	miq_Latn	7	7	**48**	toi_Latn	14	10	**40**
ewe_Latn	9	9	**52**	mkd_Cyrl	65	**69**	61	toj_Latn	12	11	**42**
fao_Latn	33	41	**55**	mlg_Latn	32	**51**	48	ton_Latn	6	7	**47**
fas_Arab	62	**68**	62	mlt_Latn	12	11	**49**	top_Latn	11	10	**25**
fij_Latn	8	7	**51**	mos_Latn	7	8	**41**	tpi_Latn	11	13	**55**
fil_Latn	47	**56**	53	mps_Latn	11	12	**54**	tpm_Latn	9	8	**47**
fin_Latn	57	**66**	56	mri_Latn	9	8	**47**	tsn_Latn	11	8	**45**
fon_Latn	5	6	**49**	mrw_Latn	15	18	**41**	tsz_Latn	10	10	**45**
fra_Latn	57	**66**	57	msa_Latn	43	**49**	46	tuc_Latn	7	9	**50**
fry_Latn	31	34	**37**	mwm_Latn	5	6	**50**	tui_Latn	8	8	**49**
gaa_Latn	5	6	**43**	mxv_Latn	8	8	**24**	tuk_Latn	23	26	**53**
gil_Latn	9	8	**44**	mya_Mymr	45	52	**54**	tum_Latn	12	12	**49**
giz_Latn	9	10	**49**	myv_Cyrl	11	7	**47**	tur_Latn	55	**66**	56
gkn_Latn	8	7	**40**	mzh_Latn	7	9	**45**	twi_Latn	9	6	**46**
gkp_Latn	5	6	**35**	nan_Latn	6	6	**30**	tyv_Cyrl	19	18	**54**
gla_Latn	28	**43**	42	naq_Latn	8	7	**42**	tzh_Latn	12	13	**42**
gle_Latn	37	**53**	40	nav_Latn	7	9	**25**	tzo_Latn	13	11	**41**
glv_Latn	10	12	**38**	nbl_Latn	20	26	**46**	udm_Cyrl	10	11	**51**
gom_Latn	10	13	**39**	nch_Latn	10	8	**39**	ukr_Cyrl	61	**67**	56
gor_Latn	17	15	**50**	ncj_Latn	7	9	**43**	urd_Arab	59	**65**	59
guc_Latn	8	6	**42**	ndc_Latn	13	13	**40**	uzb_Latn	49	**59**	56
gug_Latn	11	7	**44**	nde_Latn	20	26	**46**	uzn_Cyrl	13	17	**57**
guj_Gujr	57	**67**	63	ndo_Latn	13	9	**40**	ven_Latn	10	8	**43**
gur_Latn	6	6	**47**	nds_Latn	16	15	**42**	vie_Latn	57	**65**	55
guw_Latn	11	9	**49**	nep_Deva	56	**61**	**61**	wal_Latn	15	9	**41**
gya_Latn	5	5	**39**	ngu_Latn	8	10	**50**	war_Latn	19	21	**41**
gym_Latn	10	7	**47**	nia_Latn	11	9	**47**	wbm_Latn	7	6	**52**
hat_Latn	11	10	**59**	nld_Latn	50	**59**	55	wol_Latn	11	9	**40**
hau_Latn	34	40	**47**	nmf_Latn	9	7	**36**	xav_Latn	10	10	**40**
haw_Latn	8	7	**41**	nnb_Latn	11	8	**46**	xho_Latn	23	32	**48**
heb_Hebr	16	31	**41**	nno_Latn	49	56	**57**	yan_Latn	7	7	**46**
hif_Latn	22	37	**42**	nob_Latn	54	**60**	55	yao_Latn	10	8	**43**
hil_Latn	26	31	**60**	nor_Latn	53	**63**	55	yap_Latn	8	8	**46**
hin_Deva	54	**70**	57	npi_Deva	53	**62**	61	yom_Latn	13	9	**35**
hmo_Latn	14	13	**53**	nse_Latn	17	10	**45**	yor_Latn	11	7	**51**
hne_Deva	32	40	**59**	nso_Latn	11	7	**48**	yua_Latn	12	10	**39**
hnj_Latn	8	7	**55**	nya_Latn	12	10	**56**	yue_Hani	52	**61**	54
hra_Latn	10	7	**49**	nyn_Latn	16	7	**38**	zai_Latn	16	14	**40**
hrv_Latn	56	**63**	56	nyy_Latn	8	8	**34**	zho_Hani	55	**68**	55
hui_Latn	9	7	**43**	nzi_Latn	5	7	**40**	zlm_Latn	59	**70**	64
hun_Latn	62	**69**	53	ori_Orya	54	**65**	60	zom_Latn	11	9	**50**
hus_Latn	7	10	**39**	ory_Orya	55	**64**	61	zsm_Latn	61	**64**	63
hye_Armn	60	**68**	60	oss_Cyrl	6	6	**47**	zul_Latn	24	35	**52**

Table 20: F1 of XLM-R-B, XLM-R-L, and Glot500-m on Text Classification (Part II).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
ace_Latn	2.50	2.83	**4.56**	hye_Armn	2.32	3.25	**4.91**	pam_Latn	2.85	3.52	**4.46**
ach_Latn	3.13	4.02	**5.60**	hye_Latn	2.34	**2.98**	2.44	pan_Guru	2.11	2.73	**4.11**
acr_Latn	2.01	2.46	**2.51**	iba_Latn	2.77	3.85	**6.01**	pap_Latn	3.12	3.85	**5.46**
afr_Latn	3.17	3.66	**5.46**	ibo_Latn	2.05	2.43	**4.33**	pau_Latn	2.67	3.09	**4.09**
agw_Latn	2.51	2.80	**4.09**	ifa_Latn	1.81	2.40	**3.45**	pcm_Latn	3.81	4.44	**6.47**
ahk_Latn	1.11	**1.23**	1.22	ifb_Latn	2.22	2.58	**3.28**	pdt_Latn	2.41	3.33	**5.11**
aka_Latn	3.38	4.50	**6.48**	ikk_Latn	1.75	2.29	**3.83**	pes_Arab	2.66	3.91	**4.81**
aln_Latn	4.06	4.92	**7.39**	ilo_Latn	3.06	3.87	**6.24**	pis_Latn	1.91	2.32	**4.42**
als_Latn	3.92	4.85	**6.32**	ind_Latn	4.06	5.00	**7.60**	pls_Latn	2.14	2.57	**4.02**
alt_Cyrl	2.91	3.36	**5.32**	isl_Latn	4.40	5.22	**7.07**	plt_Latn	3.74	3.99	**6.82**
alz_Latn	3.78	4.89	**5.94**	ita_Latn	3.55	4.02	**6.18**	poh_Latn	0.92	1.10	**1.87**
amh_Ethi	3.04	3.10	**4.87**	ium_Latn	2.00	2.27	**3.46**	pol_Latn	3.94	**5.20**	5.12
amh_Latn	1.41	**1.76**	1.70	ixl_Latn	1.62	1.94	**2.14**	pon_Latn	3.53	4.51	**5.18**
aoj_Latn	1.77	1.97	**3.22**	izz_Latn	1.65	2.06	**3.12**	por_Latn	3.61	4.35	**6.12**
arb_Arab	1.07	1.47	**2.40**	jam_Latn	2.77	3.06	**3.59**	prk_Latn	2.10	2.70	**5.40**
arn_Latn	2.40	2.79	**4.51**	jav_Latn	3.10	3.67	**5.21**	prs_Arab	3.54	4.28	**6.92**
ary_Arab	0.86	1.10	**2.43**	jpn_Jpan	3.62	**4.39**	4.07	pxm_Latn	1.76	2.15	**3.40**
arz_Arab	0.83	1.14	**2.52**	kaa_Cyrl	2.99	3.91	**5.45**	qub_Latn	2.48	2.97	**4.24**
asm_Beng	2.82	2.47	**5.21**	kaa_Latn	2.34	2.96	**3.64**	quc_Latn	1.87	2.45	**2.77**
ayr_Latn	2.61	3.09	**3.93**	kab_Latn	2.51	3.08	**3.14**	qug_Latn	2.44	2.99	**5.34**
azb_Arab	2.57	3.16	**4.96**	kac_Latn	1.66	2.17	**3.34**	quh_Latn	2.91	3.46	**5.43**
aze_Cyrl	2.76	3.26	**3.62**	kal_Latn	3.00	3.90	**4.73**	quw_Latn	2.89	3.50	**5.62**
aze_Latn	4.24	5.04	**8.00**	kan_Knda	2.58	3.18	**4.05**	quy_Latn	2.69	3.15	**5.51**
bak_Cyrl	2.20	2.38	**4.35**	kan_Latn	1.62	**2.08**	1.81	quz_Latn	3.33	3.89	**6.07**
bam_Latn	3.56	4.29	**5.73**	kat_Geor	4.06	4.99	**5.53**	qvi_Latn	2.82	3.42	**4.89**
ban_Latn	2.26	2.74	**3.37**	kaz_Cyrl	3.82	4.56	**5.31**	rap_Latn	1.31	1.61	**2.31**
bar_Latn	3.11	3.81	**3.84**	kbp_Latn	1.47	1.65	**3.32**	rar_Latn	1.83	2.22	**3.27**
bba_Latn	2.43	2.80	**4.16**	kek_Latn	1.91	2.45	**2.70**	rmy_Latn	2.85	3.68	**4.83**
bbc_Latn	3.02	3.85	**5.22**	khm_Khmr	1.57	1.70	**2.82**	ron_Latn	3.33	4.00	**4.99**
bci_Latn	2.81	3.18	**3.30**	kia_Latn	2.92	3.27	**4.69**	rop_Latn	1.60	2.08	**3.46**
bcl_Latn	3.78	4.61	**8.06**	kik_Latn	2.28	2.73	**4.38**	rug_Latn	2.56	2.95	**3.60**
bel_Cyrl	3.73	4.91	**6.46**	kin_Latn	2.67	3.26	**4.19**	run_Latn	3.33	3.98	**6.82**
bem_Latn	3.06	3.77	**5.69**	kir_Cyrl	4.54	4.35	**6.36**	rus_Cyrl	4.20	5.05	**7.38**
ben_Beng	3.29	3.07	**4.99**	kjb_Latn	2.42	3.03	**3.27**	sag_Latn	2.92	3.52	**5.17**
bhw_Latn	2.91	3.47	**5.16**	kjh_Cyrl	3.13	3.81	**5.39**	sah_Cyrl	2.31	3.01	**4.98**
bim_Latn	2.54	3.29	**4.12**	kmm_Latn	2.52	3.30	**3.73**	san_Deva	2.48	2.20	**3.64**
bis_Latn	2.59	2.96	**4.68**	kmr_Cyrl	2.31	2.76	**4.30**	san_Latn	1.54	2.23	**2.35**
bod_Tibt	0.54	**3.39**	2.43	kmr_Latn	3.75	4.19	**5.70**	sba_Latn	1.88	2.24	**3.86**
bqc_Latn	2.44	3.16	**4.61**	knv_Latn	1.27	1.53	**2.09**	seh_Latn	3.44	4.20	**4.94**
bre_Latn	3.32	**3.87**	3.79	kor_Hang	2.76	3.99	**4.89**	sin_Sinh	2.55	**3.60**	3.44
bts_Latn	4.06	4.92	**7.99**	kor_Latn	0.92	**2.40**	0.90	slk_Latn	4.65	5.06	**6.43**
btx_Latn	3.23	3.88	**5.59**	kpg_Latn	2.80	3.12	**5.77**	slv_Latn	3.11	4.32	**5.23**
bul_Cyrl	3.56	4.67	**5.88**	krc_Cyrl	2.85	3.66	**4.90**	sme_Latn	2.70	3.35	**4.40**
bum_Latn	3.22	3.73	**4.89**	kri_Latn	1.90	2.52	**5.07**	smo_Latn	2.26	2.72	**4.34**
bzj_Latn	1.65	2.43	**4.48**	ksd_Latn	2.82	3.28	**5.42**	sna_Latn	2.89	3.39	**5.32**
cab_Latn	2.16	2.63	**2.98**	kss_Latn	0.99	1.09	**1.49**	snd_Arab	3.12	3.92	**5.30**
cac_Latn	1.51	1.74	**2.86**	ksw_Mymr	0.95	1.46	**4.18**	som_Latn	3.15	3.40	**4.17**
cak_Latn	1.86	2.18	**3.24**	kua_Latn	4.25	4.92	**7.31**	sop_Latn	2.80	3.55	**4.23**
caq_Latn	2.20	2.94	**3.66**	lam_Latn	2.41	3.09	**4.03**	sot_Latn	3.49	4.31	**6.96**
cat_Latn	3.76	4.04	**5.24**	lao_Laoo	2.61	3.21	**4.39**	spa_Latn	3.71	4.21	**5.86**
cbk_Latn	3.12	3.64	**4.34**	lat_Latn	4.65	5.51	**7.44**	sqi_Latn	4.07	5.07	**6.50**
cce_Latn	2.96	3.40	**4.86**	lav_Latn	3.35	4.56	**6.45**	srm_Latn	1.75	1.96	**3.23**
ceb_Latn	3.45	4.13	**5.10**	ldi_Latn	3.41	3.94	**4.29**	srn_Latn	3.40	3.86	**5.98**
ces_Latn	4.33	5.27	**7.75**	leh_Latn	2.73	3.66	**5.28**	srp_Cyrl	6.48	6.50	**10.24**
cfm_Latn	2.69	3.18	**4.52**	lhu_Latn	1.43	**1.61**	1.36	srp_Latn	4.16	5.06	**6.31**
che_Cyrl	2.50	3.02	**3.17**	lin_Latn	1.78	2.73	**4.61**	ssw_Latn	3.27	4.02	**5.72**
chk_Hani	4.88	6.75	**7.08**	lit_Latn	4.69	5.66	**7.07**	sun_Latn	2.98	3.69	**4.61**
chk_Latn	3.20	3.94	**5.36**	loz_Latn	3.35	3.91	**6.03**	suz_Deva	1.68	1.66	**2.82**
chv_Cyrl	2.25	2.77	**4.79**	ltz_Latn	3.73	3.99	**5.16**	swe_Latn	4.77	4.76	**7.09**
ckb_Arab	2.38	3.15	**3.86**	lug_Latn	2.84	3.50	**5.59**	swh_Latn	4.05	4.99	**7.27**
ckb_Latn	2.11	2.57	**3.35**	luo_Latn	3.34	4.09	**4.90**	sxn_Latn	2.08	2.54	**3.06**
cmn_Hani	3.24	4.57	**5.22**	lus_Latn	2.43	2.99	**5.20**	tam_Latn	2.59	**3.08**	2.56
cnh_Latn	2.17	2.75	**3.62**	lzh_Hani	3.21	**5.56**	5.47	tam_Taml	3.09	3.77	**5.74**
crh_Cyrl	3.14	3.79	**6.77**	mad_Latn	2.65	3.29	**4.45**	tat_Cyrl	2.13	2.62	**4.03**
crs_Latn	2.63	3.46	**4.88**	mah_Latn	2.95	3.59	**4.92**	tbz_Latn	1.62	2.03	**4.22**
csy_Latn	2.58	3.02	**4.25**	mai_Deva	1.79	2.02	**3.86**	tca_Latn	1.29	1.56	**2.77**
ctd_Latn	2.94	3.61	**4.65**	mal_Latn	2.67	**3.36**	2.71	tdt_Latn	3.20	3.48	**5.06**
ctu_Latn	1.89	2.31	**2.40**	mal_Mlym	3.19	4.13	**4.76**	tel_Telu	2.87	3.78	**3.98**
cuk_Latn	2.20	2.87	**3.09**	mam_Latn	1.84	2.20	**2.22**	teo_Latn	3.37	4.18	**4.29**
cym_Latn	3.11	3.78	**3.85**	mar_Deva	3.87	5.13	**5.65**	tgk_Cyrl	2.63	3.29	**6.11**
dan_Latn	4.06	5.03	**6.94**	mau_Latn	1.60	**1.78**	1.12	tgl_Latn	3.22	3.35	**5.16**
deu_Latn	4.85	5.19	**7.28**	mbb_Latn	2.25	2.56	**3.51**	tha_Thai	1.50	2.72	**4.10**
djk_Latn	2.07	2.46	**3.53**	mck_Latn	3.34	4.06	**5.09**	tih_Latn	2.21	2.89	**4.57**
dln_Latn	3.89	4.89	**5.23**	mcn_Latn	3.74	4.42	**5.60**	tir_Ethi	1.90	1.93	**4.03**
dtp_Latn	2.05	2.28	**3.04**	mco_Latn	1.42	1.63	**1.69**	tlh_Latn	3.02	3.52	**5.71**
dyu_Latn	2.75	3.32	**5.29**	mdy_Ethi	1.36	1.26	**2.89**	tob_Latn	1.42	1.84	**2.00**
dzo_Tibt	0.39	**2.51**	2.03	meu_Latn	3.26	3.79	**5.10**	toh_Latn	2.17	2.90	**4.41**

Table 21: Accuracy of XLM-R-B, XLM-R-L, and Glot500-m on Round Trip Alignment (Part I).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
efi_Latn	2.55	3.25	**6.23**	mfe_Latn	3.61	4.19	**6.26**	toi_Latn	3.19	4.10	**4.31**
ell_Grek	2.79	3.38	**4.77**	mgh_Latn	2.78	3.28	**3.48**	toj_Latn	1.43	1.84	**2.25**
eng_Latn	4.02	4.49	**6.39**	mgr_Latn	3.32	4.06	**6.39**	ton_Latn	2.01	2.64	**3.63**
enm_Latn	3.77	4.60	**7.19**	mhr_Cyrl	2.75	3.28	**5.32**	top_Latn	1.56	2.16	**2.19**
epo_Latn	4.01	4.83	**5.88**	min_Latn	2.62	3.05	**3.78**	tpi_Latn	2.44	2.71	**5.96**
est_Latn	4.34	5.24	**8.21**	miq_Latn	2.23	3.13	**4.12**	tpm_Latn	2.79	3.39	**4.67**
eus_Latn	3.12	3.80	**4.19**	mkd_Cyrl	3.99	4.54	**7.37**	tsn_Latn	2.82	3.12	**4.63**
ewe_Latn	2.22	2.67	**4.74**	mlg_Latn	3.34	3.81	**6.33**	tso_Latn	2.40	3.05	**5.00**
fao_Latn	3.85	4.62	**5.75**	mlt_Latn	2.94	3.57	**4.87**	tsz_Latn	2.68	3.14	**4.20**
fas_Arab	4.54	4.48	**7.00**	mos_Latn	2.71	3.24	**4.25**	tuc_Latn	1.43	1.83	**2.36**
fij_Latn	2.81	3.17	**4.94**	mps_Latn	1.50	1.65	**3.05**	tui_Latn	2.47	2.83	**4.53**
fil_Latn	3.26	3.92	**4.80**	mri_Latn	2.81	3.44	**5.49**	tuk_Cyrl	2.74	3.68	**4.33**
fin_Latn	4.06	5.19	**6.03**	mrw_Latn	2.69	3.24	**4.58**	tuk_Latn	2.43	3.23	**4.74**
fon_Latn	1.63	1.89	**3.70**	msa_Latn	3.17	3.50	**5.38**	tum_Latn	3.41	4.13	**6.15**
fra_Latn	3.19	3.97	**5.08**	mwm_Latn	1.74	1.99	**3.20**	tur_Latn	5.18	4.86	**7.45**
fry_Latn	3.36	3.99	**4.52**	mxv_Latn	1.75	2.11	**2.31**	twi_Latn	3.05	4.06	**6.70**
gaa_Latn	2.74	3.26	**6.01**	mya_Mymr	1.54	1.53	**2.46**	tyv_Cyrl	2.31	2.83	**3.33**
gil_Latn	2.76	3.20	**4.50**	myv_Cyrl	2.90	3.42	**4.46**	tzh_Latn	2.16	2.50	**3.08**
giz_Latn	3.00	3.43	**5.40**	mzh_Latn	2.62	3.02	**4.10**	tzo_Latn	2.01	2.29	**2.77**
gkn_Latn	1.93	2.07	**3.31**	nan_Latn	1.99	2.51	**2.56**	udm_Cyrl	2.90	3.48	**4.72**
gkp_Latn	1.88	2.25	**3.40**	naq_Latn	2.42	3.15	**4.41**	uig_Arab	2.58	3.11	**3.61**
gla_Latn	2.90	3.48	**3.61**	nav_Latn	1.75	2.10	**2.71**	uig_Latn	2.26	2.76	**3.79**
gle_Latn	3.52	4.24	**4.49**	nbl_Latn	3.09	3.87	**4.85**	ukr_Cyrl	5.71	5.96	**7.47**
glv_Latn	2.76	3.38	**4.45**	nch_Latn	2.18	2.74	**3.32**	urd_Arab	1.88	2.88	**3.96**
gom_Latn	3.05	3.59	**4.40**	ncj_Latn	2.64	3.40	**3.69**	urd_Latn	2.29	2.97	**3.03**
gor_Latn	2.26	2.73	**3.71**	ndc_Latn	3.32	3.85	**6.67**	uzb_Cyrl	2.73	3.26	**7.24**
grc_Grek	1.11	2.00	**2.93**	nde_Latn	4.00	4.60	**6.05**	uzb_Latn	3.32	3.98	**5.91**
guc_Latn	1.46	1.80	**2.23**	ndo_Latn	3.21	3.85	**5.61**	uzn_Cyrl	2.61	3.06	**5.86**
gug_Latn	2.60	3.23	**4.70**	nds_Latn	2.98	3.69	**4.70**	ven_Latn	2.96	3.64	**5.34**
guj_Gujr	3.18	4.15	**4.38**	nep_Deva	3.02	2.97	**6.31**	vie_Latn	3.99	4.48	**6.69**
gur_Latn	2.14	2.59	**3.22**	ngu_Latn	1.86	2.34	**3.39**	wal_Latn	2.87	3.65	**4.24**
guw_Latn	2.18	2.54	**4.56**	nia_Latn	2.75	**3.47**	3.24	war_Latn	3.04	3.74	**5.43**
gya_Latn	1.94	2.25	**4.63**	nld_Latn	2.81	3.63	**4.90**	wbm_Latn	2.44	2.86	**6.53**
gym_Latn	1.44	1.78	**2.63**	nmf_Latn	3.30	4.27	**5.05**	wol_Latn	3.47	4.48	**6.10**
hat_Latn	3.21	3.64	**6.39**	nnb_Latn	2.46	3.14	**4.08**	xav_Latn	0.87	1.03	**1.12**
hau_Latn	3.69	4.24	**6.31**	nno_Latn	3.90	4.61	**7.41**	xho_Latn	3.61	4.27	**5.90**
haw_Latn	2.25	2.63	**3.55**	nob_Latn	3.88	4.81	**5.83**	yan_Latn	2.95	3.35	**5.59**
heb_Hebr	1.85	2.41	**3.92**	nor_Latn	3.31	4.14	**5.82**	yao_Latn	2.01	2.66	**3.87**
hif_Latn	2.90	3.43	**3.60**	npi_Deva	3.29	3.30	**5.93**	yap_Latn	2.86	3.41	**3.45**
hil_Latn	2.92	3.48	**4.88**	nse_Latn	3.29	4.06	**5.74**	yom_Latn	3.25	4.00	**5.17**
hin_Deva	3.39	3.80	**5.13**	nso_Latn	3.06	3.92	**5.51**	yor_Latn	2.24	2.68	**3.88**
hin_Latn	2.94	3.20	**4.77**	nya_Latn	2.76	**3.19**	**5.96**	yua_Latn	2.04	2.26	**2.86**
hmo_Latn	2.43	2.70	**6.12**	nyn_Latn	2.77	3.50	**5.59**	yue_Hani	2.37	**3.19**	2.95
hne_Deva	2.48	2.53	**4.95**	nyy_Latn	2.21	2.74	**2.95**	zai_Latn	3.22	3.76	**5.21**
hnj_Latn	2.14	2.53	**4.28**	nzi_Latn	2.09	2.70	**4.20**	zho_Hani	2.77	4.38	**5.03**
hra_Latn	3.32	3.86	**5.19**	ori_Orya	2.73	2.77	**3.92**	zlm_Latn	4.39	5.15	**7.54**
hrv_Latn	4.14	5.24	**7.02**	ory_Orya	3.27	3.20	**4.39**	zom_Latn	3.65	4.45	**5.36**
hui_Latn	1.84	2.10	**3.47**	oss_Cyrl	2.20	2.52	**5.85**	zsm_Latn	4.49	5.07	**8.83**
hun_Latn	4.54	4.10	**5.62**	ote_Latn	1.89	2.23	**2.66**	zul_Latn	3.67	4.39	**5.44**
hus_Latn	1.70	2.00	**2.42**	pag_Latn	2.93	3.44	**4.56**				

Table 22: Accuracy of XLM-R-B, XLM-R-L, and Glot500-m on Round Trip Alignment (Part II).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
srd_Latn	87.2	66.6	**5.4**	aka_Latn	86.7	74.1	**14.2**	dyu_Latn	68.5	27.4	**10.2**
ben_Beng	5.2	**3.7**	7.2	mon_Latn	288	282.4	**33.7**	nyy_Latn	628.5	198.3	**18.0**
ajp_Arab	74.6	**34.0**	44.8	gor_Latn	89.8	140.7	**8.8**	tzh_Latn	320.3	82.8	**4.7**
tdx_Latn	688.4	716.4	**16.0**	kjb_Latn	110.8	81.1	**16.2**	hne_Deva	80.1	60.3	**9.1**
tpm_Latn	99.9	90.2	**17.9**	lhu_Latn	44.7	12.3	**2.0**	bel_Cyrl	3.4	**2.5**	5.3
grc_Grek	10.1	10.4	**3.4**	bos_Latn	6.1	**3.4**	7.9	szl_Latn	46.4	30.2	**3.1**
sxn_Latn	469.2	148.3	**14.5**	lmo_Latn	48.4	25.9	**6.1**	ksh_Latn	340.3	227.6	**19.9**
cos_Latn	52.1	22.8	**13.3**	mwn_Latn	697.8	543.8	**30.7**	pcd_Latn	61.2	40.8	**13.2**
tlh_Latn	53.6	46.3	**11.1**	aym_Latn	1084.6	727.8	**14.5**	ada_Latn	100	78.5	**9.5**
sid_Latn	1003.6	782.3	**34.5**	aoj_Latn	95.1	53.7	**7.4**	pxm_Latn	101.3	120.7	**2.7**
jam_Latn	213.3	195.2	**15.8**	est_Latn	7.7	**4.0**	22.1	xho_Latn	32.5	**9.4**	16.7
ban_Latn	40.8	76.1	**16.1**	bre_Latn	12.9	**3.7**	12.3	kaa_Cyrl	72.9	29.2	**8.8**
kin_Latn	544.1	203.2	**6.6**	bsb_Latn	74.5	45.1	**7.6**	kea_Latn	754.2	525.3	**13.4**
rop_Latn	150.7	93.4	**8.4**	yua_Latn	246.8	55.1	**4.6**	teo_Latn	587.1	271.7	**62.0**
alz_Latn	511.9	145.6	**47.7**	hrv_Latn	7.4	**4.9**	9.7	tsc_Latn	726.3	501.1	**17.0**
kwy_Latn	598.8	514.4	**30.5**	jav_Latn	20.2	**4.4**	22	hin_Deva	7.4	**3.1**	10
yor_Latn	109.1	55.9	**11.0**	mai_Deva	42.9	48.8	**6.0**	ekk_Latn	7	**3.8**	11.8
lao_Laoo	4.2	4.4	**3.8**	tyv_Cyrl	104.1	104.4	**7.3**	umb_Latn	920	838.8	**17.4**
aze_Latn	5.6	**3.6**	5.4	afb_Arab	68.7	**44.4**	55.9	tam_Taml	7.2	**2.3**	9.8
mya_Mymr	6.9	**2.7**	6.3	twi_Latn	178.9	66.7	**17.9**	toi_Latn	988.7	246.5	**20.9**
ssw_Latn	345.7	108.4	**20.2**	sme_Latn	293	368.2	**6.5**	kon_Latn	463.7	418.9	**16.3**
lus_Latn	493.5	131.2	**16.4**	yom_Latn	468	240.7	**43.1**	che_Cyrl	266.4	127.6	**5.7**
krc_Cyrl	120.1	63.2	**9.3**	tob_Latn	115	78.8	**7.2**	gaa_Latn	109.3	33.3	**13.5**
hbo_Hebr	6.3	**3.6**	5.6	mxv_Latn	69.8	29.7	**5.0**	tzo_Latn	246.5	54.3	**7.0**
mgr_Latn	737.8	254.2	**33.0**	ron_Latn	4.4	**2.9**	10.4	mon_Cyrl	5.8	**3.4**	8.6
crh_Cyrl	138.6	86.3	**5.2**	ile_Latn	67.9	40.1	**5.7**	cuk_Latn	211.5	72.1	**32.0**
ara_Arab	10.1	**6.3**	18.8	cce_Latn	468.3	123.5	**22.5**	ces_Latn	4.4	**3.1**	11.6
mar_Deva	7.5	**4.6**	11.2	uzn_Cyrl	402.4	138.7	**5.2**	rmy_Latn	288.2	349.8	**25.0**
nba_Latn	638.8	675.1	**14.6**	ibg_Latn	897.3	807.3	**21.8**	phm_Latn	914.5	678.5	**11.6**
mny_Latn	568.9	492.5	**38.7**	hat_Latn	228	113.3	**14.0**	glv_Latn	240.2	182.3	**9.4**
run_Latn	817.5	218.5	**16.9**	fij_Latn	377.3	96	**12.8**	diq_Latn	256.6	120.5	**13.4**
rus_Cyrl	3.3	**2.3**	4.5	kbp_Latn	34.6	24.5	**7.1**	poh_Latn	62.8	68.9	**3.8**
hbs_Latn	4.5	**2.6**	6	mlt_Latn	223	162.2	**10.3**	oss_Cyrl	121.8	58.7	**5.1**
lug_Latn	489	197.5	**13.1**	kjh_Cyrl	209.8	88.8	**16.4**	san_Deva	20.5	**12.4**	15.5
pls_Latn	91.7	98.9	**6.9**	ndo_Latn	892.3	178.1	**21.1**	ote_Latn	127.8	71.2	**8.0**
hif_Latn	21.6	46.7	**13.5**	rar_Latn	458.1	50.2	**12.1**	her_Latn	776	707.3	**31.6**
tll_Latn	244.6	161	**24.3**	ell_Grek	3.4	**2.6**	5.9	efi_Latn	256.8	47	**11.5**
crs_Latn	782.2	146.5	**7.4**	tvl_Latn	634.1	378.5	**7.1**	idu_Latn	117.7	90.9	**12.0**
rng_Latn	656.6	606.8	**11.7**	toj_Latn	287.1	113.6	**9.6**	hye_Armn	**3.6**	4.4	3.8
cjk_Latn	530.8	419.6	**24.0**	ikk_Latn	67.8	49.5	**8.6**	gcf_Latn	450.8	292.4	**5.5**
seh_Latn	917.8	230	**11.2**	ory_Orya	6.1	**2.8**	6.3	pus_Arab	12.9	**7.5**	12.7
rug_Latn	260.9	214.2	**5.4**	nor_Latn	5	**2.8**	8.5	sgs_Latn	119.2	124.7	**10.5**
hau_Latn	14.5	**7.1**	17.2	enm_Latn	43.1	**31.0**	36.6	mbb_Latn	177.1	138	**4.2**
uzb_Latn	5.6	**3.6**	5.8	arz_Arab	17.5	**1.5**	6.8	som_Arab	7.2	**3.1**	9.3
bim_Latn	142.2	97.3	**11.3**	bem_Latn	706.9	219.9	**27.1**	hsb_Latn	109.6	103.6	**5.2**
vep_Latn	218.1	111.5	**6.1**	gkp_Latn	33.1	30.2	**12.7**	ary_Arab	32.7	**4.6**	26
slv_Latn	7.8	**4.9**	26.9	guj_Gujr	6.2	**3.6**	6.5	hmo_Latn	509.3	77.7	**10.9**
azj_Latn	5.3	**3.3**	5.1	tbz_Latn	39.2	40.4	**8.4**	quw_Latn	177.8	157.7	**26.1**
cac_Latn	51.4	39.3	**7.0**	ven_Latn	268.3	62	**9.4**	pag_Latn	923.5	232.4	**25.8**
npi_Deva	8.6	**4.9**	7.3	crh_Latn	151	70.9	**6.5**	ber_Latn	639.1	981.4	**21.3**
lin_Latn	377.3	96.6	**15.3**	xmv_Latn	593.2	491.4	**19.4**	chk_Latn	766.9	151.6	**19.1**
zom_Latn	238.7	176.2	**22.8**	slk_Latn	4	**2.9**	11.2	kan_Knda	7.2	**2.8**	8.9
kmr_Cyrl	140.6	56.7	**4.1**	zne_Latn	854.7	658.4	**48.8**	loz_Latn	895	113.7	**27.8**
acm_Arab	113.6	**74.0**	81	cgg_Latn	565.7	454.4	**12.4**	tih_Latn	247.6	151.3	**4.9**
fin_Latn	4.2	**3.1**	21.7	vie_Latn	7.6	**3.1**	16.4	mfe_Latn	767.9	255.4	**10.1**
rmn_Grek	108.9	76.8	**3.3**	amh_Ethi	8.9	**5.3**	7.5	tel_Telu	6.5	**4.0**	7.9
wls_Latn	334.9	207.9	**4.0**	nyu_Latn	926.2	479.2	**9.3**	ina_Latn	26.9	17.1	**7.2**
hun_Latn	5.1	**3.3**	25.1	suz_Deva	63.4	76.4	**2.5**	isl_Latn	7.9	**4.9**	16.7
lij_Latn	98.8	55.1	**5.9**	tuc_Latn	108.9	80.8	**7.6**	tsz_Latn	990.6	199.7	**14.2**
quh_Latn	279	176.6	**16.5**	lub_Latn	670.8	577.5	**23.8**	ori_Orya	5.2	**3.0**	4.7
yap_Latn	507.3	195.9	**10.6**	epo_Latn	10.8	**5.2**	21	tat_Latn	168.4	65.5	**6.9**
abk_Cyrl	122.6	89.5	**20.1**	ksw_Mymr	16.6	7.5	**4.6**	arg_Latn	29.2	13.6	**7.2**
cmn_Hani	10.4	**5.0**	9.8	mwl_Latn	69.1	35.6	**4.9**	kia_Latn	132.4	126.8	**18.5**
csb_Latn	112.8	59.4	**6.1**	cak_Latn	101.7	46.1	**5.4**	afr_Latn	12.2	**7.8**	19.2
nbl_Latn	137.7	19.6	**13.9**	bar_Latn	124.7	108.9	**14.4**	myv_Cyrl	97.7	153.3	**8.5**
ndc_Latn	1188.5	374.6	**19.4**	asm_Beng	6	**3.8**	5	bik_Latn	170.4	60.3	**13.7**
oci_Latn	41.2	24.4	**8.3**	grn_Latn	199.3	141.6	**10.3**	ltz_Latn	39.7	165.1	**10.9**
fao_Latn	84.2	35.6	**5.5**	tso_Latn	506.1	115.2	**13.2**	iso_Latn	236.2	222.4	**8.7**
tui_Latn	126.1	127	**20.6**	nso_Latn	656.3	153.4	**9.1**	ewe_Latn	198	54.6	**20.0**
xav_Latn	21.4	15.9	**5.7**	bum_Latn	282.8	91.5	**22.1**	als_Latn	7.6	**2.5**	6.4

Table 23: Perplexity of all languages covered by Glot500-m (Part I).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
swc_Latn	39.2	22.5	**13.2**	top_Latn	589.2	89.6	**23.5**	hin_Latn	**11.1**	22.1	11.9
deu_Latn	4.4	**3.6**	10.2	bin_Latn	278.1	169.8	**13.3**	eng_Latn	5.7	**4.0**	7.5
caq_Latn	185.9	129	**21.6**	chw_Latn	778.9	645.8	**33.9**	hus_Latn	134.6	68.2	**5.3**
ceb_Latn	63.1	53.1	**2.1**	hyw_Cyrl	268.5	233.5	**6.3**	urh_Latn	236.8	211.5	**11.4**
nia_Latn	280.3	85.5	**7.5**	kor_Hang	7.2	**2.6**	11	mkd_Cyrl	4.3	**3.1**	6.2
urd_Arab	8.3	**5.3**	8.7	btx_Latn	463	163.1	**19.3**	wbm_Latn	58.9	47.3	**13.6**
niu_Latn	600.1	437.5	**10.1**	srn_Latn	609.3	137.2	**12.6**	kwn_Latn	1053.6	753.2	**32.0**
mrw_Latn	320.8	174.9	**7.6**	llb_Latn	555.6	589.8	**41.1**	guc_Latn	432.6	117.8	**9.4**
bul_Cyrl	3.9	**3.6**	6.8	cbk_Latn	129.5	60.4	**11.6**	quc_Latn	270.7	83.9	**5.6**
pau_Latn	333.7	147.3	**7.2**	bcl_Latn	270	60.1	**12.5**	nds_Latn	112.5	161.1	**7.4**
tha_Thai	10.8	**2.9**	14.6	csy_Latn	198.3	152.5	**21.7**	ind_Latn	8.5	**5.4**	17.1
ilo_Latn	786.7	184.4	**13.8**	ctd_Latn	249.2	166.1	**11.6**	nde_Latn	56.7	21.5	**12.1**
kss_Latn	90.4	13.2	**11.2**	plt_Latn	10.8	**3.6**	5.7	kua_Latn	1104.8	191.2	**13.4**
zai_Latn	719.4	212.5	**10.4**	smo_Latn	235.7	55.6	**7.0**	nch_Latn	705.1	166.4	**11.2**
guw_Latn	267.7	65.5	**6.9**	kab_Latn	744.5	203.5	**24.3**	por_Latn	5.1	**3.9**	9.3
kbd_Cyrl	175.7	94.4	**9.1**	gom_Deva	82.8	48.4	**9.0**	jpn_Jpan	7.9	**3.9**	10
dln_Latn	238.8	207.8	**7.5**	ukr_Cyrl	3.1	**2.9**	5.9	spa_Latn	4.6	**3.5**	7.8
war_Latn	200.9	110.7	**2.3**	ast_Latn	27.5	18.6	**4.8**	knv_Latn	129	78.3	**5.8**
tca_Latn	70.4	49	**6.0**	lvs_Latn	4.8	**2.7**	5.7	agw_Latn	150.1	73.4	**16.3**
iku_Cans	2.2	**1.9**	5.8	rmn_Cyrl	624.3	513.1	**8.7**	ige_Latn	181.1	105.2	**11.9**
bjn_Latn	41.3	17.6	**11.4**	kir_Cyrl	7.7	**2.9**	11.9	dua_Latn	232.8	152.2	**19.1**
ngu_Latn	918	110.9	**13.4**	pfl_Latn	152	101.3	**11.3**	ogo_Latn	131.3	129.7	**31.1**
kmr_Latn	68	**4.6**	10.6	bqc_Latn	102.7	71.1	**26.5**	bas_Latn	410.4	437.7	**16.7**
tgl_Latn	7.9	**4.4**	8.9	yid_Hebr	7.6	**4.8**	5.1	bpy_Beng	20	21.4	**2.9**
eus_Latn	10.7	**6.2**	37.3	fil_Latn	9.2	**2.3**	9.9	lfn_Latn	60.4	51	**6.9**
hra_Latn	212.1	177.7	**54.3**	nap_Latn	81.7	39.6	**10.5**	ton_Latn	116	65.2	**2.8**
lue_Latn	839.2	627.4	**19.8**	heb_Hebr	6.7	**4.9**	13.5	lim_Latn	66.8	43.5	**11.4**
pol_Latn	4.5	**2.7**	10.6	sba_Latn	75.7	81.8	**6.0**	lav_Latn	4.2	**2.2**	6.6
leh_Latn	476.5	253.9	**26.2**	ifa_Latn	371.9	266.1	**6.0**	bih_Deva	27.6	16.1	**5.0**
lat_Latn	15.3	**3.7**	24.5	ami_Latn	1070.7	710.2	**29.2**	gym_Latn	509.6	66.3	**17.0**
div_Thaa	1.6	**1.5**	3.5	gil_Latn	763.5	161.3	**15.7**	ish_Latn	144.9	134	**11.6**
min_Latn	105	39.7	**3.9**	djk_Latn	360.4	93.4	**13.4**	zea_Latn	69.6	27.5	**8.7**
ctu_Latn	177.4	37.9	**4.5**	new_Deva	36.1	29.8	**4.5**	aln_Latn	3.9	**2.3**	12.7
tur_Latn	9.1	**4.1**	29.5	bam_Latn	74.5	**23.7**	46.8	gcr_Latn	352.9	314.7	**7.5**
dhv_Latn	509	435.8	**11.8**	wol_Latn	236.4	158.3	**32.0**	kal_Latn	377.2	370.9	**8.3**
lua_Latn	706	784.5	**21.7**	alt_Cyrl	140.7	50.9	**9.3**	dan_Latn	6	**3.6**	13.1
rmy_Cyrl	488.1	389.3	**9.3**	kri_Latn	87.6	35.8	**8.6**	tah_Latn	363	330.9	**4.8**
zpa_Latn	476.1	550.1	**13.6**	kom_Cyrl	93.4	57	**4.9**	kik_Latn	205.8	55.5	**12.1**
gom_Latn	405.7	282.9	**27.9**	sah_Cyrl	99.9	91.1	**4.5**	vmw_Latn	828.8	434.8	**17.8**
dtp_Latn	166.4	78.7	**5.5**	mzh_Latn	132.8	133.4	**9.6**	eml_Latn	283.4	144.9	**6.6**
fra_Latn	4.1	**2.8**	6.9	sna_Latn	316.6	331.1	**16.4**	sco_Latn	28.1	15.5	**9.8**
cat_Latn	4.1	**2.2**	7.3	bzj_Latn	264.7	75.8	**10.9**	kac_Latn	189.9	76.3	**17.9**
xmf_Geor	71.2	72.3	**3.8**	nld_Latn	5.7	**4.5**	12	ttj_Latn	865.2	509.5	**15.5**
ixl_Latn	53	29.6	**4.2**	gug_Latn	626.9	141.6	**8.4**	lun_Latn	720.1	565.6	**31.9**
ckb_Arab	72.2	80.6	**6.0**	yue_Hani	17.8	**10.6**	10.8	sot_Latn	269.1	122.4	**8.1**
ahk_Latn	44.8	9.1	**2.1**	fry_Latn	16.1	**15.4**	17.2	mau_Latn	199.7	13.6	**8.4**
sag_Latn	491.4	68.7	**11.1**	jbo_Latn	132.3	187.1	**9.0**	yan_Latn	134.4	108.4	**31.4**
qug_Latn	505	135.2	**13.7**	iba_Latn	529.3	87	**16.6**	ido_Latn	79.8	24.2	**7.1**
nyn_Latn	834.8	236.9	**16.8**	nya_Latn	319.6	256.8	**12.7**	rmn_Latn	968.8	1062.8	**22.9**
koo_Latn	481.3	321.6	**13.8**	tat_Cyrl	99.8	116	**4.1**	sat_Olck	1.4	**1.2**	4.6
uig_Arab	8.1	**2.4**	5.5	nzi_Latn	113.7	47.4	**12.5**	mad_Latn	132.7	90.2	**7.9**
kam_Latn	225.9	155.7	**10.3**	wal_Latn	492.7	120.3	**18.1**	hil_Latn	366	38.7	**9.6**
gkn_Latn	248	74.6	**9.4**	pdt_Latn	417.7	143	**13.3**	khm_Khmr	4.8	**3.2**	4.5
twx_Latn	1209.8	978.2	**15.5**	apc_Arab	74.8	42.2	**37.2**	fon_Latn	71.8	27	**10.4**
skg_Latn	665.4	624.1	**15.8**	mdy_Ethi	65.7	68.4	**5.4**	ngl_Latn	664.9	518.3	**15.9**
arb_Arab	4.1	**2.1**	6	rue_Cyrl	18.7	11.4	**4.5**	tcf_Latn	224.5	225.4	**6.9**
mco_Latn	295	37.6	**4.6**	azb_Arab	194.1	141.8	**4.8**	gur_Latn	86.2	39	**17.9**
sqi_Latn	6.2	**2.1**	8.4	bci_Latn	129.6	95.6	**8.7**	qvi_Latn	863.4	91.5	**12.3**
cnh_Latn	496	154.4	**16.3**	kmm_Latn	193.3	164.9	**20.2**	izz_Latn	95.5	78.5	**5.5**
sin_Sinh	7.5	**5.4**	9.8	bak_Cyrl	99	79	**5.3**	kur_Arab	90.3	76.3	**5.7**
kmb_Latn	564.8	465.8	**15.6**	miq_Latn	347.4	198.9	**23.6**	hbs_Cyrl	3.7	**2.3**	4.3
vol_Latn	78.4	67.7	**2.4**	kaa_Latn	94.2	100.6	**7.3**	ach_Latn	488.8	114.6	**77.3**
msa_Latn	**8.2**	26.1	15	bod_Tibt	8.8	**4.0**	6.3	wuu_Hani	35.9	16.8	**11.7**
bba_Latn	75.5	65.5	**16.3**	glg_Latn	5.9	**4.6**	9.2	quz_Latn	804.5	269.4	**12.2**
tgk_Latn	11.9	11.7	**7.5**	tum_Latn	516.4	168.3	**10.2**	tok_Latn	592.4	423	**94.5**
tiv_Latn	912.3	716.3	**29.3**	bbc_Latn	787.9	203.7	**13.6**	bis_Latn	727.1	47.7	**10.7**
hmn_Latn	60.9	52.5	**8.8**	kek_Latn	126.4	40.6	**4.3**	fur_Latn	196.5	142.8	**7.7**
swh_Latn	12.6	**5.8**	24.4	ace_Latn	81.5	54	**6.4**	ium_Latn	36.6	33.1	**7.2**
pis_Latn	563.2	64.7	**9.7**	pam_Latn	59.6	276.7	**28.2**	nse_Latn	771.7	292.3	**13.7**
mzn_Arab	50	34.3	**6.3**	fas_Arab	8	**4.1**	14.1	zul_Latn	36.3	**10.1**	21.7

Table 24: Perplexity of all languages covered by Glot500-m (Part II).

Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m	Language-Script	XLM-R-B	XLM-R-L	Glot500-m
bts_Latn	205.7	204.5	**8.8**	tsn_Latn	264.7	137.8	**12.5**	orm_Latn	23.4	**8.6**	16
gla_Latn	11.5	12.7	**7.2**	pon_Latn	928.4	181.9	**19.2**	luo_Latn	699.4	258.5	**85.1**
kat_Latn	36.4	24.8	**18.3**	nmf_Latn	297.6	310.6	**44.9**	pcm_Latn	38.3	169.6	**3.6**
uig_Latn	188.8	173.9	**15.2**	ajg_Latn	147.1	149.5	**22.6**	nnb_Latn	364.1	95	**28.6**
kat_Geor	6	**3.9**	6.4	tir_Ethi	28.3	15.7	**4.4**	kaz_Cyrl	**4.3**	5.4	9.6
mlg_Latn	10.9	**4.4**	7.6	bhw_Latn	411.2	126.2	**21.6**	dzo_Tibt	8.5	**3.3**	5.7
arn_Latn	382.7	96.7	**17.6**	mhr_Cyrl	122.9	168.4	**5.8**	sun_Latn	23.6	**11.9**	17
tuk_Latn	456.7	197.8	**5.8**	swe_Latn	4.8	**3.5**	12.7	vec_Latn	40.6	21.1	**9.2**
vls_Latn	97.7	39.6	**9.7**	scn_Latn	117	64.9	**7.8**	ayr_Latn	261.1	237.6	**27.7**
hyw_Armn	15.8	9.1	**4.3**	udm_Cyrl	356.7	224.9	**6.7**	oke_Latn	209.2	220.1	**13.0**
que_Latn	447.9	536.1	**11.9**	ifb_Latn	246.3	177.9	**5.1**	kur_Latn	14.2	**6.8**	10.3
snd_Arab	13.2	**4.1**	19.5	naq_Latn	136.8	60.2	**15.7**	mgh_Latn	680	272.8	**23.7**
giz_Latn	81.9	82.9	**37.7**	zlm_Latn	5.6	**3.3**	4.6	tgk_Cyrl	181.3	153	**4.5**
ita_Latn	4.5	**3.3**	7.2	hrx_Latn	478.1	679.1	**14.9**	sop_Latn	607.5	228.2	**29.5**
qub_Latn	283.2	312.7	**9.4**	lzh_Hani	70	58	**21.8**	mos_Latn	272.6	118.3	**13.2**
nav_Latn	228.5	126.5	**5.2**	pap_Latn	674.4	149.3	**18.1**	rap_Latn	36.1	31.1	**2.8**
kqn_Latn	825.9	686.6	**17.5**	cfm_Latn	235.1	155	**14.0**	prk_Latn	69.4	45.9	**7.1**
toh_Latn	758.3	216.6	**19.6**	chv_Cyrl	122.5	73.8	**5.4**	uzb_Cyrl	236.2	138.4	**4.9**
mah_Latn	314.7	81.8	**17.3**	tdt_Latn	641.9	78.6	**9.7**	tog_Latn	821.1	777.7	**13.4**
wes_Latn	144.6	103.9	**14.3**	pan_Guru	4.4	**2.5**	4.3	mal_Mlym	5	**3.7**	6.2
nob_Latn	6.8	**4.0**	9.5	pms_Latn	83.6	46.2	**3.6**	nyk_Latn	1182.6	914.2	**16.5**
ext_Latn	68.3	38.2	**8.1**	roh_Latn	243.5	170	**7.0**	quy_Latn	949.7	320.2	**14.5**
lam_Latn	233.7	160.8	**21.6**	prs_Arab	6.8	**3.5**	4.8	abn_Latn	245.2	272.5	**8.7**
mwm_Latn	44.8	53.1	**7.1**	tuk_Cyrl	277.4	86.3	**6.7**	mcn_Latn	120.7	129.7	**43.6**
kpg_Latn	165.9	122.6	**15.1**	srm_Latn	257.5	74.5	**12.3**	nep_Deva	8.8	**6.3**	10
hau_Arab	5.3	**3.0**	8.1	gsw_Latn	288.2	181.2	**22.3**	gle_Latn	10.5	**3.7**	9.8
ksd_Latn	150	154.9	**7.7**	fat_Latn	192.3	149	**17.6**	cab_Latn	1216.7	155.6	**15.4**
zsm_Latn	12.2	**2.9**	22.7	ldi_Latn	394.8	107.1	**38.2**	mps_Latn	75.2	55.2	**17.4**
hui_Latn	209.9	177	**10.0**	kos_Latn	470.7	485.7	**27.0**	pnb_Arab	51.8	30.8	**7.1**
cym_Latn	8.2	**4.8**	11.2	acr_Latn	155.7	90.7	**5.8**	swa_Latn	11.4	**6.4**	20
srp_Latn	10.9	**7.9**	13.3	mri_Latn	63	59.5	**8.7**	hnj_Latn	88.3	92.5	**11.3**
bak_Latn	347.1	211	**7.5**	frr_Latn	117.6	101	**9.5**	haw_Latn	63.5	66.7	**7.4**
zho_Hani	20.7	**5.9**	31.3	mck_Latn	369.3	164.8	**24.7**	tpi_Latn	891.8	67.8	**8.8**
nno_Latn	**9.9**	12.7	10.4	pes_Arab	5.5	**3.1**	5.3	ncj_Latn	1019	136.2	**13.7**
gya_Latn	31	24.3	**16.5**	san_Latn	94.4	96.8	**12.0**	som_Latn	14.1	**6.9**	22.2
ibo_Latn	77.1	90.1	**8.5**	yao_Latn	738.9	162.4	**13.8**	mam_Latn	132.7	62.4	**6.1**
meu_Latn	380.2	158.5	**26.7**	srp_Cyrl	7.4	**4.5**	8.4	lit_Latn	4.4	**2.5**	10.6
ncx_Latn	1084.7	948.5	**14.6**	ful_Latn	104	105.6	**13.1**				

Table 25: Perplexity of all languages covered by Glot500-m (Part III).

Chapter 3

MaLA-500: Massive Language Adaptation of Large Language Models

MaLA-500: Massive Language Adaptation of Large Language Models

Peiqin Lin[*1,2], **Shaoxiong Ji**[*3], **Jörg Tiedemann**[3], **André F. T. Martins**[4,5,6], **Hinrich Schütze**[1,2]
[1]Center for Information and Language Processing, LMU Munich
[2]Munich Center for Machine Learning [3]University of Helsinki
[4]Instituto Superior Técnico (Lisbon ELLIS Unit)
[5]Instituto de Telecomunicações [6]Unbabel
linpq@cis.lmu.de, shaoxiong.ji@helsinki.fi

Abstract

Large language models (LLMs) have advanced the state of the art in natural language processing. However, their predominant design for English or a limited set of languages creates a substantial gap in their effectiveness for low-resource languages. To bridge this gap, we introduce MaLA-500, a novel large language model designed to cover an extensive range of 534 languages. To train MaLA-500, we employ vocabulary extension and continued pretraining on LLaMA 2 with Glot500-c. Our intrinsic evaluation demonstrates that MaLA-500 is better at predicting the given texts of low-resource languages than existing multilingual LLMs. Moreover, the extrinsic evaluation of in-context learning shows that MaLA-500 outperforms previous LLMs on SIB200 and Taxi1500 by a significant margin, i.e., 11.68% and 4.82% marco-average accuracy across languages. We release MaLA-500 at https://huggingface.co/MaLA-LM.

1 Introduction

Large Language Models (LLMs), e.g., LLaMA (Touvron et al., 2023a;b), Mistral (Jiang et al., 2023; 2024), and ChatGPT,[1] have shown remarkable performance in natural language understanding and generation. Follow-up studies (Bang et al., 2023; Lai et al., 2023; Ahuja et al., 2023a;b) observe that these English-centric LLMs, such as LLaMA with mainly English as the training data, are capable of handling some high-resource non-English languages, benefiting from the inclusion of non-English language data during pretraining. However, their applicability to low-resource languages is still limited due to data scarcity.

Previous studies have released pretrained multilingual models with mostly encoder-only transformer architectures, e.g., multilingual BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), for around 100 languages. The paradigm shift from encoder-only to decoder-only achieves scalability for large language models with billions of model parameters, leading to the development of open multilingual models. Recently, several generative multilingual LLMs, such as XGLM (Lin et al., 2021), mGPT (Shliazhko et al., 2022), and BLOOM (Scao et al., 2022), have emerged. Notably, the current language coverage for these generative LLMs is limited to up to 60 languages, highlighting the remaining need for further work on massively multilingual LLMs for many natural languages.

ImaniGooghari et al. (2023) have achieved a significant milestone in the realm of massive language adaptation by extending the language coverage of a small-scale multilingual language model, XLM-R (Conneau et al., 2020) - an auto-encoding model with 278M parameters, from 100 languages to an impressive number of 534 languages, and introducing an extended model, Glot500-m with 395M parameters. ImaniGooghari et al. (2023) introduce the Glot500-c corpora spanning 534 languages from 47 language families, and then apply

[*]Equal contribution.
[1]https://openai.com/blog/chatgpt

vocabulary extension and continued pretraining to create Glot500-m. The introduction of Glot500-c mitigates the challenge of data scarcity for low-resource languages. Moreover, the adaptation method is more favorable than training from scratch, as it requires fewer computational resources and emits a smaller carbon footprint. This success serves as a strong motivation for our exploration into the massive language adaptation of LLMs.

This work aims to extend the capabilities of LLMs to encompass a wider range of languages. Existing works like ImaniGooghari et al. (2023) on language adaptation of pretrained models provide extended coverage across a wide linguistic spectrum but are limited to relatively small model sizes - mostly at the hundred million scales, while other works like Yong et al. (2022) extended generative LLMs but are limited to a small number of languages. Our study pushes the boundaries by exploring language adaptation techniques for LLMs with model parameters scaling up to 10 billion for 534 languages. Our investigation delves into generative LLMs with a substantial increase in model parameters and their in-context learning capabilities in diverse languages, especially low-resource languages. This augmentation enables us to enhance contextual and linguistic relevance across a diverse range of languages.

We address the challenges of adapting LLMs to low-resource languages, such as data sparsity, domain-specific vocabulary, and linguistic diversity. Specifically, we study continued pretraining of open LLM, i.e., LLaMA 2 (Touvron et al., 2023b), vocabulary extension, and adaptation techniques, i.e., LoRA low-rank reparameterization (Hu et al., 2022). We deploy distributed training and release MaLA-500 that covers more than 500 languages in various domains. We evaluate MaLA-500 using intrinsic measures on held-out Glot500-c test set and parallel data and extrinsic metrics on downstream benchmarks: SIB200 and Taxi1500. The results show that MaLA-500 outperforms existing open LLMs of close or slightly larger model size. This work broadens the accessibility of LLMs, making them valuable for a more diverse set of language-specific use cases, especially for low-resource ones, and addressing the equality issue by removing language barriers for speakers of many languages, especially those underrepresented languages covered by existing LLMs.

2 Massive Language Adaptation

The principle of massive language adaptation of large language models accommodates the utilization of a massively multilingual corpus (Section 2.1), the strong base LLM (Section 2.2), and the technique for effective language adaptation: vocabulary extension (Section 2.3) and continued pretraining (Section 2.4).

2.1 Data

We use Glot500-c (ImaniGooghari et al., 2023) covering 534 languages[2] as the training data of MaLA-500. See §A for the list of languages with their data amounts. The original number of sentences ranges from 10 thousand to 63 million. Note that Glot500-c does not put full effort into collecting data for high-resource languages but focuses on low-resource languages. We sample languages from the imbalanced dataset according to a multinomial distribution, with $\alpha = 0.3$ for vocabulary extension and continued pretraining. We use different scales for sampling data to be used in model training and vocabulary construction. After sampling, the number of sentences for training ranges from 600 thousand to 8 million per language, leading to 1 billion sentences in total. The number of sentences for vocabulary construction ranges from 30 thousand to 400 thousand, making a total of 50 million sentences.

2.2 Model

We choose LLaMA 2 (Touvron et al., 2023b) to start continual training. LLaMA series models (Touvron et al., 2023a), with model weights released publicly, have gained popularity in the research community. Despite being English-centric compared to their multilingual

[2]We define languages using the ISO 639-3 code combined with the corresponding written script. For example, "`eng_Latn`" represents English written in the Latin script.

counterparts, they have shown remarkable capacity for multiple languages (Ahuja et al., 2023b). We choose the latest LLaMA 2, trained on 2 trillion tokens, as our base model to benefit from its outstanding language capacity. Our study chooses the 7B model with 32 transformer layers, and leaves the extension of LLMs with larger sizes as a future work.

2.3 Vocabulary Extension

The original LLaMA 2's 32,000 tokenizer covers English and a small fraction of other European languages using Latin or Cyrillic scripts. To enhance its capability and encoding efficiency for a broader range of languages, we extend the vocabulary with Glot500-c. Specifically, we initially train a multilingual tokenizer with SentencePiece (Kudo & Richardson, 2018) on the sampled Glot500-c with a vocabulary of 250,000. Subsequently, we merge the trained tokenizer with the original LLaMA 2 tokenizer by taking the union of their vocabularies. As a result, we obtain the MaLA-500's tokenizer with a vocabulary size of 260,164. After vocabulary extension and resizing the embedding layer, the model size becomes 8.6B.

We measure the impact of vocabulary extension on the development set of Glot500-c by analyzing the reduction in segmentation length for each language. The results indicate that the effect of vocabulary extension varies, ranging from 8% (English, `eng_Latn`) to 88% (Oriya, `ori_Orya`). Unsurprisingly, vocabulary extension has a larger effect on languages written in non-Latin scripts than on those in the Latin script. However, for some low-resource languages written in the Latin script, e.g., Kabiyè (`kbp_Latn`) and Vietnamese (`vie_Latn`), the segmentation length is shortened by around 50%.

2.4 Continued Pretraining

We employ continued pretraining for language adaptation with low-rank adaptation (LoRA, Hu et al., 2022) to enable parameter-efficient training, given the limitation of our computing resources. LoRA injects trainable rank decomposition matrices, which approximate the large weight matrices with a lower rank, to the pretrained model weights. It reduces the computational complexity and thus saves the training cost while retaining high model quality (Hu et al., 2022). We continually train the casual language model to update the rank-decomposition matrices, embedding layer, and language modeling head while freezing the transformer weights of pretrained models, allowing the continually trained language model to learn from new data in new languages without completely losing its previous language capacity. Continual training of large language models requires substantial computational resources. We adopt efficient distributed training setups on supercomputers to make the training process feasible.

2.5 Training

Hardware and Software We train our model on the computing cluster with the theoretical peak performance of 2 petaflops on GPU nodes. We deploy distributed training on 24 Nvidia Ampere A100 GPUs. As for software, we utilize the Huggingface Transformers (Wolf et al., 2020), PEFT (Parameter-Efficient Fine-Tuning),[3] and DeepSpeed (Rasley et al., 2020). We use the ZeRO redundancy optimizer (Rajbhandari et al., 2020) and maximize the batch size that fits the memory of each GPU. We employ mixed-precision training using the bfloat16 format.

Hyperparameters The learning rate is set at 3e-4. A weight decay of 0.01 is applied to penalize large weights and mitigate overfitting. The trainable LoRA module targets the query and value matrices. The language model head is not decomposed by a LoRA module but is trained in a full-parameter manner. In our setting, the final model has 10B parameters in total, in which 2B parameters are trainable. The LoRA module is incorporated with a rank of 8, an alpha value of 32, and a dropout rate of 0.05, contributing to the model's adaptability and regularization during training. The context window is 4k. We maximize the batch size

[3] `https://huggingface.co/docs/peft/index`

to fit the memory, making a global batch size of 384. The model undergoes three training epochs. Checkpoints are saved every 500 steps, and we employ early stopping to select the checkpoint that exhibits the most favorable average performance on downstream tasks.

Environmental Impacts We train our model on a carbon-neutral data center, with all electricity generated with renewable hydropower, and the waste heat is utilized in district heating to further reduce CO2 footprint.[4]

3 Evaluation

3.1 Benchmarks and Setup

We consider both intrinsic and extrinsic measures for evaluation. Evaluation dataset statistics are shown in Table 1.

	Datasets	Metric	‖Data‖	‖Lang‖	Domain
Intrinsic	Glot500-c test (ImaniGooghari et al., 2023)	*NLL*	1000	534	Misc
	PBC (Mayer & Cysouw, 2014)	*NLL*	500	370	Bible
Extrinsic	SIB200 (Adelani et al., 2023)	ACC	204	177	Misc
	Taxi1500 (Ma et al., 2023)	ACC	111	351	Bible

Table 1: Evaluation dataset statistics. ‖Data‖: test set size per language. ‖Lang‖: number of evaluated languages. *NLL*: negative log-likelihood. ACC: Accuracy.

For intrinsic evaluation, perplexity is not comparable across models and languages due to different text segmentations. Inspired by Xue et al. (2022); Yu et al. (2023), we instead measure the negative log-likelihood (*NLL*) of the text using the given LLMs.

We concatenate the dataset as the input text and adopt the sliding-window strategy.[5] The evaluation of different LLMs uses the same data with the concatenation of sentences per language, thus making *NLL* model-comparable. In addition, we consider language-comparable *NLL* by measuring *NLL* on parallel data, in which every sample in different languages contains the same semantic information. We report the model-comparable *NLL* of Glot500-c test set covering all 534 considered languages (§3.2), and language-comparable *NLL* on Parallel Bible Corpus (PBC, Mayer & Cysouw, 2014), covering 370 languages (§3.3).

For extrinsic evaluation, we evaluate the few-shot learning capability of MaLA-500 and compare it with other LLMs on SIB200 (Adelani et al., 2023) and Taxi1500 (Ma et al., 2023).

SIB200 is a topic classification dataset. The classification task involves seven classes, namely science/technology, travel, politics, sports, health, entertainment, and geography. Our evaluation spans a diverse set of 177 languages, obtained by intersecting the language sets of SIB200 and Glot500-c. Note that the flores200-based SIB200 evaluation set is included in the training data since Glot500-c includes flores200, but the classification labels are not provided.

Taxi1500 is another text classification dataset spanning 351 languages. It involves six classes, namely, Recommendation, Faith, Description, Sin, Grace, and Violence. Our evaluation efforts aim to cover as many languages as possible. However, the evaluation of massively multilingual language models is a challenging task. Due to the lack of real-world multilingual evaluation benchmarks, we use this benchmark that contains religious content.

For in-context learning evaluation, the evaluated LLM receives a structured prompt, which is the concatenation of few-shot examples and the sample intended for prediction. The format for both a few-shot example and the sample to predict is defined as follows:

Template for SIB200:

[4] `https://www.csc.fi/sustainable-development`

[5] `https://huggingface.co/docs/transformers/en/perplexity`

The topic of the news [sent] is [label]

Template for Taxi1500:

The topic of the verse [sent] is [label]

where [sent] is the sentence for classification, and [label] is the ground truth. [label] is included when the sample serves as a few-shot example but is omitted when predicting the sample. The constructed prompt is then used as input to the LLM. Subsequently, the evaluated LLM is prompted to estimate the probability of the label over the label set based on the provided prompt.

For SIB200, few-shots examples are randomly sampled from the in-language training sets. Since randomly selecting few-shot examples for in-context learning yields random results for both MaLA-500 and previous LLMs on Taxi1500, we consider the retriever-based in-context learning (Liu et al., 2022). Specifically, we use average word embeddings in layer 8 of the Glot500 (ImaniGooghari et al., 2023) for retrieving semantic-similar samples as suggested in previous work (Sabet et al., 2020) for all the compared models. The evaluation process is implemented using the lm-evaluation-harness,[6] and we use accuracy (ACC) to measure the performance of classification.

3.2 Comparison across LLMs

We compare MaLA-500 with LLaMA 2-7B, mGPT-13B, BLOOM-7B1, and XGLM-7.5B on Glot500-c test set, SIB200, Taxi1500 by computing the averaged performance across languages, and the result are given in Table 2. Among the evaluated LLMs, LLaMA 2-7B performs second-best, indicating that LLaMA 2-7B has a strong multilingual capacity and that it is reasonable to select it as the base model. MaLA-500 outperforms all compared LLMs with a close or slightly larger model size across all the evaluated tasks. Notably, compared to LLaMA 2-7B, MaLA-500 gains a lower NLL on the Glot500-c test set by 39.33, and has 14.94% and 4.82% improvements on SIB200 and Taxi1500, respectively. It highlights MaLA-500's substantial contribution to enhancing the multilingual capacity of LLMs.

Model	Glot500-c test ($NLL \downarrow$)	SIB200 (ACC $\uparrow$)	Taxi1500 (ACC $\uparrow$)
LLaMA 2-7B	190.58	42.08	44.07
mGPT-13B	282.46	45.34	40.98
BLOOM-7B1	202.95	44.63	43.98
XGLM-7.5B	205.07	34.36	43.24
MaLA-500	**151.25**	**57.02**	**48.89**

Table 2: Averaged results across languages on Glot500-c test (measured by NLL), SIB200, and Taxi1500 (measured by accuracy (%)) of different LLMs. mGPT has no model with around 7B parameters, so we choose a larger one with 13B parameters. $\downarrow$ indicates the lower, the better. $\uparrow$ indicates the higher, the better. The best results are **bold**.

Figures 1 to 3 provide detailed performance analysis across languages on Glot500-c test, SIB200, and Taxi1500. In those figures, we group scores into different performance bins and display them in different colors. For Glot500-c test, MaLA-500 has more languages achieving better NLL, i.e., 61 languages with NLL less than 100 and 171 languages with NLL between 100 and 150. Besides, MaLA-500 has 54 (10%) languages achieving NLL larger than 200, which may indicate the languages are not well covered by the measured LLM. Nevertheless, the number is much less than other LLMs. For example, the second-best LLM, LLaMA 2-7B, has 231 (43%) languages achieving NLL larger than 200. For both SIB200 and Taxi1500, MaLA-500 surpasses previous LLMs in the sense that it obtains random results in fewer languages and achieves impressive performance in more languages than its counterparts.

[6] `https://github.com/EleutherAI/lm-evaluation-harness`

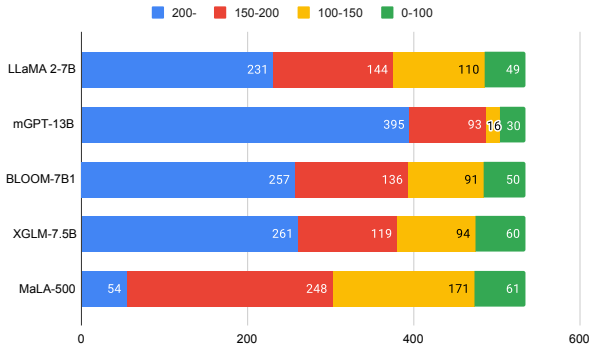


Figure 1: *NLL* (lower is better) on Glot500-c test with the scores grouped into four bins displayed in different colors. X-axis: the number of languages in performance ranges.

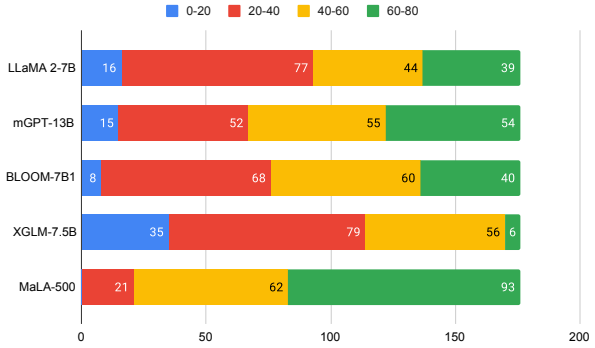


Figure 2: Accuracy (higher is better) on SIB200 with the scores grouped into four bins displayed in different colors. X-axis: the number of languages in performance ranges (%).

3.3 Comparison across Languages

To check in detail how MaLA-500 performs across languages, we check the performance across language families[7] shown in Table 3. We observe that more high-resource language families, e.g., Indo-European (`indo1319`) and Dravidian (`drav1251`), achieve slightly better performance than low-resource language families, e.g., Sino-Tibetan (`sino1245`).

In Table 4, we present a comprehensive analysis of the top 5 performance improvements and declines across languages on SIB200 from MaLA-500 compared to LLaMA 2-7B. We observe that MaLA-500 has substantial improvements on low-resource scripts, e.g., Kannada (`kan_Knda`), while has worse performance on high-resource languages, e.g., Swedish (`swe_Latn`), which have been well covered by LLaMA 2-7B.

In our comprehensive analysis of contributing factors on SIB200, we note that the corpus size of a language exhibits a weak correlation of 0.13 with its performance gain. In contrast, the corpus size of the language family to which a language belongs demonstrates a moderate correlation of 0.40. A moderately high Pearson correlation of 0.53 is observed between the effect of vocabulary extension, i.e., the reduction in segmentation length, and the performance gain. This observation holds true for languages with both non-Latin scripts, such as Kannada (`kan_Knda`), Malayalam (`mal_Mlym`), and Tigrinya (`tir_Ethi`), as well as Latin scripts, such as Igbo (`ibo_Latn`) and Yoruba (`yor_Latn`). It demonstrates the effectiveness of vocabulary extension.

[7]We assign languages to families based on Glottolog: `https://glottolog.org/glottolog/family`.

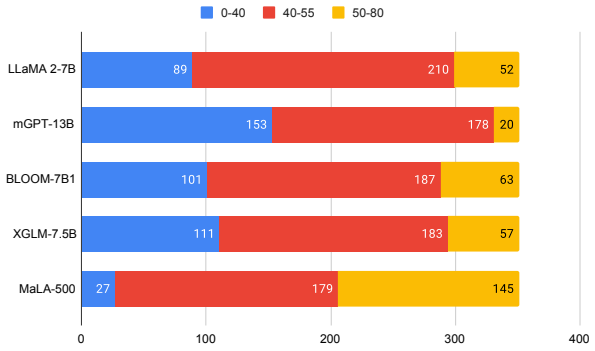


Figure 3: Accuracy (higher is better) on Taxi1500 with the scores grouped into four bins displayed in different colors. X-axis: the number of languages in performance ranges (%).

family	‖Sent‖	PBC (*NLL* ↓)	SIB200 (ACC ↑)	Taxi1500 (ACC ↑)
indo1319	988M	145.35	63.53	53.03
drav1251	135M	**131.29**	56.25	**54.65**
aust1307	113M	147.37	62.83	49.69
turk1311	109M	161.71	57.08	52.55
afro1255	100M	165.46	52.00	43.74
atla1278	57M	141.92	42.90	45.52
ural1272	50M	137.52	**66.67**	48.58
sino1245	29M	155.64	49.30	49.31
other	60M	167.69	55.74	46.67

Table 3: Performance comparison across language families on PBC, SIB200, and Taxi1500. ‖Sent‖: sentence number used for continued pretraining in total. ↓ indicates the lower, the better. ↑ indicates the higher, the better.

3.4 Effect of Number of Shots

Figure 4 illustrates the relationship between accuracy and the number of in-context examples (i.e., shots) on SIB200. As the number of in-context shots increases, there is a corresponding rise in accuracy. Notably, with just 1-shot, accuracy exhibits randomness at 30.88%, indicating 1-shot provides limited information for task learning. The transition from 1 shot to 2 shots/3 shots results in a notable improvement, with performances boosted by 19.83% and 26.14%, respectively. This highlights the effectiveness of increasing the number of shots. MaLA-500 achieves its peak performance at approximately 65% accuracy with 6-10 in-context shots. This may be attributed to the multi-class nature of the SIB200 dataset, necessitating more shots for learning intricate input-label mappings.

In Figure 5, a more nuanced portrayal of results aligns with the observations made in Figure 4. In the realm of 1-shot in-context learning, approximately 50 languages exhibit erratic results. As the number of shots increases, there is a reduction in the number of languages achieving low accuracy (25-50%), coupled with a growing cohort achieving high accuracy (75-100%).

Further examination into individual language trends reveals that some low-resource languages require more shots to achieve better performance (e.g., `pes_Arab` for Persian) or even exhibit poor performance with 10 shots (e.g., `dzo_Tibt` for Dzongkha and `ayr_Latn` for Central Aymara). In contrast, high-resource languages, such as `fra_Latn` for French, demonstrate impressive performance even with fewer shots, and increasing the number of shots results in only marginal improvement.

high end				low end			
Language	LLaMA 2-7B	MaLA-500	Δ	Language	LLaMA 2-7B	MaLA-500	Δ
kan_Knda	17.16	57.35	40.19	swe_Latn	71.08	60.29	-10.79
ckb_Arab	19.61	60.29	40.68	rus_Cyrl	71.57	65.20	-06.37
asm_Beng	17.16	58.82	41.66	dan_Latn	69.12	63.24	-05.88
pan_Guru	14.22	58.82	44.60	pol_Latn	74.51	68.63	-05.88
sin_Sinh	15.20	60.29	45.09	ukr_Cyrl	71.57	65.69	-05.88

Table 4: Results for five languages each with the largest (high end) and smallest (low end) gains from MaLA-500 vs. LLaMA 2-7B for SIB200. Δ indicates the difference between the scores of MaLA-500 and LLaMA 2-7B. See §B for detailed results for each task.

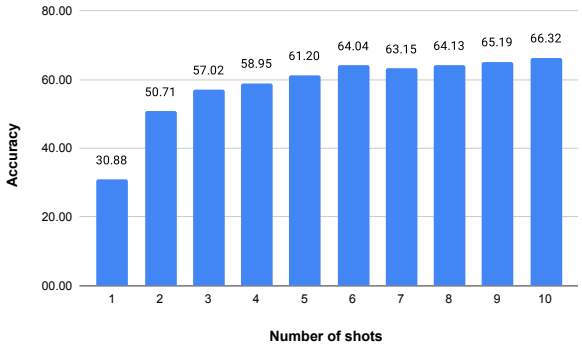


Figure 4: In-context learning macro-average accuracy (%) on SIB200 with different number of shots using MaLA-500.

4 Related Work

4.1 Multilingual Language Models

Language model development has endeavored to broaden the scope of pretraining languages to address multilingual scenarios. Pretrained multilingual models have been able to accommodate up to a hundred or more languages. Noteworthy examples include mBERT Devlin et al. (2019), which supports 104 languages, XLM-R (Conneau et al., 2020) covering 100 languages, mBART (Liu et al., 2020) designed for 25 languages, mT5 (Xue et al., 2021) spanning 101 languages, XGLM (Lin et al., 2021) across 30 languages, GPT-3 covering 118 languages (93% English), mGPT (Shliazhko et al., 2022) accommodating 60 languages, and BLOOM (Scao et al., 2022) supporting 46 languages and 13 programming languages.

Surprisingly, two recent multilingual language models have surpassed the conventional limit by supporting more than 400 languages. Glot500-m (ImaniGooghari et al., 2023) spans 511 languages through vocabulary extension and continued training based on XLM-R. SERENGETI (Adebara et al., 2022) goes even further by supporting 517 African languages and language varieties, written in five different scripts, employing models inspired by both ELECTRA (Clark et al., 2020) and XLM-R. MADLAD (Kudugunta et al., 2023) covers 419 languages and trains an 8B language model from scratch with an adapted UL2 objective (Tay et al., 2022). Our work is concurrent with the MADLAD-400 language model. We distinguish it by: 1) language coverage. Our work covered more than 500 languages, a number comparable to that of encoder-only models and surpassing MADLAD-400 by an additional 100 languages. 2) training methods. We consider continual training to benefit from the learned knowledge of the original models. 3) model architecture. We adopt an open model architecture, i.e., LLaMA, while MADLAD uses decoder-only T5 architecture, which has not been supported by the HuggingFace ecosystem at the time of writing, thus leading to additional difficulty in usage.

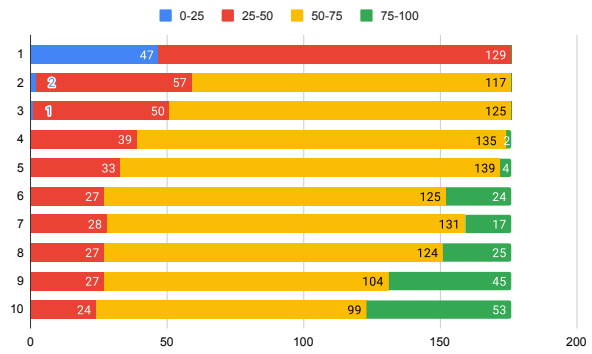


Figure 5: Detailed results of in-context learning on SIB200 using MaLA-500. X-axis: the number of languages in different accuracy ranges (%). Y-axis: number of shots.

4.2 Language Adaptation

Before the advent of LLMs, diverse approaches are employed to adapt small-scale multilingual language models to new languages. These methods include using adapters (Pfeiffer et al., 2020; Üstün et al., 2020; Pfeiffer et al., 2020; Nguyen et al., 2021; Faisal & Anastasopoulos, 2022; Yong et al., 2022), vocabulary extension and substitution (Chau et al., 2020; Wang et al., 2020; Müller et al., 2020; 2021; Pfeiffer et al., 2021; Chen et al., 2023; Downey et al., 2023), leveraging monolingual corpora (Ebrahimi & Kann, 2021; Alabi et al., 2022), and utilizing bilingual lexicons (Wang et al., 2022).

While language models have been scaled up notably, their coverage is limited to a specific set of languages. To address this constraint, various methods have been proposed to expand the applicability of these large language models across a broader range of languages, catering to both general-purpose tasks and specific applications like machine translation. These methods also involve vocabulary extension (Cui et al., 2023), continued pretraining and instruction-tuning (Yong et al., 2022; Cui et al., 2023; Chen et al., 2024; Zhao et al., 2024), and parallel corpora exploitation (Cahyawijaya et al., 2023; Yang et al., 2023; Zhu et al., 2023; Xu et al., 2023). Despite these efforts, massive language adaptation of LLMs for general-purpose tasks across diverse languages, e.g., covering many languages families and more than one hundred languages, remains an area yet to be thoroughly explored.

5 Conclusion and Future Work

We present a pioneering effort in massive language adaptation on LLMs, focusing on extending LLaMA 7B to our model, MaLA-500. This adaptation involves vocabulary extension and continued pretraining with LoRA. Our approach leads to MaLA-500 achieving state-of-the-art in-context learning capabilities, as demonstrated on the benchmarks of SIB200 and Taxi1500. We release the training scripts and model weights publicly to facilitate future research. This work marks a substantial advancement in applying LLMs to a diverse range of languages.

Our future work will focus on further improving the model capacity, for example, on machine translation across many language pairs. Alves et al. (2023) showed that LLMs (LLaMA-7B and LLaMA-13B) exhibited poor performance even on English-centric high-resource language pairs in some cases. Translation with LLMs on low-resource languages is more challenging. The LLaMA-7B model performed poorly in our preliminary experiments. Besides, our pretraining corpus does not intentionally include bilingual texts, and our MaLA-500 model is not instruction-tuned with translation data. We leave the inclusion of bilingual text during continual pretraining, instruction fine-tuning with translation data, and the evaluation on machine translation as future works.

Ethical Statement

LLMs have been known to exhibit biases present in their training data. When extending LLMs to low-resource languages, there is a risk of propagating biases from high-resource languages to underrepresented ones. Careful attention must be paid to mitigate bias and ensure fairness in data collection and model training. The paper aims to make LLMs more accessible for underrepresented languages. Still, there is a risk of creating a digital language divide if certain communities are left out due to limited technological access. Future work would address biases by conducting bias audits on the training data, debiasing the models during generation, and continuously monitoring model outputs.

Reproducibility Statement

We make the following efforts to ensure reproducible research. We release the model weights (`https://huggingface.co/MaLA-LM`) and codes for training and evaluation (`https://github.com/MaLA-LM/mala-500`). We use publicly available evaluation benchmarks which can be obtained freely or by request. The results are reproducible with our released model weights and evaluation scripts,

Acknowledgements

We thank José Pombal for constructive suggestions on training. This work is funded by The European Research Council (grants #740516, #771113 and #758969), EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), and the European Union's Horizon Europe research and innovation programme under grant agreement No 101070350 and from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant #10052546]. The authors wish to acknowledge CSC – IT Center for Science, Finland, for generous computational resources on the Mahti supercomputer and LUMI supercomputer through the LUMI extreme scale access (MOOMIN and LumiNMT). Shaoxiong Ji and Peiqin Lin acknowledge travel support from ELISE (GA no 951847).

References

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. SERENGETI: Massively multilingual language models for africa. *arXiv preprint arXiv:2212.10785*, 2022.

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. *CoRR*, abs/2309.07445, 2023. doi: 10.48550/arXiv.2309.07445. URL `https://doi.org/10.48550/arXiv.2309.07445`.

Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. MEGA: multilingual evaluation of generative AI. *CoRR*, abs/2303.12528, 2023a. doi: 10.48550/arXiv.2303.12528. URL `https://doi.org/10.48550/arXiv.2303.12528`.

Sanchit Ahuja, Divyanshu Aggarwal, Varun Gumma, Ishaan Watts, Ashutosh Sathe, Millicent Ochieng, Rishav Hada, Prachi Jain, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. MEGAVERSE: benchmarking large language models across languages, modalities, models and tasks. *CoRR*, abs/2311.07463, 2023b. doi: 10.48550/ARXIV.2311.07463. URL `https://doi.org/10.48550/arXiv.2311.07463`.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. Adapting pre-trained language models to african languages via multilingual adaptive fine-tuning. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao

Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (eds.), *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pp. 4336–4349. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.382.

Duarte Alves, Nuno Guerreiro, João Alves, José Pombal, Ricardo Rei, José de Souza, Pierre Colombo, and André FT Martins. Steering large language models for machine translation with finetuning and in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 11127–11148, 2023.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *CoRR*, abs/2302.04023, 2023. doi: 10.48550/arXiv.2302.04023. URL https://doi.org/10.48550/arXiv.2302.04023.

Samuel Cahyawijaya, Holy Lovenia, Tiezheng Yu, Willy Chung, and Pascale Fung. Instruct-align: Teaching novel languages with to LLMs through alignment-based cross-lingual instruction. *CoRR*, abs/2305.13627, 2023. doi: 10.48550/arXiv.2305.13627. URL https://doi.org/10.48550/arXiv.2305.13627.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. Parsing with multilingual bert, a small treebank, and a small corpus. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 1324–1334. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.118. URL https://doi.org/10.18653/v1/2020.findings-emnlp.118.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Barry Haddow, and Kenneth Heafield. Monolingual or multilingual instruction tuning: Which makes a better Alpaca. In *Findings of the Association for Computational Linguistics: EACL*, 2024. URL https://doi.org/10.48550/arXiv.2309.08958.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Ifeoluwa Adelani, Pontus Stenetorp, Sebastian Riedel, and Mikel Artetxe. Improving language plasticity via pretraining with active forgetting. *CoRR*, abs/2307.01163, 2023. doi: 10.48550/arXiv.2307.01163. URL https://doi.org/10.48550/arXiv.2307.01163.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, 2020.

Yiming Cui, Ziqing Yang, and Xin Yao. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *CoRR*, abs/2304.08177, 2023. doi: 10.48550/ARXIV.2304.08177. URL https://doi.org/10.48550/arXiv.2304.08177.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

C. M. Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. *CoRR*, abs/2309.04679, 2023. doi: 10.48550/arXiv.2309.04679. URL https://doi.org/10.48550/arXiv.2309.04679.

Abteen Ebrahimi and Katharina Kann. How to adapt your pretrained multilingual model to 1600 languages. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 4555–4567. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.351. URL https://doi.org/10.18653/v1/2021.acl-long.351.

Fahim Faisal and Antonios Anastasopoulos. Phylogeny-inspired adaptation of multilingual models to new languages. *CoRR*, abs/2205.09634, 2022. doi: 10.48550/arXiv.2205.09634. URL https://doi.org/10.48550/arXiv.2205.09634.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ayyoob ImaniGooghari, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. Glot500: Scaling multilingual corpora and language models to 500 languages. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1082–1117, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.61. URL https://aclanthology.org/2023.acl-long.61.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

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, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pp. 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-2012. URL https://doi.org/10.18653/v1/d18-2012.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, et al. MADLAD-400: A multilingual and document-level large audited dataset. *arXiv preprint arXiv:2309.04662*, 2023.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. *CoRR*, abs/2304.05613, 2023. doi: 10.48550/arXiv.2304.05613. URL https://doi.org/10.48550/arXiv.2304.05613.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668, 2021. URL https://arxiv.org/abs/2112.10668.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In Eneko Agirre, Marianna Apidianaki, and Ivan Vulic (eds.), *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pp. 100–114. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.DEELIO-1.10. URL https://doi.org/10.18653/v1/2022.deelio-1.10.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

Chunlan Ma, Ayyoob ImaniGooghari, Haotian Ye, Ehsaneddin Asgari, and Hinrich Schütze. Taxi1500: A multilingual dataset for text classification in 1500 languages, 2023.

Thomas Mayer and Michael Cysouw. Creating a massively parallel bible corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pp. 3158–3163. European Language Resources Association (ELRA), 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/summaries/220.html.

Benjamin Müller, Benoît Sagot, and Djamé Seddah. Can multilingual language models transfer to an unseen dialect? A case study on north african arabizi. *CoRR*, abs/2005.00318, 2020. URL https://arxiv.org/abs/2005.00318.

Benjamin Müller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. When being unseen from mbert is just the beginning: Handling new languages with multilingual language models. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 448–462. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.38. URL https://doi.org/10.18653/v1/2021.naacl-main.38.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In Dimitra Gkatzia and Djamé Seddah (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, April 19-23, 2021*, pp. 80–90. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.eacl-demos.10. URL https://doi.org/10.18653/v1/2021.eacl-demos.10.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. MAD-X: an adapter-based framework for multi-task cross-lingual transfer. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 7654–7673. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.617. URL https://doi.org/10.18653/v1/2020.emnlp-main.617.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. Unks everywhere: Adapting multilingual language models to new scripts. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 10186–10203. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.800. URL https://doi.org/10.18653/v1/2021.emnlp-main.800.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3505–3506, 2020.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 1627–1643. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.147. URL https://doi.org/10.18653/v1/2020.findings-emnlp.147.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. mGPT: Few-shot learners go multilingual. *CoRR*, abs/2204.07580, 2022. doi: 10.48550/arXiv.2204.07580. URL https://doi.org/10.48550/arXiv.2204.07580.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2022.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a. doi: 10.48550/arXiv.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023b. doi: 10.48550/arXiv.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. Udapter: Language adaptation for truly universal dependency parsing. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 2302–2315. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.180. URL https://doi.org/10.18653/v1/2020.emnlp-main.180.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 863–877. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.acl-long.61.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. Extending multilingual BERT to low-resource languages. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pp. 2649–2656. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.240. URL https://doi.org/10.18653/v1/2020.findings-emnlp.240.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. A paradigm shift in machine translation: Boosting translation performance of large language models. *CoRR*, abs/2309.11674, 2023. doi: 10.48550/ARXIV.2309.11674. URL https://doi.org/10.48550/arXiv.2309.11674.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 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*, pp. 483–498, 2021.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Trans. Assoc. Comput. Linguistics*, 10:291–306, 2022. doi: 10.1162/tacl_a_00461. URL https://doi.org/10.1162/tacl_a_00461.

Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. Bigtrans: Augmenting large language models with multilingual translation capability over 100 languages. *CoRR*, abs/2305.18098, 2023. doi: 10.48550/arXiv.2305.18098. URL https://doi.org/10.48550/arXiv.2305.18098.

Zheng Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M. Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Indra Winata, Stella Biderman, Dragomir Radev, and Vassilina Nikoulina. BLOOM+1: adding language support to BLOOM for zero-shot prompting. *CoRR*, abs/2212.09535, 2022. doi: 10.48550/arXiv.2212.09535. URL https://doi.org/10.48550/arXiv.2212.09535.

Lili Yu, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. MEGABYTE: predicting million-byte sequences with multiscale transformers. *CoRR*, abs/2305.07185, 2023. doi: 10.48550/arXiv.2305.07185. URL https://doi.org/10.48550/arXiv.2305.07185.

Jun Zhao, Zhihao Zhang, Qi Zhang, Tao Gui, and Xuanjing Huang. LLaMA beyond English: An empirical study on language capability transfer. *arXiv preprint arXiv:2401.01055*, 2024.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. Extrapolating large language models to non-english by aligning languages. *CoRR*, abs/2308.04948, 2023. doi: 10.48550/arXiv.2308.04948. URL https://doi.org/10.48550/arXiv.2308.04948.

A Languages

The list of languages of Glot500-c used to train MaLA-500 with the number of available sentences and language family information for each language is available in Tables 5, 6 and 7.

Lang	‖*Sent*‖	Family	Lang	‖*Sent*‖	Family	Lang	‖*Sent*‖	Family
hbs_Latn	63411156	indo1319	hin_Deva	7046700	indo1319	ton_Latn	1216118	aust1307
mal_Mlym	48098273	drav1251	kor_Hang	6468444	kore1284	tah_Latn	1190747	aust1307
aze_Latn	46300705		ory_Orya	6266475	indo1319	lat_Latn	1179913	indo1319
guj_Gujr	45738685	indo1319	urd_Arab	6009594	indo1319	srn_Latn	1172349	indo1319
ben_Beng	43514870	indo1319	swa_Latn	5989369		ewe_Latn	1161605	atla1278
kan_Knda	41836495	drav1251	sqi_Latn	5526836	indo1319	bem_Latn	1111969	atla1278
tel_Telu	41580525	drav1251	bel_Cyrl	5319675	indo1319	efi_Latn	1082621	atla1278
mlt_Latn	40654838	afro1255	afr_Latn	5157787	indo1319	bis_Latn	1070170	indo1319
fra_Latn	39197581	indo1319	nno_Latn	4899103	indo1319	orm_Latn	1067699	
spa_Latn	37286756	indo1319	tat_Cyrl	4708088	turk1311	haw_Latn	1062491	aust1307
eng_Latn	36122761	indo1319	ast_Latn	4683554	indo1319	hmo_Latn	1033636	pidg1258
fil_Latn	33493255	aust1307	mon_Cyrl	4616960	mong1349	kat_Geor	1004297	kart1248
nob_Latn	32869205	indo1319	hbs_Cyrl	4598073	indo1319	pag_Latn	983637	aust1307
rus_Cyrl	31787973	indo1319	hau_Latn	4368483	afro1255	loz_Latn	964418	atla1278
deu_Latn	31015993	indo1319	sna_Latn	4019596	atla1278	fry_Latn	957422	indo1319
tur_Latn	29184662	turk1311	msa_Latn	3929084		mya_Mymr	945180	sino1245
pan_Guru	29052537	indo1319	som_Latn	3916769	afro1255	nds_Latn	944715	indo1319
mar_Deva	28748897	indo1319	srp_Cyrl	3864091	indo1319	run_Latn	943828	atla1278
por_Latn	27824391	indo1319	mlg_Latn	3715802		pnb_Arab	899895	indo1319
nld_Latn	25061426	indo1319	zul_Latn	3580113	atla1278	rar_Latn	894515	aust1307
ara_Arab	24524122		arz_Arab	3488224	afro1255	fij_Latn	887134	aust1307
zho_Hani	24143786		nya_Latn	3409030	atla1278	wls_Latn	882167	aust1307
ita_Latn	23539857	indo1319	tam_Taml	3388255	drav1251	ckb_Arab	874441	indo1319
ind_Latn	23018106	aust1307	hat_Latn	3226932	indo1319	ven_Latn	860249	atla1278
ell_Grek	22033282	indo1319	uzb_Latn	3223485	turk1311	zsm_Latn	859947	aust1307
bul_Cyrl	21823004	indo1319	sot_Latn	3205510	atla1278	chv_Cyrl	859863	turk1311
swe_Latn	20725883	indo1319	uzb_Cyrl	3029947	turk1311	lua_Latn	854359	atla1278
ces_Latn	20376340	indo1319	cos_Latn	3015055	indo1319	que_Latn	838486	
isl_Latn	19547941	indo1319	als_Latn	2954874	indo1319	sag_Latn	771048	atla1278
pol_Latn	19339945	indo1319	amh_Ethi	2862985	afro1255	guw_Latn	767918	atla1278
ron_Latn	19190217	indo1319	sun_Latn	2586011	aust1307	bre_Latn	748954	indo1319
dan_Latn	19174573	indo1319	war_Latn	2584810	aust1307	toi_Latn	745385	atla1278
hun_Latn	18800025	ural1272	div_Thaa	2418687	indo1319	pus_Arab	731992	indo1319
tgk_Cyrl	18659517	indo1319	yor_Latn	2392359	atla1278	che_Cyrl	728201	nakh1245
srp_Latn	18371769	indo1319	fao_Latn	2365271	indo1319	pis_Latn	714783	indo1319
fas_Arab	18277593		uzn_Cyrl	2293672	turk1311	kon_Latn	685194	
ceb_Latn	18149215	aust1307	smo_Latn	2290439	aust1307	oss_Cyrl	683517	indo1319
heb_Hebr	18128962	afro1255	bak_Cyrl	2264196	turk1311	hyw_Armn	679819	indo1319
hrv_Latn	17882932	indo1319	ilo_Latn	2106531	aust1307	iso_Latn	658789	atla1278
glg_Latn	17852274	indo1319	tso_Latn	2100708	atla1278	nan_Latn	656389	sino1245
fin_Latn	16730388	ural1272	mri_Latn	2046850	aust1307	lub_Latn	654390	atla1278
slv_Latn	15719210	indo1319	hmn_Latn	1903898		lim_Latn	652078	indo1319
vie_Latn	15697827	aust1305	asm_Beng	1882353	indo1319	tuk_Latn	649411	turk1311
mkd_Cyrl	14717004	indo1319	hil_Latn	1798875	aust1307	tir_Ethi	649117	afro1255
slk_Latn	14633631	indo1319	nso_Latn	1619354	atla1278	tgk_Latn	636541	indo1319
nor_Latn	14576191	indo1319	ibo_Latn	1543820	atla1278	yua_Latn	610052	maya1287
est_Latn	13600579		kin_Latn	1521612	atla1278	min_Latn	609065	aust1307
ltz_Latn	12997242	indo1319	hye_Armn	1463123	indo1319	lue_Latn	599429	atla1278
eus_Latn	12775959		oci_Latn	1449128	indo1319	khm_Khmr	590429	aust1305
lit_Latn	12479626	indo1319	lin_Latn	1408460	atla1278	tum_Latn	589857	atla1278
kaz_Cyrl	12378727	turk1311	tpi_Latn	1401844	indo1319	tll_Latn	586530	atla1278
lav_Latn	12143980	indo1319	twi_Latn	1400979	atla1278	ekk_Latn	582595	ural1272
bos_Latn	11014744	indo1319	kir_Cyrl	1397566	turk1311	lug_Latn	566948	atla1278
epo_Latn	8737198	arti1236	pap_Latn	1360138	indo1319	niu_Latn	566715	aust1307
cat_Latn	8648271	indo1319	nep_Deva	1317291	indo1319	tzo_Latn	540262	maya1287
tha_Thai	7735209	taik1256	azj_Latn	1315834	turk1311	mah_Latn	534614	aust1307
ukr_Cyrl	7462046	indo1319	bcl_Latn	1284493	aust1307	tvl_Latn	521556	aust1307
tgl_Latn	7411064	aust1307	xho_Latn	1262364	atla1278	jav_Latn	516833	aust1307
sin_Sinh	7293178	indo1319	cym_Latn	1244783	indo1319	vec_Latn	514240	indo1319
gle_Latn	7225513	indo1319	gaa_Latn	1222307	atla1278	jpn_Jpan	510722	japo1237

Table 5: List of languages of Glot500-c (Part I).

Lang	‖*Sent*‖	Family	Lang	‖*Sent*‖	Family	Lang	‖*Sent*‖	Family
lus_Latn	509250	sino1245	kmb_Latn	296269	atla1278	ncx_Latn	162558	utoa1244
crs_Latn	508755	indo1319	zai_Latn	277632	otom1299	qug_Latn	162500	quec1387
kqn_Latn	507913	atla1278	gym_Latn	274512	chib1249	rmn_Latn	162069	indo1319
ndo_Latn	496613	atla1278	bod_Tibt	273489	sino1245	cjk_Latn	160645	atla1278
snd_Arab	488730	indo1319	nde_Latn	269931	atla1278	arb_Arab	159884	afro1255
yue_Hani	484700	sino1245	fon_Latn	268566	atla1278	kea_Latn	158047	indo1319
tiv_Latn	483064	atla1278	ber_Latn	264426		mck_Latn	157521	atla1278
kua_Latn	473535	atla1278	nbl_Latn	259158	atla1278	arn_Latn	155882	arau1255
kwy_Latn	473274	atla1278	kmr_Latn	256677	indo1319	pdt_Latn	155485	indo1319
hin_Latn	466175	indo1319	guc_Latn	249044	araw1281	her_Latn	154827	atla1278
iku_Cans	465011		mam_Latn	248348	maya1287	gla_Latn	152563	indo1319
kal_Latn	462430	eski1264	nia_Latn	247406	aust1307	kmr_Cyrl	151728	indo1319
tdt_Latn	459818	aust1307	nyn_Latn	241992	atla1278	mwl_Latn	150054	indo1319
gsw_Latn	449240	indo1319	cab_Latn	240101	araw1281	nav_Latn	147702	atha1245
mfe_Latn	447435	indo1319	top_Latn	239232	toto1251	ksw_Mymr	147674	sino1245
swc_Latn	446378	atla1278	tog_Latn	231969	atla1278	mxv_Latn	147591	otom1299
mon_Latn	437950	mong1349	mco_Latn	231209	mixe1284	hif_Latn	147261	indo1319
mos_Latn	437666	atla1278	tzh_Latn	230706	maya1287	wol_Latn	146992	atla1278
kik_Latn	437228	atla1278	pms_Latn	227748	indo1319	sme_Latn	146803	ural1272
cnh_Latn	436667	sino1245	wuu_Hani	224088	sino1245	gom_Latn	143937	indo1319
gil_Latn	434529	aust1307	plt_Latn	220413	aust1307	bum_Latn	141673	atla1278
pon_Latn	434522	aust1307	yid_Hebr	220214	indo1319	mgr_Latn	138953	atla1278
umb_Latn	431589	atla1278	ada_Latn	219427	atla1278	ahk_Latn	135068	sino1245
lvs_Latn	422952	indo1319	iba_Latn	213615	aust1307	kur_Arab	134160	indo1319
sco_Latn	411591	indo1319	kek_Latn	209932	maya1287	bas_Latn	133436	atla1278
ori_Orya	410827		koo_Latn	209375	atla1278	bin_Latn	133256	atla1278
arg_Latn	410683	indo1319	sop_Latn	206501	atla1278	tsz_Latn	133251	tara1323
kur_Latn	407169	indo1319	kac_Latn	205542	sino1245	sid_Latn	130406	afro1255
dhv_Latn	405711	aust1307	qvi_Latn	205447	quec1387	diq_Latn	128908	indo1319
luo_Latn	398974	nilo1247	cak_Latn	204472	maya1287	srd_Latn	127064	
lun_Latn	395764	atla1278	kbp_Latn	202877	atla1278	tcf_Latn	126050	otom1299
nzi_Latn	394247	atla1278	ctu_Latn	201662	maya1287	bzj_Latn	124958	indo1319
gug_Latn	392227	tupi1275	kri_Latn	201087	indo1319	udm_Cyrl	121705	ural1272
bar_Latn	387070	indo1319	mau_Latn	199134	otom1299	cce_Latn	120636	atla1278
bci_Latn	384059	atla1278	scn_Latn	199068	indo1319	meu_Latn	120273	aust1307
chk_Latn	380596	aust1307	tyv_Cyrl	198649	turk1311	chw_Latn	119751	atla1278
roh_Latn	377067	indo1319	ina_Latn	197315	arti1236	cbk_Latn	118789	indo1319
aym_Latn	373329	ayma1253	btx_Latn	193701	aust1307	ibg_Latn	118733	aust1307
yap_Latn	358929	aust1307	nch_Latn	193129	utoa1244	bhw_Latn	117381	aust1307
ssw_Latn	356561	atla1278	ncj_Latn	192962	utoa1244	ngu_Latn	116851	utoa1244
quz_Latn	354781	quec1387	pau_Latn	190529	aust1307	nyy_Latn	115914	atla1278
sah_Cyrl	352697	turk1311	toj_Latn	189651	maya1287	szl_Latn	112496	indo1319
tsn_Latn	350954	atla1278	pcm_Latn	187594	indo1319	ish_Latn	111814	atla1278
lmo_Latn	348135	indo1319	dyu_Latn	186367	mand1469	naq_Latn	109747	khoe1240
ido_Latn	331239	arti1236	kss_Latn	185868	atla1278	toh_Latn	107583	atla1278
abk_Cyrl	321578	abkh1242	afb_Arab	183694	afro1255	ttj_Latn	106925	atla1278
zne_Latn	318871	atla1278	urh_Latn	182214	atla1278	nse_Latn	105189	atla1278
quy_Latn	311040	quec1387	quc_Latn	181559	maya1287	hsb_Latn	104802	indo1319
kam_Latn	310659	atla1278	new_Deva	181427	sino1245	ami_Latn	104559	aust1307
bbc_Latn	310420	aust1307	yao_Latn	179965	atla1278	alz_Latn	104392	nilo1247
vol_Latn	310399	arti1236	ngl_Latn	178498	atla1278	apc_Arab	102392	afro1255
wal_Latn	309873	gong1255	nyu_Latn	177483	atla1278	vls_Latn	101900	indo1319
uig_Arab	307302	turk1311	kab_Latn	176015	afro1255	mhr_Cyrl	100474	ural1272
vmw_Latn	306899	atla1278	tuk_Cyrl	175769	turk1311	djk_Latn	99234	indo1319
kwn_Latn	305362	atla1278	xmf_Geor	174994	kart1248	wes_Latn	98492	indo1319
pam_Latn	303737	aust1307	ndc_Latn	174305	atla1278	gkn_Latn	97041	atla1278
seh_Latn	300243	atla1278	san_Deva	165616	indo1319	grc_Grek	96986	indo1319
tsc_Latn	298442	atla1278	nba_Latn	163485	atla1278	hbo_Hebr	96484	afro1255
nyk_Latn	297976	atla1278	bpy_Beng	162838	indo1319	swh_Latn	95776	atla1278

Table 6: List of languages of Glot500-c (Part II).

Lang	‖*Sent*‖	Family	Lang	‖*Sent*‖	Family	Lang	‖*Sent*‖	Family
alt_Cyrl	95148	turk1311	mny_Latn	50581	atla1278	csy_Latn	34126	sino1245
rmn_Grek	94533	indo1319	gkp_Latn	50549	mand1469	azb_Arab	33758	turk1311
miq_Latn	94343	misu1242	kat_Latn	50424	kart1248	csb_Latn	33743	indo1319
kaa_Cyrl	88815	turk1311	bjn_Latn	49068	aust1307	tpm_Latn	33517	atla1278
kos_Latn	88603	aust1307	acr_Latn	48886	maya1287	quw_Latn	33449	quec1387
grn_Latn	87568		dtp_Latn	48468	aust1307	rmy_Cyrl	33351	indo1319
lhu_Latn	87255	sino1245	lam_Latn	46853	atla1278	ixl_Latn	33289	maya1287
lzh_Hani	86035	sino1245	bik_Latn	46561		mbb_Latn	33240	aust1307
ajp_Arab	83297	afro1255	poh_Latn	46454	maya1287	pfl_Latn	33148	indo1319
cmn_Hani	80745	sino1245	phm_Latn	45862	atla1278	pcd_Latn	32867	indo1319
gcf_Latn	80737	indo1319	hrx_Latn	45716	indo1319	tlh_Latn	32863	arti1236
rmn_Cyrl	79925	indo1319	quh_Latn	45566	quec1387	suz_Deva	32811	sino1245
kjh_Cyrl	79262	turk1311	hyw_Cyrl	45379	indo1319	gcr_Latn	32676	indo1319
rng_Latn	78177	atla1278	rue_Cyrl	45369	indo1319	jbo_Latn	32619	arti1236
mgh_Latn	78117	atla1278	eml_Latn	44630	indo1319	tbz_Latn	32264	atla1278
xmv_Latn	77896	aust1307	acm_Arab	44505	afro1255	bam_Latn	32150	mand1469
ige_Latn	77114	atla1278	tob_Latn	44473	guai1249	prk_Latn	32085	aust1305
rmy_Latn	76991	indo1319	ach_Latn	43974	nilo1247	jam_Latn	32048	indo1319
srm_Latn	76884	indo1319	vep_Latn	43076	ural1272	twx_Latn	32028	atla1278
bak_Latn	76809	turk1311	npi_Deva	43072	indo1319	nmf_Latn	31997	sino1245
gur_Latn	76151	atla1278	tok_Latn	42820	arti1236	caq_Latn	31903	aust1305
idu_Latn	75106	atla1278	sgs_Latn	42467	indo1319	rop_Latn	31889	indo1319
yom_Latn	74818	atla1278	lij_Latn	42447	indo1319	tca_Latn	31852	ticu1244
tdx_Latn	74430	aust1307	myv_Cyrl	42147	ural1272	yan_Latn	31775	misu1242
mzn_Arab	73719	indo1319	tih_Latn	41873	aust1307	xav_Latn	31765	nucl1710
cfm_Latn	70227	sino1245	tat_Latn	41640	turk1311	bih_Deva	31658	
zpa_Latn	69237	otom1299	lfn_Latn	41632	arti1236	cuk_Latn	31612	chib1249
kbd_Cyrl	67914	abkh1242	cgg_Latn	41196	atla1278	kjb_Latn	31471	maya1287
lao_Laoo	66966	taik1256	ful_Latn	41188	atla1278	hne_Deva	31465	indo1319
nap_Latn	65826	indo1319	gor_Latn	41174	aust1307	wbm_Latn	31394	aust1305
qub_Latn	64973	quec1387	ile_Latn	40984	arti1236	zlm_Latn	31345	aust1307
oke_Latn	64508	atla1278	ium_Latn	40683	hmon1336	tui_Latn	31161	atla1278
ote_Latn	64224	otom1299	teo_Latn	40203	nilo1247	ifb_Latn	30980	aust1307
bsb_Latn	63634	aust1307	kia_Latn	40035	atla1278	izz_Latn	30894	atla1278
ogo_Latn	61901	atla1278	crh_Cyrl	39985	turk1311	rug_Latn	30857	aust1307
abn_Latn	61830	atla1278	crh_Latn	39896	turk1311	aka_Latn	30704	atla1278
ldi_Latn	61827	atla1278	enm_Latn	39809	indo1319	pxm_Latn	30698	book1242
ayr_Latn	61570	ayma1253	sat_Olck	39614	aust1305	kmm_Latn	30671	sino1245
gom_Deva	61140	indo1319	mad_Latn	38993	aust1307	mcn_Latn	30666	afro1255
bba_Latn	61123	atla1278	cac_Latn	38812	maya1287	ifa_Latn	30621	aust1307
aln_Latn	60989	indo1319	hnj_Latn	38611	hmon1336	dln_Latn	30620	sino1245
leh_Latn	59944	atla1278	ksh_Latn	38130	indo1319	ext_Latn	30605	indo1319
ban_Latn	59805	aust1307	ikk_Latn	38071	atla1278	ksd_Latn	30550	aust1307
ace_Latn	59333	aust1307	sba_Latn	38040	cent2225	mzh_Latn	30517	mata1289
pes_Arab	57511	indo1319	zom_Latn	37013	sino1245	llb_Latn	30480	atla1278
skg_Latn	57228	aust1307	bqc_Latn	36881	mand1469	hra_Latn	30472	sino1245
ary_Arab	56933	afro1255	bim_Latn	36835	atla1278	mwm_Latn	30432	cent2225
hus_Latn	56176	maya1287	mdy_Ethi	36370	gong1255	krc_Cyrl	30353	turk1311
glv_Latn	55641	indo1319	bts_Latn	36216	aust1307	tuc_Latn	30349	aust1307
fat_Latn	55609	atla1278	gya_Latn	35902	atla1278	mrw_Latn	30304	aust1307
frr_Latn	55254	indo1319	ajg_Latn	35631	atla1278	pls_Latn	30136	otom1299
mwn_Latn	54805	atla1278	agw_Latn	35585	aust1307	rap_Latn	30102	aust1307
mai_Deva	54687	indo1319	kom_Cyrl	35249	ural1272	fur_Latn	30052	indo1319
dua_Latn	53392	atla1278	knv_Latn	35196		kaa_Latn	30031	turk1311
dzo_Tibt	52732	sino1245	giz_Latn	35040	afro1255	prs_Arab	26823	indo1319
ctd_Latn	52135	sino1245	hui_Latn	34926	nucl1709	san_Latn	25742	indo1319
nnb_Latn	52041	atla1278	kpg_Latn	34900	aust1307	som_Arab	14199	afro1255
sxn_Latn	51749	aust1307	zea_Latn	34426	indo1319	uig_Latn	9637	turk1311
mps_Latn	50645	tebe1251	aoj_Latn	34349	nucl1708	hau_Arab	9593	afro1255

Table 7: List of languages of Glot500-c (Part III).

B Detailed Results

Detailed results of evaluation are shown in Tables 8-15 (NLL on Glot500-c), Tables 16-21 (NLL on PBC), Tables 22-23 (ACC on SIB200), and Tables 24-29 (ACC on Taxi1500).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
abk_Cyrl	234.09	249.16	258.26	231.44	164.61
abn_Latn	140.01	197.81	153.58	152.90	111.86
ace_Latn	235.15	332.18	244.00	259.64	168.79
ach_Latn	179.03	227.84	194.55	197.05	161.01
acm_Arab	119.15	153.09	106.29	101.35	135.82
acr_Latn	301.73	399.80	321.79	316.49	194.71
ada_Latn	132.76	168.56	150.19	137.99	103.17
afb_Arab	134.03	169.73	112.55	110.59	152.58
afr_Latn	52.43	84.47	73.24	75.60	64.25
agw_Latn	228.22	318.95	246.48	242.04	152.59
ahk_Latn	229.45	377.60	245.81	241.21	163.96
ajg_Latn	146.48	185.41	170.89	155.21	113.83
ajp_Arab	153.34	199.79	129.62	124.24	164.80
aka_Latn	163.59	223.13	166.49	185.41	131.50
aln_Latn	191.62	259.76	218.75	267.34	143.64
als_Latn	191.60	271.51	219.17	260.14	155.23
alt_Cyrl	199.25	220.77	200.70	215.71	139.18
alz_Latn	167.89	214.64	185.35	171.34	155.03
amh_Ethi	328.25	834.56	407.68	550.50	268.11
ami_Latn	122.67	168.42	131.77	132.36	109.13
aoj_Latn	318.62	495.44	340.07	316.36	196.64
apc_Arab	131.19	153.97	106.78	109.24	145.81
ara_Arab	111.05	155.64	80.72	84.86	140.73
arb_Arab	166.93	318.76	135.76	137.80	173.03
arg_Latn	173.62	306.23	171.32	178.40	160.08
arn_Latn	202.09	292.40	204.32	216.04	163.87
ary_Arab	198.80	309.90	184.82	176.58	173.37
arz_Arab	122.74	248.72	95.61	100.43	131.75
asm_Beng	264.49	409.59	172.35	311.81	184.77
ast_Latn	208.41	325.35	184.93	192.86	178.77
aym_Latn	143.36	183.42	149.06	154.45	117.28
ayr_Latn	274.31	342.40	288.57	293.48	185.87
azb_Arab	254.60	293.24	273.20	285.61	162.94
aze_Latn	156.58	230.45	195.32	189.59	110.56
azj_Latn	168.12	228.08	212.31	199.86	126.98
bak_Cyrl	274.50	348.47	288.93	307.95	169.00
bak_Latn	191.06	259.97	196.98	213.41	152.50
bam_Latn	195.29	251.28	203.50	215.62	171.51
ban_Latn	205.77	297.97	213.20	213.89	186.89
bar_Latn	210.97	287.33	234.73	208.66	188.90
bas_Latn	137.53	172.78	143.37	147.13	110.71
bba_Latn	233.68	286.30	258.58	238.94	164.18
bbc_Latn	172.78	216.78	181.59	170.06	148.89
bci_Latn	176.81	223.93	190.52	189.46	171.00
bcl_Latn	149.22	209.44	162.25	174.40	132.55
bel_Cyrl	110.77	174.19	142.62	147.27	85.11
bem_Latn	182.62	222.50	198.45	150.51	158.31
ben_Beng	92.79	162.83	50.33	55.42	73.86
ber_Latn	88.37	120.03	87.79	101.52	71.90
bhw_Latn	186.42	245.14	194.41	188.81	155.12
bih_Deva	248.12	422.46	176.37	204.17	180.31
bik_Latn	151.63	218.03	173.42	187.11	137.28
bim_Latn	229.29	284.29	244.21	245.34	166.16
bin_Latn	137.28	175.41	152.32	152.02	109.51
bis_Latn	165.83	250.17	179.61	190.13	130.32
bjn_Latn	200.57	302.58	202.67	199.15	182.65
bod_Tibt	437.54	1690.09	461.35	80.21	286.05
bos_Latn	87.13	175.82	131.95	149.85	110.92
bpy_Beng	251.20	471.67	154.31	172.17	155.64
bqc_Latn	208.00	266.53	226.49	205.65	153.58
bre_Latn	222.93	276.71	208.07	260.44	184.35
bsb_Latn	236.62	358.90	275.10	306.64	204.50
bts_Latn	214.80	292.93	232.31	217.74	156.31
btx_Latn	169.13	227.44	181.86	174.25	148.25
bul_Cyrl	47.01	90.81	77.70	42.90	57.12
bum_Latn	183.88	237.35	194.64	195.91	156.33
bzj_Latn	167.62	244.15	188.25	194.46	137.81

Table 8: Detailed results of *NLL* on Glot500-c (Part I).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
cab_Latn	222.05	292.04	234.53	237.57	168.63
cac_Latn	293.47	395.52	310.33	301.30	192.22
cak_Latn	295.24	394.87	317.52	309.03	200.69
caq_Latn	240.00	323.71	264.17	257.49	164.95
cat_Latn	94.68	212.17	83.26	86.26	130.00
cbk_Latn	143.05	221.60	145.69	159.41	137.96
cce_Latn	178.45	226.07	190.01	192.54	152.70
ceb_Latn	136.44	278.02	164.94	183.55	123.31
ces_Latn	44.83	98.77	68.48	76.15	58.42
cfm_Latn	240.20	305.25	252.92	256.79	185.94
cgg_Latn	121.16	160.92	127.35	129.19	107.91
che_Cyrl	199.15	272.63	203.57	197.17	158.57
chk_Latn	189.52	258.69	201.19	200.61	145.98
chv_Cyrl	246.19	292.36	252.81	229.56	157.91
chw_Latn	139.07	174.73	142.88	121.98	121.16
cjk_Latn	125.30	158.06	134.03	128.75	106.21
ckb_Arab	372.24	437.95	370.20	521.30	243.30
cmn_Hani	52.17	92.04	40.75	49.81	62.30
cnh_Latn	185.01	242.39	198.20	198.57	147.90
cos_Latn	192.02	323.30	210.38	211.96	185.03
crh_Cyrl	236.43	282.79	239.67	260.03	141.08
crh_Latn	149.67	240.28	168.79	157.01	131.91
crs_Latn	153.11	202.53	153.34	87.81	129.39
csb_Latn	238.86	336.99	261.46	294.41	166.29
csy_Latn	226.53	299.52	249.53	245.14	172.03
ctd_Latn	210.45	276.87	227.39	224.34	158.35
ctu_Latn	216.90	310.89	226.68	220.32	157.27
cuk_Latn	233.42	325.97	252.00	247.83	190.81
cym_Latn	233.91	369.64	306.05	332.89	217.29
dan_Latn	43.75	84.32	69.51	66.96	54.56
deu_Latn	37.46	68.68	49.65	33.88	53.45
dhv_Latn	121.21	170.85	126.68	128.57	95.81
diq_Latn	174.75	265.78	180.00	190.78	147.56
div_Thaa	314.55	565.83	314.34	17.32	153.76
djk_Latn	188.44	249.39	201.50	207.16	163.39
dln_Latn	217.51	288.73	231.93	238.10	165.40
dtp_Latn	267.22	373.92	279.80	287.18	184.75
dua_Latn	131.20	169.64	136.03	129.20	109.86
dyu_Latn	186.37	237.65	193.19	205.47	157.89
dzo_Tibt	238.61	842.40	244.70	47.40	154.48
efi_Latn	178.91	251.07	205.96	203.93	134.40
ekk_Latn	155.86	223.64	194.37	89.18	141.19
ell_Grek	52.85	86.68	67.98	36.04	54.45
eml_Latn	213.57	278.33	224.10	225.17	163.91
eng_Latn	30.45	62.73	31.32	34.36	48.60
enm_Latn	79.08	193.74	108.20	119.78	87.78
epo_Latn	68.89	99.75	79.80	87.72	70.22
est_Latn	70.18	100.28	88.33	40.53	67.38
eus_Latn	79.07	87.15	48.33	45.59	70.49
ewe_Latn	208.53	269.62	218.53	195.99	148.78
ext_Latn	216.92	338.22	211.26	231.30	177.17
fao_Latn	202.04	284.61	227.56	263.89	165.45
fas_Arab	138.13	193.21	163.46	166.76	133.69
fat_Latn	134.67	180.66	144.54	144.50	106.86
fij_Latn	159.86	219.85	191.04	137.83	147.71
fil_Latn	120.89	206.21	162.04	161.84	120.27
fin_Latn	46.88	86.18	79.58	35.79	58.35
fon_Latn	237.19	295.74	256.54	262.29	160.24
fra_Latn	32.26	63.71	31.08	32.74	49.22
frr_Latn	192.91	299.41	206.26	211.00	144.13
fry_Latn	191.87	247.81	205.02	221.64	168.86
ful_Latn	447.47	550.03	457.25	511.87	339.38
fur_Latn	231.23	313.99	234.38	250.02	183.57
gaa_Latn	188.66	232.67	222.71	158.83	146.37
gcf_Latn	132.36	173.10	130.03	91.07	103.54
gcr_Latn	113.22	157.83	115.02	79.46	94.40
gil_Latn	175.92	237.54	187.79	181.71	154.60
giz_Latn	244.47	332.32	268.61	266.29	168.09

Table 9: Detailed results of *NLL* on Glot500-c (Part II).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
gkn_Latn	223.58	304.46	253.54	245.24	167.81
gkp_Latn	261.56	358.97	280.80	270.48	186.41
gla_Latn	220.92	382.20	293.89	315.23	210.51
gle_Latn	203.10	345.45	276.11	299.80	206.52
glg_Latn	120.88	204.76	108.43	122.45	132.58
glv_Latn	232.86	326.69	247.79	265.04	182.93
gom_Deva	328.82	462.17	324.77	358.50	233.15
gom_Latn	244.57	318.36	259.71	257.90	209.13
gor_Latn	217.70	326.26	232.98	239.37	168.23
grc_Grek	126.86	277.73	181.00	127.62	141.80
grn_Latn	293.70	382.11	298.10	316.62	204.94
gsw_Latn	180.67	226.37	199.03	171.72	157.34
guc_Latn	241.99	340.92	257.19	234.87	183.29
gug_Latn	197.04	258.55	201.92	214.05	158.39
guj_Gujr	118.82	291.38	74.12	194.71	90.02
gur_Latn	222.48	311.22	243.52	233.99	173.11
guw_Latn	210.29	215.37	235.91	246.28	146.55
gya_Latn	242.48	350.56	274.82	258.26	170.00
gym_Latn	231.32	324.92	249.32	191.13	178.06
hat_Latn	237.00	341.48	251.07	150.39	201.88
hau_Arab	173.08	330.75	130.96	129.69	230.02
hau_Latn	228.21	300.72	257.22	265.65	191.68
haw_Latn	190.18	300.25	217.54	213.20	174.30
hbo_Hebr	140.73	315.06	194.19	200.98	155.08
hbs_Cyrl	206.87	503.80	370.83	417.41	225.22
hbs_Latn	209.02	451.95	333.95	375.92	223.11
heb_Hebr	48.34	63.09	58.19	63.73	56.05
her_Latn	140.31	172.32	146.72	136.86	109.29
hif_Latn	396.80	613.23	471.65	465.81	371.92
hil_Latn	145.89	207.79	161.39	182.01	126.09
hin_Deva	142.07	289.53	105.86	106.38	166.12
hin_Latn	150.11	247.31	166.34	164.94	176.00
hmn_Latn	241.00	375.11	282.60	284.95	182.91
hmo_Latn	165.38	236.46	178.12	142.53	133.19
hne_Deva	201.66	298.38	171.37	184.06	161.30
hnj_Latn	231.56	324.18	263.80	278.94	141.56
hra_Latn	215.87	271.46	228.66	229.46	169.57
hrv_Latn	43.03	82.09	63.02	69.82	54.08
hrx_Latn	131.34	182.33	140.90	135.17	105.20
hsb_Latn	182.90	293.15	211.79	235.34	127.71
hui_Latn	297.34	388.25	319.23	318.32	197.57
hun_Latn	45.03	79.27	75.21	79.06	59.30
hus_Latn	247.96	352.19	260.90	258.32	180.85
hye_Armn	286.18	602.02	372.75	454.38	202.92
hyw_Armn	145.46	263.04	186.63	213.52	110.19
hyw_Cyrl	162.17	231.84	171.73	165.61	117.61
iba_Latn	150.03	192.62	157.75	151.54	133.67
ibg_Latn	115.37	152.94	119.10	122.19	106.08
ibo_Latn	232.57	333.59	223.37	296.17	184.97
ido_Latn	140.94	273.88	153.94	164.61	121.37
idu_Latn	153.00	209.20	162.53	157.46	106.21
ifa_Latn	252.33	328.31	270.66	266.03	172.20
ifb_Latn	257.92	340.56	278.23	272.79	183.83
ige_Latn	148.85	199.02	173.80	176.02	111.50
ikk_Latn	249.37	330.44	284.76	310.26	166.74
iku_Cans	261.21	877.71	343.18	496.50	174.80
ile_Latn	100.28	199.76	105.32	115.20	100.35
ilo_Latn	172.24	227.41	186.11	208.36	146.96
ina_Latn	209.38	408.99	230.14	236.01	201.92
ind_Latn	42.59	69.80	35.50	36.82	56.03
ish_Latn	126.54	178.71	144.92	146.15	101.29
isl_Latn	103.40	156.83	127.49	139.76	83.51
iso_Latn	148.38	175.75	168.85	167.42	104.67
ita_Latn	39.35	79.02	49.94	40.47	53.36
ium_Latn	247.28	361.48	264.84	266.46	167.10
ixl_Latn	327.09	506.74	353.08	348.23	222.05
izz_Latn	301.73	400.14	346.61	361.39	193.24
jam_Latn	204.69	291.31	223.99	231.17	157.87

Table 10: Detailed results of *NLL* on Glot500-c (Part III).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
jav_Latn	208.92	275.29	212.60	220.31	180.00
jbo_Latn	103.91	200.82	109.86	112.61	117.25
jpn_Jpan	136.26	301.32	197.23	149.70	150.43
kaa_Cyrl	281.21	363.07	300.20	317.13	146.98
kaa_Latn	284.60	354.51	292.04	309.67	192.43
kab_Latn	192.58	264.51	185.56	216.46	161.31
kac_Latn	210.47	267.38	223.77	249.95	166.44
kal_Latn	240.15	262.90	259.85	155.45	182.71
kam_Latn	153.84	194.60	156.10	186.00	115.78
kan_Knda	216.22	556.40	146.43	355.75	175.17
kat_Geor	302.53	413.90	435.47	483.85	239.51
kat_Latn	184.94	308.06	217.07	208.25	184.20
kaz_Cyrl	257.67	341.78	280.01	297.13	187.85
kbd_Cyrl	212.12	229.63	198.20	202.85	146.86
kbp_Latn	232.17	306.53	257.45	246.16	161.08
kea_Latn	118.17	159.93	121.92	122.29	105.69
kek_Latn	234.79	332.19	244.69	228.87	164.18
khm_Khmr	257.14	815.56	317.46	437.56	167.88
kia_Latn	222.01	298.21	245.77	236.71	164.39
kik_Latn	208.26	277.92	213.92	237.26	159.49
kin_Latn	206.40	237.66	174.18	234.91	168.37
kir_Cyrl	265.65	308.15	277.34	313.50	175.71
kjb_Latn	263.79	353.35	280.16	278.13	179.76
kjh_Cyrl	200.11	251.59	211.84	217.34	147.81
kmb_Latn	132.84	166.09	137.48	118.00	112.99
kmm_Latn	246.57	330.77	263.79	266.44	180.90
kmr_Cyrl	224.23	284.40	226.51	221.22	154.70
kmr_Latn	183.95	220.51	194.67	215.02	142.36
knv_Latn	430.56	581.45	456.13	427.27	232.18
kom_Cyrl	224.18	302.71	249.08	213.41	134.88
kon_Latn	112.77	131.61	116.89	119.41	96.00
koo_Latn	132.73	167.13	144.33	134.74	111.26
kor_Hang	129.20	224.06	180.21	95.71	151.37
kos_Latn	146.15	191.23	153.05	154.26	123.85
kpg_Latn	221.52	321.94	246.33	245.73	148.93
kqn_Latn	125.33	149.57	128.12	109.60	106.08
krc_Cyrl	247.13	292.86	248.83	267.39	167.05
kri_Latn	166.50	240.92	193.15	192.19	140.20
ksd_Latn	198.81	269.96	210.59	212.57	138.81
ksh_Latn	204.72	261.51	220.93	218.50	161.62
kss_Latn	310.35	477.02	335.25	300.31	226.38
ksw_Mymr	210.34	266.24	226.59	154.55	124.78
kua_Latn	179.05	206.09	187.92	151.87	140.72
kur_Arab	402.78	464.44	400.97	550.57	253.61
kur_Latn	633.22	779.47	678.30	748.20	424.98
kwn_Latn	136.80	170.23	141.88	111.31	107.21
kwy_Latn	131.93	160.78	137.77	134.01	110.55
lam_Latn	209.07	276.89	228.12	203.17	176.61
lao_Laoo	405.48	978.35	435.37	583.11	225.06
lat_Latn	167.49	274.19	186.97	210.22	183.32
lav_Latn	193.22	257.06	227.60	252.31	162.80
ldi_Latn	178.84	230.26	185.58	191.19	160.61
leh_Latn	216.80	273.56	230.25	201.57	172.92
lfn_Latn	232.59	368.62	246.45	258.76	187.82
lhu_Latn	209.10	365.95	220.56	219.50	142.74
lij_Latn	328.66	483.81	345.62	348.28	249.64
lim_Latn	199.01	290.80	236.94	239.44	180.52
lin_Latn	161.88	173.63	158.33	180.17	135.66
lit_Latn	163.71	220.62	195.08	225.98	147.53
llb_Latn	135.01	180.06	146.51	135.39	120.02
lmo_Latn	222.22	378.21	247.54	242.80	182.01
loz_Latn	179.54	194.46	185.77	142.19	147.86
ltz_Latn	190.70	303.65	202.02	174.00	169.36
lua_Latn	126.47	147.86	131.94	102.71	102.36
lub_Latn	136.45	143.64	140.96	99.41	111.01
lue_Latn	128.48	158.40	135.27	129.94	103.72
lug_Latn	225.72	318.09	221.56	272.90	196.21
lun_Latn	135.96	170.81	142.71	136.26	113.31

Table 11: Detailed results of *NLL* on Glot500-c (Part IV).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
luo_Latn	177.43	224.04	194.07	187.72	156.23
lus_Latn	192.97	251.35	203.37	212.72	163.95
lvs_Latn	154.85	211.40	185.87	198.99	138.63
lzh_Hani	149.57	215.19	130.32	153.38	151.15
mad_Latn	232.71	325.38	245.39	249.29	176.81
mah_Latn	178.50	246.26	188.98	183.17	145.35
mai_Deva	245.94	389.84	189.93	223.00	185.84
mal_Mlym	96.92	171.55	57.45	129.61	72.46
mam_Latn	232.38	315.16	247.28	244.43	189.28
mar_Deva	85.13	143.31	55.38	103.23	70.08
mau_Latn	186.46	333.61	204.91	193.09	161.45
mbb_Latn	282.70	410.99	309.56	307.47	175.50
mck_Latn	191.94	244.49	202.17	191.28	152.16
mcn_Latn	207.28	276.32	220.35	230.56	158.99
mco_Latn	271.45	368.23	281.55	260.70	206.54
mdy_Ethi	306.26	529.46	293.68	369.22	166.26
meu_Latn	177.74	235.19	188.09	168.43	143.62
mfe_Latn	147.50	194.41	143.47	92.23	129.23
mgh_Latn	193.72	257.45	207.05	200.68	166.17
mgr_Latn	183.96	226.09	194.25	149.77	160.18
mhr_Cyrl	230.20	298.73	235.59	236.71	167.55
min_Latn	161.40	266.18	164.13	170.30	166.91
miq_Latn	207.63	276.27	228.42	223.78	160.37
mkd_Cyrl	81.62	144.52	112.99	98.33	74.40
mlg_Latn	185.23	250.78	189.32	226.85	148.82
mlt_Latn	109.60	184.08	139.75	146.69	85.14
mny_Latn	133.04	170.16	135.30	126.14	112.38
mon_Cyrl	397.63	535.59	446.51	555.16	249.95
mon_Latn	354.75	411.54	383.60	383.02	282.85
mos_Latn	197.23	229.14	206.05	212.55	159.69
mps_Latn	347.99	496.26	378.75	366.78	213.10
mri_Latn	154.38	247.38	181.49	179.85	134.55
mrw_Latn	235.11	306.78	250.41	253.18	169.69
msa_Latn	164.05	261.28	155.14	151.77	190.44
mwl_Latn	275.26	410.83	270.47	280.98	202.89
mwm_Latn	293.40	430.46	315.11	294.17	162.95
mwn_Latn	131.84	162.91	138.48	111.20	123.37
mxv_Latn	206.13	324.92	222.48	222.86	171.82
mya_Mymr	383.74	576.49	472.04	277.91	252.84
myv_Cyrl	267.24	357.29	263.68	276.10	188.74
mzh_Latn	257.70	370.86	285.03	276.60	169.96
mzn_Arab	192.75	263.60	200.51	204.50	136.03
nan_Latn	172.36	311.98	186.78	200.62	153.96
nap_Latn	159.24	246.36	179.36	167.94	151.29
naq_Latn	195.43	261.60	207.68	207.27	150.47
nav_Latn	258.40	380.88	284.18	286.04	181.13
nba_Latn	123.68	154.25	130.25	126.08	99.29
nbl_Latn	175.10	238.64	194.74	211.98	154.90
nch_Latn	206.55	287.53	220.86	221.43	183.56
ncj_Latn	185.32	260.91	201.13	196.79	173.80
ncx_Latn	115.71	168.08	121.23	122.70	98.71
ndc_Latn	167.38	222.72	176.18	184.45	158.24
nde_Latn	169.75	235.54	185.98	211.96	151.45
ndo_Latn	192.10	227.02	204.45	150.28	149.69
nds_Latn	195.44	272.44	213.17	204.47	184.93
nep_Deva	232.93	425.83	167.54	291.83	210.52
new_Deva	169.64	330.40	128.26	135.07	103.54
ngl_Latn	134.87	177.05	140.92	115.46	104.59
ngu_Latn	205.16	282.39	215.65	213.78	167.56
nia_Latn	202.30	269.59	214.87	196.19	167.95
niu_Latn	105.04	142.53	111.71	113.36	88.11
nld_Latn	37.77	65.47	55.52	51.54	51.45
nmf_Latn	222.98	290.53	242.04	246.36	167.31
nnb_Latn	200.64	248.60	210.13	212.23	161.20
nno_Latn	138.72	234.11	192.13	199.51	146.16
nob_Latn	50.27	96.43	78.24	73.64	59.05
nor_Latn	78.04	146.26	126.19	123.99	99.50
npi_Deva	212.50	399.24	143.71	290.95	166.12

Table 12: Detailed results of *NLL* on Glot500-c (Part V).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
nse_Latn	176.20	234.62	184.77	174.57	161.52
nso_Latn	170.49	227.97	170.96	201.29	142.72
nya_Latn	203.45	299.12	222.51	224.89	175.69
nyk_Latn	131.13	166.47	142.89	138.50	105.26
nyn_Latn	174.51	229.51	189.05	194.14	149.17
nyu_Latn	126.29	172.40	132.49	127.67	99.26
nyy_Latn	215.07	271.23	234.37	220.80	168.74
nzi_Latn	191.21	256.55	219.47	209.42	152.30
oci_Latn	202.93	343.11	207.95	210.24	185.26
ogo_Latn	134.14	185.86	149.15	143.22	118.08
oke_Latn	131.90	166.07	146.98	149.55	102.72
ori_Orya	323.51	839.80	179.33	665.17	203.94
orm_Latn	225.00	334.29	288.51	313.08	201.60
ory_Orya	232.83	572.34	134.20	474.35	164.65
oss_Cyrl	229.49	279.89	229.34	227.24	151.79
ote_Latn	237.06	362.46	254.31	241.61	176.73
pag_Latn	173.32	223.39	184.05	184.76	157.30
pam_Latn	259.01	373.98	274.16	280.47	237.10
pan_Guru	242.88	510.70	153.50	395.85	180.54
pap_Latn	162.79	213.88	174.63	173.16	138.20
pau_Latn	176.42	243.03	188.84	187.10	150.24
pcd_Latn	144.96	228.79	143.18	150.08	140.39
pcm_Latn	159.00	346.35	182.00	179.53	147.50
pdt_Latn	192.69	252.34	199.07	199.80	144.40
pes_Arab	153.46	199.83	175.97	179.97	139.01
pfl_Latn	220.11	315.47	241.84	225.74	176.25
phm_Latn	117.81	162.32	128.28	125.57	100.73
pis_Latn	153.04	237.95	173.91	179.21	130.98
pls_Latn	237.55	350.88	251.43	251.35	175.28
plt_Latn	159.36	220.84	158.06	193.44	131.96
pms_Latn	132.94	257.06	137.39	146.18	106.52
pnb_Arab	345.25	418.35	279.85	240.35	237.22
poh_Latn	389.80	589.86	417.71	416.42	230.35
pol_Latn	44.19	82.29	66.66	71.91	60.02
pon_Latn	177.92	236.38	190.47	189.62	149.49
por_Latn	37.00	66.01	35.14	33.91	48.72
prk_Latn	220.51	301.85	230.42	238.15	148.46
prs_Arab	163.01	218.38	191.40	195.64	141.99
pus_Arab	259.45	327.43	277.81	340.38	203.38
pxm_Latn	299.37	391.48	317.99	307.01	180.85
qub_Latn	210.38	265.76	222.82	172.89	152.70
quc_Latn	248.16	320.50	271.51	258.06	187.13
que_Latn	144.31	170.69	154.62	96.53	121.19
qug_Latn	176.78	225.11	187.16	136.85	143.62
quh_Latn	257.89	293.32	275.35	187.44	175.55
quw_Latn	154.10	205.67	162.83	142.63	142.35
quy_Latn	177.21	202.67	190.48	125.92	139.15
quz_Latn	180.20	211.40	192.52	123.67	142.21
qvi_Latn	178.08	234.53	188.58	156.22	145.79
rap_Latn	204.53	354.21	219.29	226.89	158.90
rar_Latn	169.22	249.96	191.91	189.56	168.88
rmn_Cyrl	129.46	181.44	143.84	137.02	102.76
rmn_Grek	135.82	190.47	141.78	125.21	92.56
rmn_Latn	133.75	175.58	146.05	143.75	112.55
rmy_Cyrl	135.65	184.00	147.87	137.05	109.18
rmy_Latn	189.65	244.12	198.92	205.44	168.77
rng_Latn	122.59	150.36	125.06	129.81	104.16
roh_Latn	235.38	312.78	242.57	253.77	161.16
ron_Latn	44.70	84.55	68.14	74.76	54.82
rop_Latn	233.05	351.35	257.34	275.70	155.36
rue_Cyrl	223.89	402.99	299.90	265.32	179.38
rug_Latn	257.50	348.10	277.13	275.47	169.94
run_Latn	184.59	218.12	161.96	207.06	157.49
rus_Cyrl	65.34	155.39	116.17	67.59	84.56
sag_Latn	162.87	194.78	175.45	155.14	149.65
sah_Cyrl	383.55	455.30	382.36	423.03	218.84
san_Deva	182.35	287.49	189.83	201.00	186.46
san_Latn	242.46	324.45	278.75	282.93	199.18

Table 13: Detailed results of *NLL* on Glot500-c (Part VI).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
sat_Olck	654.37	3377.97	667.66	40.17	311.96
sba_Latn	272.45	372.47	303.48	293.62	167.13
scn_Latn	236.20	355.24	263.10	270.02	191.69
sco_Latn	147.94	341.79	193.24	193.20	170.39
seh_Latn	173.46	231.41	177.38	174.40	138.70
sgs_Latn	248.33	313.78	251.35	277.73	182.16
sid_Latn	135.53	180.29	147.72	139.17	114.24
sin_Sinh	82.29	173.16	114.00	137.98	70.77
skg_Latn	128.02	172.67	131.34	145.32	116.16
slk_Latn	62.89	116.82	86.67	103.39	63.93
slv_Latn	42.18	85.28	64.75	73.49	55.26
sme_Latn	288.98	357.31	301.64	295.23	205.46
smo_Latn	220.26	338.16	250.74	252.76	190.00
sna_Latn	221.02	311.60	221.92	258.38	189.74
snd_Arab	209.83	264.61	217.96	260.53	163.07
som_Arab	230.91	410.59	192.88	175.01	265.88
som_Latn	235.21	346.36	286.69	312.99	212.51
sop_Latn	176.17	207.78	188.41	157.21	167.90
sot_Latn	200.82	271.71	205.51	235.65	157.18
spa_Latn	37.28	70.48	34.26	38.65	53.39
sqi_Latn	207.58	295.58	241.22	296.90	172.78
srd_Latn	228.12	341.00	242.74	251.01	179.87
srm_Latn	229.46	318.77	250.79	246.75	173.83
srn_Latn	161.18	183.34	171.30	179.59	132.77
srp_Cyrl	45.22	100.88	77.59	81.95	57.85
srp_Latn	33.66	57.89	43.91	46.74	42.31
ssw_Latn	194.10	264.22	212.99	230.20	165.70
sun_Latn	220.72	314.99	228.18	237.07	203.32
suz_Deva	255.00	400.13	262.34	257.16	157.30
swa_Latn	156.02	208.21	125.78	94.55	151.68
swc_Latn	103.75	133.69	98.32	71.66	102.14
swe_Latn	42.72	82.20	68.89	60.92	56.18
swh_Latn	178.28	223.65	151.05	97.98	161.49
sxn_Latn	243.81	346.98	263.76	260.44	183.47
szl_Latn	132.77	348.45	156.37	177.33	111.32
tah_Latn	114.41	158.18	124.60	121.22	101.40
tam_Taml	231.12	444.83	152.34	146.94	205.53
tat_Cyrl	251.96	301.03	256.66	276.29	159.44
tat_Latn	248.71	338.00	261.10	278.92	186.84
tbz_Latn	273.90	352.17	299.25	281.11	164.62
tca_Latn	306.13	452.15	328.77	316.81	174.51
tcf_Latn	133.72	193.63	138.67	133.58	102.94
tdt_Latn	158.16	217.96	172.56	182.04	130.27
tdx_Latn	125.88	167.70	130.54	135.72	113.29
tel_Telu	94.93	152.33	54.92	47.52	72.02
teo_Latn	193.42	250.17	206.10	193.68	159.90
tgk_Cyrl	313.76	369.08	333.83	342.42	196.57
tgk_Latn	296.86	412.46	342.18	352.59	248.69
tgl_Latn	56.44	94.00	76.30	77.15	64.98
tha_Thai	192.70	331.25	242.28	116.12	175.60
tih_Latn	233.30	329.24	255.15	254.70	158.13
tir_Ethi	267.84	579.39	319.29	424.77	189.73
tiv_Latn	133.38	168.19	140.43	126.42	116.08
tlh_Latn	163.23	258.64	183.94	184.72	111.43
tll_Latn	138.57	167.75	152.44	126.10	105.23
tob_Latn	299.95	450.25	316.77	324.19	182.95
tog_Latn	127.47	165.93	133.37	115.35	102.88
toh_Latn	181.85	238.80	196.33	194.76	146.50
toi_Latn	185.04	233.23	194.93	164.94	165.33
toj_Latn	232.66	311.24	239.53	236.11	198.17
tok_Latn	46.19	61.55	50.56	43.88	47.57
ton_Latn	172.88	243.40	178.94	190.23	141.17
top_Latn	221.27	303.90	232.90	223.12	212.88
tpi_Latn	139.90	209.92	155.89	170.67	120.65
tpm_Latn	214.33	280.83	241.97	231.70	154.99
tsc_Latn	131.42	150.62	130.29	132.42	104.44
tsn_Latn	209.69	291.69	203.77	245.18	169.85
tso_Latn	182.87	208.89	176.69	194.90	142.27

Table 14: Detailed results of *NLL* on Glot500-c (Part VII).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
tsz_Latn	183.82	253.97	200.16	176.35	153.98
ttj_Latn	133.35	174.08	142.10	146.60	112.98
tuc_Latn	325.34	444.23	346.52	291.84	180.84
tui_Latn	247.40	330.20	266.54	265.71	181.39
tuk_Cyrl	196.40	248.01	210.45	219.39	143.19
tuk_Latn	217.31	235.95	217.78	238.66	155.71
tum_Latn	184.51	236.91	190.41	153.36	153.44
tur_Latn	48.52	66.76	60.61	34.71	63.33
tvl_Latn	114.81	156.00	123.30	121.62	97.96
twi_Latn	169.99	229.39	171.42	190.50	139.81
twx_Latn	123.52	172.96	130.82	135.08	106.56
tyv_Cyrl	270.89	314.09	275.97	304.11	174.60
tzh_Latn	195.49	274.47	208.05	202.10	162.63
tzo_Latn	223.14	324.35	237.54	228.92	173.78
udm_Cyrl	222.45	277.14	231.98	219.71	160.30
uig_Arab	336.01	432.84	320.43	463.38	207.25
uig_Latn	254.59	292.36	270.85	285.12	203.29
ukr_Cyrl	101.99	240.89	173.79	160.03	136.57
umb_Latn	129.60	165.59	135.06	139.75	100.07
urd_Arab	77.96	105.77	53.61	51.92	81.62
urh_Latn	145.52	153.19	164.21	161.54	108.55
uzb_Cyrl	307.70	353.00	332.86	314.77	178.07
uzb_Latn	307.44	363.61	357.04	383.26	220.74
uzn_Cyrl	233.89	270.06	254.96	247.92	145.01
vec_Latn	163.22	261.93	181.25	168.76	170.03
ven_Latn	190.45	233.75	198.94	198.18	151.65
vep_Latn	316.12	456.77	326.76	243.40	192.08
vie_Latn	108.65	169.92	86.74	91.41	138.89
vls_Latn	200.17	292.89	242.66	253.44	171.13
vmw_Latn	141.25	176.25	143.12	107.10	102.92
vol_Latn	94.00	260.01	85.47	87.18	83.77
wal_Latn	190.62	261.79	201.98	177.73	158.07
war_Latn	127.41	249.86	146.46	166.29	153.84
wbm_Latn	222.06	311.86	234.78	240.27	150.33
wes_Latn	64.78	106.54	73.37	73.73	86.61
wls_Latn	114.80	157.93	125.63	124.58	99.38
wol_Latn	197.17	251.63	171.70	208.78	173.01
wuu_Hani	152.90	283.11	127.83	152.82	145.05
xav_Latn	350.22	619.11	379.76	371.80	201.63
xho_Latn	224.10	315.12	219.57	265.57	187.35
xmf_Geor	260.61	315.58	316.49	376.33	170.15
xmv_Latn	125.37	168.73	129.48	139.97	111.94
yan_Latn	228.46	314.62	248.18	243.68	165.66
yao_Latn	196.25	253.72	209.77	198.91	166.06
yap_Latn	197.98	274.54	212.39	209.00	169.09
yid_Hebr	437.75	571.08	480.37	590.32	295.70
yom_Latn	176.11	220.86	184.62	189.29	150.95
yor_Latn	233.75	283.33	193.55	286.20	185.60
yua_Latn	195.86	284.05	208.08	205.70	161.16
yue_Hani	74.79	131.83	62.91	83.80	74.28
zai_Latn	170.49	223.03	179.18	188.38	148.03
zea_Latn	174.18	271.42	212.95	222.74	155.52
zho_Hani	57.89	99.40	48.19	55.24	70.80
zlm_Latn	106.37	176.09	92.63	93.81	118.56
zne_Latn	127.57	167.13	134.43	115.53	104.95
zom_Latn	214.60	277.57	233.64	228.48	170.06
zpa_Latn	127.29	180.39	129.07	132.30	107.04
zsm_Latn	102.42	171.64	92.39	94.59	123.31
zul_Latn	208.94	340.58	235.91	257.18	192.84
all	190.58	282.46	202.95	205.07	151.25

Table 15: Detailed results of *NLL* on Glot500-c (Part VIII).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
ace_Latn	137.43	196.93	144.50	152.49	97.89
ach_Latn	113.66	152.08	123.29	125.31	102.45
acr_Latn	177.86	233.22	188.27	182.33	114.27
afr_Latn	80.43	132.25	116.34	129.21	95.33
agw_Latn	130.32	186.58	136.17	138.27	95.93
ahk_Latn	175.75	291.31	187.63	179.54	116.76
aka_Latn	98.41	135.74	99.46	108.01	78.20
aln_Latn	101.54	147.77	115.11	139.73	82.71
als_Latn	93.47	134.99	106.68	127.57	78.53
alt_Cyrl	122.23	146.47	125.04	134.88	90.21
alz_Latn	107.41	139.39	116.48	109.62	102.48
amh_Ethi	100.60	255.43	121.36	161.34	98.24
aoj_Latn	175.25	270.87	185.66	171.29	114.19
arb_Arab	94.03	186.47	77.67	78.12	104.22
arn_Latn	141.08	205.53	143.63	154.61	113.59
ary_Arab	128.97	212.49	125.35	118.25	104.58
arz_Arab	80.59	185.56	64.91	66.52	92.22
asm_Beng	123.16	196.03	79.80	147.49	101.89
ayr_Latn	149.96	188.09	154.45	157.13	106.66
azb_Arab	134.00	160.29	139.49	144.68	93.09
aze_Latn	97.68	131.80	113.03	106.38	90.96
bak_Cyrl	134.49	169.96	133.79	150.46	93.49
bam_Latn	109.68	147.72	110.24	118.13	91.88
ban_Latn	138.98	195.92	147.93	149.04	111.51
bar_Latn	114.49	154.37	121.74	113.26	108.65
bba_Latn	132.00	166.51	146.31	131.24	96.87
bbc_Latn	110.66	143.87	117.12	107.02	100.17
bci_Latn	117.42	156.47	125.70	124.26	126.80
bcl_Latn	101.39	146.46	109.03	116.78	88.46
bel_Cyrl	92.30	137.26	110.12	118.30	88.89
bem_Latn	125.52	158.97	135.60	104.16	107.57
ben_Beng	111.68	194.50	68.00	77.83	105.61
bhw_Latn	124.94	169.40	130.40	123.48	101.65
bim_Latn	124.64	162.78	132.77	130.01	96.33
bis_Latn	126.46	196.19	136.72	148.29	95.85
bod_Tibt	138.16	525.70	144.33	30.40	105.99
bqc_Latn	113.18	149.13	122.07	112.76	91.11
bre_Latn	120.49	151.33	111.97	139.13	105.99
bts_Latn	111.90	154.57	120.61	110.17	89.16
btx_Latn	118.13	163.25	128.43	125.76	103.19
bul_Cyrl	66.25	124.78	104.01	42.33	85.30
bum_Latn	116.16	153.66	121.83	121.74	101.82
bzj_Latn	115.75	175.63	128.70	135.59	93.15
cab_Latn	164.07	215.31	172.22	174.35	123.20
cac_Latn	169.42	231.73	176.29	175.63	116.03
cak_Latn	185.42	246.76	193.62	191.54	123.65
caq_Latn	128.13	174.12	141.21	138.17	95.54
cat_Latn	54.93	118.69	44.29	45.98	76.47
cbk_Latn	103.50	154.23	105.08	108.15	91.19
cce_Latn	124.20	159.68	133.40	132.89	106.02
ceb_Latn	99.37	146.70	113.69	132.72	94.43
ces_Latn	62.40	133.26	101.82	114.59	86.91
cfm_Latn	138.07	179.26	142.58	143.43	107.20
che_Cyrl	152.68	188.52	146.76	148.42	126.87
chk_Latn	128.34	180.14	133.81	134.01	97.76
chv_Cyrl	132.89	166.12	138.37	128.58	91.96
ckb_Arab	126.47	155.90	125.59	164.22	100.65
cmn_Hani	63.67	121.22	51.49	60.91	76.95
cnh_Latn	129.26	175.83	134.65	139.53	104.21
crh_Cyrl	128.56	166.14	128.91	139.13	82.61
crs_Latn	100.72	139.95	101.88	57.70	80.86
csy_Latn	125.81	172.44	138.22	132.16	100.90
ctd_Latn	120.85	163.07	128.99	125.52	92.79
ctu_Latn	156.04	220.78	162.45	157.63	112.41
cuk_Latn	151.95	213.08	159.59	156.10	119.01
cym_Latn	110.34	165.10	135.91	147.72	103.89

Table 16: Detailed results of *NLL* on PBC (Part I).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
dan_Latn	63.65	114.43	97.95	101.13	86.68
deu_Latn	57.09	109.69	84.08	54.00	80.90
djk_Latn	143.19	192.80	147.13	153.74	120.66
dln_Latn	113.43	155.19	118.82	125.73	92.37
dtp_Latn	158.44	222.01	165.46	169.63	111.77
dyu_Latn	122.24	161.61	126.53	132.73	103.04
dzo_Tibt	157.37	550.44	162.42	36.35	99.22
efi_Latn	121.73	173.61	139.58	136.78	90.15
ell_Grek	80.65	169.16	109.07	57.11	105.74
eng_Latn	28.40	93.81	40.01	42.56	46.91
enm_Latn	45.43	113.74	62.99	66.87	55.22
epo_Latn	79.83	125.27	88.81	100.79	85.24
est_Latn	93.49	128.66	109.45	45.04	99.10
eus_Latn	133.89	145.19	101.06	78.92	150.43
ewe_Latn	140.69	190.49	147.85	133.36	103.15
fao_Latn	101.92	150.02	113.16	134.84	93.21
fas_Arab	87.19	121.18	99.32	104.15	77.85
fij_Latn	110.29	158.90	130.18	97.89	97.65
fil_Latn	74.66	130.09	106.51	109.03	84.32
fin_Latn	68.42	125.52	116.35	38.52	91.75
fon_Latn	160.80	210.76	176.23	178.40	107.16
fra_Latn	46.01	105.73	38.57	44.16	73.33
fry_Latn	111.69	146.88	111.45	123.28	100.32
gaa_Latn	128.54	165.53	145.88	107.90	100.68
gil_Latn	125.22	171.28	131.71	130.68	106.24
giz_Latn	131.75	183.07	145.21	143.35	97.84
gkn_Latn	151.99	210.57	167.40	166.78	116.75
gkp_Latn	159.33	219.00	168.31	166.05	110.30
gla_Latn	102.90	174.06	129.10	138.42	100.51
gle_Latn	102.09	161.80	132.57	146.14	116.86
glv_Latn	122.94	172.06	126.34	134.37	98.35
gom_Latn	149.35	199.59	155.54	159.44	129.81
gor_Latn	156.67	215.11	170.13	167.89	115.02
grc_Grek	64.91	153.70	93.39	68.67	81.49
guc_Latn	193.75	271.60	202.31	190.66	138.75
gug_Latn	139.06	183.84	146.45	151.28	114.14
guj_Gujr	121.18	329.05	86.23	202.19	107.88
gur_Latn	143.42	208.51	152.80	148.18	106.41
guw_Latn	142.60	155.00	158.22	166.16	98.92
gya_Latn	130.25	197.61	146.23	137.31	99.85
gym_Latn	180.93	262.58	196.74	161.03	135.73
hat_Latn	112.20	159.68	116.00	48.45	90.71
hau_Latn	105.95	146.45	117.21	127.18	96.63
haw_Latn	91.42	140.04	102.87	102.50	91.03
heb_Hebr	86.85	197.96	113.81	125.21	143.56
hif_Latn	104.78	161.10	114.69	116.63	107.93
hil_Latn	103.93	151.84	112.82	130.28	90.13
hin_Deva	87.35	175.19	62.49	63.21	103.09
hin_Latn	102.01	144.04	112.84	112.96	109.68
hmo_Latn	119.64	179.32	128.46	103.09	91.86
hne_Deva	124.72	183.69	106.59	120.10	94.27
hnj_Latn	126.88	186.09	144.08	149.87	89.64
hra_Latn	116.66	151.27	122.49	122.14	96.72
hrv_Latn	62.52	125.68	96.82	107.18	73.96
hui_Latn	151.46	203.46	161.05	161.36	108.54
hun_Latn	69.17	118.92	117.60	125.55	94.04
hus_Latn	170.91	241.76	179.70	177.42	120.81
hye_Armn	111.94	219.94	141.97	171.24	89.75
iba_Latn	102.40	135.32	109.00	102.90	87.43
ibo_Latn	131.16	189.15	130.12	172.79	112.01
ifa_Latn	140.53	194.86	151.53	148.37	102.38
ifb_Latn	149.93	198.42	157.12	156.49	107.60
ikk_Latn	132.84	186.95	150.31	163.16	95.14
ilo_Latn	119.72	162.55	127.85	146.58	102.18
ind_Latn	66.39	121.78	58.14	58.36	80.77
isl_Latn	92.39	137.42	113.54	123.83	94.12
ita_Latn	54.57	116.50	73.53	52.57	78.23

Table 17: Detailed results of *NLL* on PBC (Part II).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
ium_Latn	150.62	222.39	155.20	157.52	99.54
ixl_Latn	190.07	299.20	206.08	202.92	127.52
izz_Latn	167.28	228.45	195.19	198.57	118.78
jam_Latn	119.85	181.93	134.52	139.07	96.42
jav_Latn	134.11	171.16	136.06	140.01	109.34
jpn_Jpan	67.67	114.11	84.64	61.57	88.53
kaa_Cyrl	136.14	179.48	138.63	153.33	84.79
kaa_Latn	134.02	172.76	135.14	145.18	99.15
kab_Latn	137.81	193.54	129.87	159.45	117.96
kac_Latn	141.33	187.59	150.24	163.68	110.99
kal_Latn	120.90	143.71	134.44	90.38	109.58
kan_Knda	128.60	336.06	93.77	210.99	110.09
kat_Geor	103.81	132.04	144.43	155.32	93.39
kaz_Cyrl	129.49	166.60	137.43	150.12	108.56
kbp_Latn	151.83	205.24	166.76	156.21	105.09
kek_Latn	161.79	230.77	168.62	155.46	110.43
khm_Khmr	141.48	453.97	161.21	233.38	100.53
kia_Latn	122.81	171.17	136.22	131.73	95.76
kik_Latn	141.34	189.92	143.91	155.69	106.53
kin_Latn	110.75	137.92	101.14	123.88	99.96
kir_Cyrl	125.74	148.16	127.29	148.79	94.02
kjb_Latn	152.31	205.47	156.49	160.88	109.02
kjh_Cyrl	133.84	168.82	142.31	145.53	97.43
kmm_Latn	137.88	185.46	149.16	145.91	107.81
kmr_Cyrl	139.23	182.66	137.99	142.19	103.56
kmr_Latn	120.54	149.31	124.74	136.78	96.93
knv_Latn	249.77	346.55	266.68	245.87	135.66
kor_Hang	66.58	119.14	92.53	42.28	82.45
kpg_Latn	128.18	190.92	139.68	135.05	90.65
krc_Cyrl	123.42	149.60	119.82	130.97	89.22
kri_Latn	118.15	172.62	134.67	131.33	96.69
ksd_Latn	108.75	155.22	117.82	116.91	84.44
kss_Latn	248.46	385.70	269.58	224.81	174.70
ksw_Mymr	145.34	187.44	155.38	107.97	94.71
kua_Latn	118.00	142.31	125.97	104.16	99.83
lam_Latn	145.51	199.78	154.91	139.07	115.48
lao_Laoo	163.17	414.25	172.28	234.46	116.39
lat_Latn	56.98	102.85	65.60	73.77	73.15
lav_Latn	90.61	119.37	103.75	114.03	94.92
ldi_Latn	118.61	161.07	122.03	124.26	112.27
leh_Latn	131.72	169.51	140.67	124.57	104.47
lhu_Latn	147.32	262.73	152.94	153.96	100.83
lin_Latn	113.81	128.30	110.20	123.56	92.52
lit_Latn	92.16	120.15	107.63	123.52	97.69
loz_Latn	119.93	140.69	125.00	98.71	99.46
ltz_Latn	114.62	156.92	114.49	104.79	96.89
lug_Latn	117.59	174.83	115.12	143.99	107.58
luo_Latn	118.37	158.54	129.88	126.61	108.96
lus_Latn	122.17	159.02	125.37	133.65	103.21
lzh_Hani	62.06	88.07	54.92	60.19	66.36
mad_Latn	136.26	192.90	146.43	145.94	103.63
mah_Latn	113.96	159.42	120.91	110.27	97.45
mai_Deva	136.92	209.91	108.91	126.23	100.39
mal_Mlym	111.12	210.81	72.27	126.62	105.00
mam_Latn	173.35	227.62	181.33	179.63	138.57
mar_Deva	105.80	184.52	83.30	141.12	106.37
mau_Latn	139.06	259.48	153.06	140.49	148.96
mbb_Latn	160.77	237.84	174.36	171.35	101.96
mck_Latn	124.95	161.95	131.37	123.87	99.72
mcn_Latn	110.95	153.55	120.44	123.48	96.39
mco_Latn	203.59	285.23	205.92	192.68	159.16
mdy_Ethi	164.72	284.41	157.66	188.38	92.89
meu_Latn	111.26	152.92	120.09	103.47	91.50
mfe_Latn	99.68	136.00	98.60	55.39	80.86
mgh_Latn	131.75	181.11	140.22	136.00	118.72
mgr_Latn	126.60	154.99	129.40	106.55	108.42

Table 18: Detailed results of *NLL* on PBC (Part III).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
mhr_Cyrl	122.42	160.48	119.36	127.42	100.09
min_Latn	139.41	194.79	136.22	138.87	133.30
miq_Latn	129.28	182.98	144.36	141.12	104.92
mkd_Cyrl	85.29	151.22	112.67	89.21	89.46
mlg_Latn	107.66	135.73	106.88	128.75	86.60
mlt_Latn	108.58	168.92	134.24	137.26	107.12
mos_Latn	129.97	161.07	135.30	138.61	112.98
mps_Latn	196.29	283.21	212.92	204.15	126.56
mri_Latn	87.56	138.21	103.82	111.33	88.68
mrw_Latn	127.39	174.88	134.59	133.06	99.21
msa_Latn	104.71	152.69	97.60	93.04	113.32
mwm_Latn	159.30	238.34	171.46	159.27	99.80
mxv_Latn	146.98	235.76	162.84	164.53	126.52
mya_Mymr	162.62	248.51	185.69	84.78	107.92
myv_Cyrl	148.95	192.16	140.65	152.76	110.76
mzh_Latn	146.28	217.81	160.03	153.09	101.97
nan_Latn	130.85	204.44	144.08	138.29	118.30
naq_Latn	126.47	179.33	139.25	135.80	100.90
nav_Latn	167.01	233.91	176.25	183.89	119.97
nbl_Latn	109.14	148.07	114.06	127.75	96.55
nch_Latn	155.09	212.74	165.40	171.74	144.74
ncj_Latn	131.14	184.55	137.38	140.86	129.63
ndc_Latn	106.50	151.16	111.70	117.94	107.15
nde_Latn	106.83	152.97	114.79	133.97	100.83
ndo_Latn	132.12	162.83	138.17	107.11	105.82
nds_Latn	123.29	166.55	124.21	125.62	123.87
nep_Deva	109.47	199.05	81.70	141.10	103.11
ngu_Latn	148.78	204.17	156.73	156.92	120.15
nia_Latn	135.60	192.37	143.95	130.11	111.86
nld_Latn	58.81	114.31	96.47	97.28	82.78
nmf_Latn	122.39	165.17	130.43	134.30	98.07
nnb_Latn	122.26	163.28	127.40	131.83	98.36
nno_Latn	80.33	133.43	102.92	112.53	86.17
nob_Latn	61.45	126.38	100.25	98.89	80.02
nor_Latn	56.27	104.11	87.94	86.18	71.86
npi_Deva	115.63	219.43	78.62	159.24	96.97
nse_Latn	116.86	157.47	124.34	116.64	109.55
nso_Latn	116.55	160.63	114.34	132.40	97.49
nya_Latn	112.30	160.76	116.85	124.27	101.20
nyn_Latn	120.67	159.71	127.46	131.05	106.34
nyy_Latn	153.10	189.04	164.69	160.66	121.06
nzi_Latn	130.01	179.62	150.60	141.28	101.29
ori_Orya	148.25	392.96	91.33	296.43	98.06
ory_Orya	143.02	352.28	95.95	282.70	106.99
oss_Cyrl	140.22	182.75	141.83	139.80	97.09
ote_Latn	160.20	247.13	175.42	168.28	119.40
pag_Latn	123.49	163.26	131.05	133.56	109.90
pam_Latn	117.54	163.02	121.69	130.95	103.78
pan_Guru	130.07	286.36	90.80	208.67	106.44
pap_Latn	110.09	149.82	118.80	114.73	92.08
pau_Latn	125.22	178.67	132.62	131.85	104.80
pcm_Latn	76.80	127.05	89.92	91.28	79.44
pdt_Latn	124.83	175.03	129.63	126.61	97.29
pes_Arab	91.68	129.42	105.39	105.84	84.63
pis_Latn	118.50	180.76	130.26	133.14	95.70
pls_Latn	147.97	217.00	152.42	153.90	104.29
plt_Latn	113.52	139.96	112.22	139.93	89.18
poh_Latn	240.95	363.61	257.65	256.21	140.88
pol_Latn	61.88	111.24	97.90	107.87	85.46
pon_Latn	123.40	164.68	131.48	125.12	105.87
por_Latn	53.69	106.83	42.29	45.85	75.88
prk_Latn	118.66	167.68	121.37	128.55	94.48
prs_Arab	88.26	123.81	99.63	105.09	80.28
pxm_Latn	154.30	207.27	160.81	160.27	102.81
qub_Latn	133.85	172.77	139.98	107.69	93.49
quc_Latn	176.66	222.18	191.60	178.35	124.30

Table 19: Detailed results of *NLL* on PBC (Part IV).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
qug_Latn	124.11	158.66	131.62	95.23	95.07
quh_Latn	148.83	174.81	154.34	107.04	106.24
quw_Latn	104.78	139.63	109.91	95.69	92.58
quy_Latn	119.84	140.49	127.93	85.16	94.14
quz_Latn	126.18	149.08	134.60	85.68	96.32
qvi_Latn	134.03	177.51	139.73	114.64	100.81
rap_Latn	139.27	239.23	152.39	157.18	100.81
rar_Latn	136.30	205.48	152.87	149.88	123.36
rmy_Latn	124.05	164.59	129.35	132.97	108.84
ron_Latn	71.75	145.55	113.22	136.42	92.16
rop_Latn	141.24	218.11	152.37	163.59	93.46
rug_Latn	144.21	200.64	155.21	151.73	99.72
run_Latn	111.11	140.11	101.95	120.78	99.61
rus_Cyrl	57.09	115.06	85.44	48.93	78.66
sag_Latn	118.57	144.91	123.32	113.66	101.74
sah_Cyrl	140.83	175.34	139.86	155.36	99.78
san_Deva	120.11	183.77	128.71	131.38	123.45
san_Latn	133.78	188.35	151.17	152.84	112.82
sba_Latn	147.44	205.90	167.36	154.66	98.05
seh_Latn	116.71	159.73	123.08	121.65	100.51
sin_Sinh	133.79	283.43	166.13	228.72	113.72
slk_Latn	75.89	141.01	105.13	123.74	89.45
slv_Latn	75.67	140.40	111.88	127.31	95.15
sme_Latn	134.17	166.51	131.28	132.85	103.75
smo_Latn	113.64	165.04	126.68	127.57	96.65
sna_Latn	107.03	157.69	112.48	124.30	99.14
snd_Arab	141.08	183.48	144.96	173.65	107.47
som_Latn	114.80	163.60	131.06	149.83	110.34
sop_Latn	120.92	148.69	129.81	113.62	117.37
sot_Latn	112.14	155.55	113.59	127.04	95.35
spa_Latn	49.64	107.41	43.22	48.95	69.30
sqi_Latn	106.17	145.44	116.56	140.41	92.13
srm_Latn	172.30	242.13	187.65	185.42	124.54
srn_Latn	112.06	137.43	113.65	121.74	91.24
srp_Cyrl	57.16	129.53	97.17	99.06	71.36
srp_Latn	61.53	124.70	95.02	105.00	71.54
ssw_Latn	120.48	172.73	132.69	140.25	104.17
sun_Latn	123.92	165.15	124.81	129.90	111.93
suz_Deva	141.06	222.01	143.66	139.17	93.18
swe_Latn	60.78	124.59	105.53	99.90	86.99
swh_Latn	97.92	131.52	87.87	54.27	90.87
sxn_Latn	173.04	249.27	188.33	183.25	124.96
tam_Taml	109.64	213.05	70.91	64.81	100.45
tat_Cyrl	136.52	167.63	136.15	147.39	94.42
tbz_Latn	135.12	176.55	145.64	137.50	88.42
tca_Latn	202.44	294.68	215.39	207.66	112.33
tdt_Latn	114.70	164.66	123.50	129.26	93.86
tel_Telu	122.18	196.41	91.03	65.40	115.17
teo_Latn	115.64	157.71	122.24	117.99	99.95
tgk_Cyrl	128.86	144.47	127.10	140.25	101.04
tgl_Latn	74.71	130.27	109.14	110.56	85.29
tha_Thai	107.69	187.16	134.02	58.96	101.09
tih_Latn	129.95	188.97	139.91	137.24	89.82
tir_Ethi	122.75	258.00	143.76	190.93	99.03
tlh_Latn	87.59	142.90	97.02	97.38	62.55
tob_Latn	179.90	269.36	189.99	191.60	107.12
toh_Latn	127.60	171.62	136.60	136.06	104.79
toi_Latn	124.75	166.04	133.32	114.25	114.54
toj_Latn	175.52	237.75	181.56	177.72	148.72
ton_Latn	120.61	179.89	125.92	137.67	98.37
top_Latn	165.19	223.46	174.82	173.24	164.19
tpi_Latn	105.47	161.38	117.02	128.35	84.22
tpm_Latn	120.52	166.93	131.95	129.07	89.91
tsn_Latn	112.13	163.36	113.76	129.32	96.63
tso_Latn	125.25	155.16	120.37	134.66	103.51
tsz_Latn	129.96	184.88	142.29	126.39	110.66
tuc_Latn	187.91	261.93	196.20	166.43	106.46

Table 20: Detailed results of *NLL* on PBC (Part V).

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MALA-500
tui_Latn	135.41	187.71	146.21	146.43	107.57
tuk_Cyrl	127.20	168.02	136.69	145.86	94.72
tuk_Latn	123.72	144.22	124.67	135.21	97.42
tum_Latn	127.49	165.45	130.05	109.19	102.35
tur_Latn	76.97	118.11	102.96	57.95	99.22
twi_Latn	110.21	159.12	110.54	122.43	93.81
tyv_Cyrl	165.82	197.37	164.25	181.87	107.33
tzh_Latn	147.06	205.37	157.00	148.16	118.46
tzo_Latn	166.45	248.81	178.03	173.52	122.42
udm_Cyrl	138.00	176.90	140.39	137.21	102.56
uig_Arab	166.57	226.61	157.03	229.43	114.04
uig_Latn	145.11	165.68	156.02	157.78	121.77
ukr_Cyrl	68.45	134.95	101.40	93.65	92.95
urd_Arab	99.74	141.13	74.20	63.53	110.49
uzb_Cyrl	128.48	149.94	136.85	135.60	88.90
uzb_Latn	118.83	138.99	132.96	145.00	95.19
uzn_Cyrl	136.04	160.00	145.95	142.34	94.12
ven_Latn	131.18	172.05	138.68	137.88	104.82
vie_Latn	74.42	115.85	56.37	59.30	91.10
wal_Latn	129.99	180.12	134.43	122.12	105.68
war_Latn	111.26	159.23	118.85	131.06	113.74
wbm_Latn	120.96	174.48	126.11	128.99	94.78
wol_Latn	115.67	154.99	101.97	127.09	102.93
xav_Latn	243.35	430.83	263.49	257.10	137.76
xho_Latn	112.96	155.63	109.11	135.39	107.43
yan_Latn	125.81	179.37	136.34	131.31	98.47
yao_Latn	143.68	187.96	148.92	143.16	114.54
yap_Latn	150.28	207.86	157.67	157.91	123.07
yom_Latn	118.61	155.58	121.64	126.20	100.18
yor_Latn	129.77	166.68	100.87	155.88	105.79
yua_Latn	148.34	218.12	155.65	156.40	118.30
yue_Hani	64.57	122.47	54.42	62.71	87.78
zai_Latn	121.90	161.31	121.61	129.12	108.99
zho_Hani	64.02	115.19	51.79	63.22	69.53
zlm_Latn	57.83	101.11	48.96	51.87	64.76
zom_Latn	119.86	159.31	128.06	125.99	98.96
zsm_Latn	60.40	110.60	51.75	52.51	70.43
zul_Latn	103.20	157.55	113.26	130.35	98.06
all	122.10	180.54	129.55	131.31	101.67

Table 21: Detailed results of *NLL* on PBC (Part VI).

Lang	LLaMA 2 7B	mGPT 13B	BLOOM 7B1	XGLM 7.5B	MaLA-500 1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot	9-shot	10-shot
ace_Latn	44.12	47.55	50.00	36.76	34.31	52.94	60.29	60.78	65.69	67.65	64.22	65.20	68.63	71.57
acm_Arab	52.45	65.69	69.12	58.33	32.35	53.43	59.31	63.73	63.73	67.16	66.67	69.12	66.67	66.67
afr_Latn	68.14	55.39	53.92	40.20	41.18	62.25	65.69	69.12	71.08	74.02	73.53	74.51	76.96	78.92
ajp_Arab	47.55	64.22	68.63	53.43	33.33	56.86	59.80	59.80	65.20	63.73	63.24	69.12	68.14	66.67
als_Latn	41.67	46.57	45.59	28.43	27.94	51.96	63.73	62.25	69.12	71.08	69.61	72.06	75.98	77.45
amh_Ethi	15.69	18.63	16.67	13.24	25.00	36.76	45.59	51.47	51.96	53.92	51.96	53.43	54.90	53.92
apc_Arab	46.57	65.69	68.14	53.43	31.37	55.88	60.29	65.69	65.69	67.16	65.20	68.63	67.65	72.06
arb_Arab	53.43	63.24	68.14	57.35	32.35	54.90	60.29	63.73	65.20	68.14	67.16	69.12	68.63	70.59
ary_Arab	45.10	57.84	69.12	50.49	26.47	52.45	55.39	56.37	60.29	59.80	64.22	63.73	59.31	64.71
arz_Arab	50.98	64.22	68.14	56.86	30.88	52.45	59.31	60.78	64.22	66.18	68.14	69.12	66.67	69.12
asm_Beng	17.16	49.02	61.27	37.25	31.37	53.43	58.82	65.20	67.65	67.65	68.14	67.65	67.65	67.65
ast_Latn	69.12	60.78	69.12	55.39	34.31	65.69	70.10	70.59	74.02	75.00	75.98	77.94	79.90	79.90
ayr_Latn	25.00	26.96	32.35	19.61	20.10	29.41	38.24	38.73	38.24	43.14	40.20	43.14	44.61	42.16
azb_Arab	25.49	41.67	32.84	24.51	25.98	41.18	45.59	45.10	46.57	49.51	50.00	49.02	50.49	49.51
azj_Latn	34.80	64.22	37.25	32.84	30.88	57.84	64.71	68.63	64.22	72.55	69.61	70.59	74.02	72.55
bak_Cyrl	38.73	61.27	32.35	32.35	34.80	51.47	60.29	61.27	69.12	68.63	68.14	68.63	73.53	70.10
bam_Latn	25.49	24.51	29.41	20.10	22.55	25.98	34.80	42.16	43.14	44.12	45.10	42.16	46.08	44.12
ban_Latn	58.82	51.47	58.82	43.14	28.92	55.39	63.24	65.69	66.67	72.06	72.06	72.55	72.06	71.57
bel_Cyrl	47.55	59.80	28.92	30.39	40.69	60.29	63.24	66.18	67.65	70.10	72.55	72.06	72.55	73.04
bem_Latn	31.37	28.92	38.24	25.49	21.08	34.80	43.14	48.04	50.49	50.49	53.43	53.43	52.45	53.92
ben_Beng	25.49	61.27	64.22	52.45	31.37	54.90	63.24	62.25	67.65	70.10	70.10	69.12	66.18	68.63
bjn_Latn	48.53	51.96	61.76	42.65	32.35	62.75	66.18	68.14	71.57	75.98	73.04	72.55	75.00	77.45
bod_Tibt	15.20	12.75	15.69	15.69	22.06	34.80	37.75	37.75	38.73	39.71	39.71	39.71	41.67	44.12
bos_Latn	65.20	64.71	45.59	33.82	37.75	65.20	72.06	70.10	71.57	75.98	75.00	76.47	76.96	77.45
bul_Cyrl	66.18	63.24	38.73	52.94	45.10	62.25	67.65	67.16	68.14	75.00	71.57	72.06	73.53	75.00
cat_Latn	71.08	60.78	66.67	60.29	34.31	59.31	68.14	68.63	71.57	72.55	69.12	73.04	76.96	76.47
ceb_Latn	50.49	50.98	49.02	39.71	39.22	60.29	66.67	66.67	68.63	73.04	72.06	71.57	74.51	74.02
ces_Latn	69.12	62.75	47.55	40.69	39.22	62.25	69.61	70.59	72.55	76.47	72.55	74.02	80.88	76.96
cjk_Latn	27.94	30.39	34.31	26.47	22.55	30.88	31.86	32.84	38.24	38.24	38.24	35.78	39.22	42.65
ckb_Arab	19.61	22.55	23.04	12.25	28.92	53.43	60.29	57.35	65.20	65.20	62.75	65.69	65.69	70.59
cmn_Hani	73.04	65.20	67.65	54.90	39.71	69.12	74.02	72.06	76.47	77.45	76.47	75.98	79.41	76.47
crh_Latn	38.24	56.37	40.20	36.76	29.41	51.96	62.25	61.76	64.22	69.12	64.22	70.10	69.12	71.08
cym_Latn	39.22	28.43	34.80	21.57	28.43	55.39	62.75	63.73	66.18	72.06	74.02	72.06	75.00	77.45
dan_Latn	69.12	64.22	55.39	44.12	38.24	54.41	63.24	65.20	70.10	71.08	72.06	71.57	73.53	74.51
deu_Latn	74.02	60.29	61.27	55.39	41.18	63.73	68.63	71.57	69.12	75.00	75.49	76.47	77.45	77.45
dyu_Latn	28.43	28.92	32.35	20.10	21.08	29.90	38.73	39.71	46.57	44.12	41.67	47.06	44.61	43.63
dzo_Tibt	14.71	10.29	13.73	12.75	21.57	30.39	36.76	37.25	39.71	36.76	39.22	37.75	43.14	39.22
ell_Grek	47.55	63.73	28.43	60.29	43.63	62.75	69.61	66.67	69.12	69.61	70.59	73.04	72.06	72.06
eng_Latn	71.57	59.80	71.08	67.65	48.04	63.24	70.59	69.12	69.12	74.02	73.04	74.51	76.96	75.98
epo_Latn	52.94	50.49	52.94	43.63	27.94	49.51	66.18	66.67	68.63	72.55	73.53	71.57	74.51	75.98
est_Latn	48.04	54.90	41.18	57.35	29.41	55.88	62.75	66.67	70.10	72.06	70.10	69.12	71.57	73.04
eus_Latn	36.27	59.80	64.22	55.88	27.94	52.45	61.76	66.18	66.18	73.53	73.53	72.55	73.53	75.49
ewe_Latn	23.53	23.53	29.90	17.16	23.53	28.43	38.24	35.29	41.18	43.63	38.73	44.61	39.71	43.63
fao_Latn	41.18	43.63	38.73	29.90	35.29	52.94	56.86	57.84	62.25	61.27	62.25	61.76	64.22	69.12
fij_Latn	27.45	27.45	36.27	24.02	24.51	37.75	48.53	47.55	53.92	48.53	50.00	51.47	51.47	53.92
fin_Latn	67.65	63.24	38.73	56.86	36.76	61.76	70.10	70.10	72.55	74.51	72.06	73.53	75.00	75.49
fon_Latn	25.49	22.06	31.37	19.12	23.53	29.90	30.88	37.25	35.78	38.24	39.71	38.24	37.25	46.57
fra_Latn	72.06	64.71	66.18	59.80	36.76	58.82	71.08	67.65	71.08	74.51	71.57	74.02	77.45	77.45
ful_Latn	27.45	31.37	32.84	24.02	21.57	31.86	36.76	40.69	43.14	45.10	44.12	47.06	47.06	46.08
fur_Latn	58.82	50.00	55.88	38.73	35.78	53.92	55.88	62.25	66.67	68.63	68.14	67.16	75.00	72.55
gla_Latn	37.75	24.51	27.45	17.16	24.51	50.98	55.88	57.84	57.35	59.80	63.24	61.27	62.75	65.20
gle_Latn	39.71	25.49	25.00	17.16	27.94	53.43	57.84	63.24	64.22	67.65	64.71	63.24	62.75	73.53
glg_Latn	68.63	62.25	64.22	55.39	29.41	66.67	70.10	71.08	74.02	76.96	73.53	76.47	77.45	80.39
grn_Latn	42.16	47.06	48.53	33.82	25.98	52.45	62.75	64.71	62.25	67.16	64.22	65.69	61.76	69.61
guj_Gujr	15.20	09.31	62.25	12.75	28.43	50.98	54.90	60.29	63.73	63.73	64.22	65.69	62.75	67.16
hat_Latn	41.67	38.73	42.16	45.10	34.80	57.35	63.73	62.25	65.20	72.06	69.61	70.10	73.53	73.04
hau_Latn	25.49	28.43	29.90	20.59	28.43	47.06	55.39	57.84	61.27	65.69	62.75	65.69	66.18	65.20
heb_Hebr	37.75	63.24	20.59	11.76	26.47	41.67	47.06	51.47	54.90	51.96	50.98	54.90	53.92	54.41
hin_Deva	44.61	62.75	62.75	51.96	33.82	55.39	60.78	66.67	65.20	69.61	70.59	74.02	73.04	72.06
hne_Deva	37.75	58.82	59.80	49.02	28.43	55.39	55.88	65.20	62.75	68.63	66.18	65.69	68.14	71.08
hrv_Latn	66.18	65.20	44.12	36.27	40.69	63.73	73.04	71.08	73.53	76.47	72.06	74.51	78.43	78.43
hun_Latn	71.08	63.24	41.67	27.94	30.88	60.78	67.16	70.59	68.63	75.49	73.53	74.02	73.04	76.47
hye_Armn	20.59	17.16	13.73	12.75	32.84	58.82	59.80	67.16	65.20	69.61	68.14	69.12	69.12	72.55
ibo_Latn	24.02	26.47	38.24	19.12	30.39	51.47	57.35	63.73	67.16	69.12	68.14	69.12	68.63	72.06
ilo_Latn	45.10	45.59	48.04	32.35	27.45	54.90	61.76	61.76	68.14	68.14	70.10	73.04	73.53	70.10
ind_Latn	74.02	62.75	70.10	54.90	40.69	62.75	68.63	71.57	70.59	75.49	75.49	76.96	80.39	77.94
isl_Latn	35.29	36.76	28.92	24.51	38.73	55.88	60.78	58.33	60.29	64.71	63.73	63.24	62.75	65.20
ita_Latn	69.61	62.25	62.75	57.84	40.20	64.22	70.59	70.59	74.51	77.94	75.98	76.96	80.39	76.96
jav_Latn	50.49	52.94	55.39	38.24	31.86	53.43	60.78	64.22	65.20	72.55	69.12	73.04	68.63	73.04
jpn_Jpan	73.53	60.29	63.24	55.88	38.73	67.16	72.06	75.49	78.92	79.41	80.39	78.92	81.86	81.37
kab_Latn	16.18	16.67	20.10	12.25	20.59	24.02	22.55	30.39	31.86	34.80	28.43	33.33	32.35	34.31
kac_Latn	25.98	24.51	28.43	20.59	20.10	24.02	29.90	35.78	35.78	43.14	37.75	37.25	43.63	39.71
kam_Latn	26.96	34.31	34.80	26.47	22.06	36.76	38.73	37.75	40.69	41.67	46.57	41.67	42.16	42.16
kan_Knda	17.16	11.27	61.27	11.27	25.49	50.98	57.35	60.29	61.27	65.20	63.24	64.22	65.69	67.16
kat_Geor	29.41	61.27	18.14	14.71	32.84	56.86	60.78	62.25	65.20	67.65	70.59	70.59	70.10	74.51
kaz_Cyrl	37.75	62.75	29.90	28.43	34.31	53.43	57.35	62.25	65.69	67.16	65.20	65.69	69.12	67.65
kbp_Latn	24.51	22.06	30.39	16.18	21.08	28.43	36.76	38.73	40.69	39.22	40.69	39.71	38.73	40.20
kea_Latn	53.43	51.96	56.86	39.71	32.84	56.86	63.73	65.20	67.16	69.61	69.12	71.57	71.57	72.06
khm_Khmr	27.45	11.27	25.49	15.20	39.22	61.76	67.16	67.65	68.63	72.06	73.53	75.00	76.47	76.47
kik_Latn	29.41	32.84	38.73	26.96	21.57	37.75	49.51	50.49	50.49	56.86	56.37	56.37	52.94	56.86
kin_Latn	26.47	32.35	50.49	24.51	27.45	40.69	49.02	52.94	57.84	58.33	60.78	56.86	59.31	59.80
kir_Cyrl	35.78	60.78	34.80	27.45	29.90	45.59	58.33	60.78	60.29	65.20	60.29	64.71	66.18	66.18
kmb_Latn	26.47	28.43	33.82	25.00	21.08	31.86	35.29	39.71	38.24	41.18	41.67	37.25	41.67	44.61
kmr_Latn	29.41	33.82	33.33	21.57	25.98	37.75	47.06	47.55	52.45	52.45	54.90	54.41	58.82	61.76
kon_Latn	33.33	33.82	40.69	32.35	22.06	39.71	46.57	51.96	53.92	64.71	64.22	60.78	64.22	65.20
kor_Hang	67.65	63.24	43.14	56.37	45.10	63.24	67.65	69.12	71.57	73.04	70.59	75.98	76.96	76.47
lao_Laoo	24.02	14.22	26.47	16.67	39.71	55.39	62.25	63.73	68.63	70.59	70.10	68.63	70.59	70.59
lij_Latn	55.88	53.43	56.37	44.61	37.25	58.82	67.65	66.67	69.61	71.57	71.08	74.02	76.47	74.02

Table 22: Detailed results on SIB200 (Part I). For previous LLMs, 3-shot results are presented.

Lang	LLaMA 2 7B	mGPT 13B	BLOOM 7B1	XGLM 7.5B	MaLA-500 1-shot	2-shot	3-shot	4-shot	5-shot	6-shot	7-shot	8-shot	9-shot	10-shot
lim_Latn	60.78	50.00	50.00	33.82	32.84	56.37	60.78	63.24	66.67	67.16	69.12	72.55	70.59	72.06
lin_Latn	36.76	40.20	43.14	34.31	23.04	38.24	47.06	53.92	57.84	61.27	56.86	60.29	62.75	65.69
lit_Latn	40.20	60.29	41.18	30.39	32.35	55.39	62.75	65.20	64.71	70.59	68.14	66.67	69.61	73.04
lmo_Latn	57.84	50.98	55.88	41.67	34.80	59.31	65.69	66.18	70.10	71.57	70.59	70.59	75.00	75.49
ltz_Latn	55.88	47.06	52.94	39.22	39.22	56.37	65.20	61.27	70.59	68.14	71.08	70.59	71.08	74.02
lua_Latn	32.35	33.33	39.22	28.43	20.10	33.82	40.20	42.65	49.02	51.96	50.00	50.00	49.51	50.00
lug_Latn	27.94	25.00	33.82	19.61	22.06	35.78	40.20	43.63	48.04	51.47	47.06	43.63	49.02	49.02
luo_Latn	28.43	28.43	32.84	25.49	21.57	31.37	37.25	42.65	48.04	49.51	46.57	51.47	49.51	51.47
lus_Latn	43.63	42.16	49.02	31.37	25.49	45.59	51.96	53.92	52.45	54.90	57.84	60.29	58.33	60.29
lvs_Latn	43.14	67.16	43.63	29.41	31.37	57.84	65.20	63.24	68.14	72.55	67.65	69.61	71.08	72.55
mai_Deva	40.69	59.31	60.29	51.47	33.33	57.84	61.76	66.67	67.65	69.12	69.12	70.10	71.57	69.12
mal_Mlym	20.10	60.29	64.71	13.24	25.98	52.45	59.31	60.29	62.75	62.25	65.69	63.24	63.73	68.14
mar_Deva	29.90	56.86	63.73	37.75	36.27	51.96	57.35	64.22	63.73	63.73	63.73	66.67	66.18	68.14
min_Latn	48.04	55.39	59.80	39.71	31.37	57.35	69.12	68.14	68.63	77.94	72.06	75.98	75.49	77.45
mkd_Cyrl	60.78	52.45	32.84	44.12	44.12	66.18	68.63	69.12	68.63	73.04	73.04	72.55	73.04	76.96
mlt_Latn	49.51	45.10	46.08	29.90	35.78	64.71	67.16	68.14	67.16	77.45	75.49	76.96	77.45	76.96
mon_Cyrl	23.53	54.90	20.10	18.63	38.24	50.00	56.86	55.88	63.24	64.22	63.24	63.24	65.20	67.16
mos_Latn	25.49	23.53	29.90	20.59	20.59	27.94	36.76	37.75	37.75	40.69	41.67	45.10	41.67	45.10
mri_Latn	30.39	24.02	30.88	17.65	28.43	44.12	49.02	51.47	51.47	57.84	55.88	58.82	56.37	58.33
mya_Mymr	19.12	60.29	19.61	60.29	23.53	38.73	43.14	53.43	53.43	50.98	52.45	54.90	51.96	54.90
nld_Latn	70.10	59.80	55.88	46.08	45.59	64.71	69.12	68.63	73.04	73.53	75.49	74.02	79.41	80.88
nno_Latn	64.71	61.76	52.45	45.59	35.29	52.94	64.22	62.75	66.18	68.63	68.63	70.10	69.12	73.04
npi_Deva	39.22	51.96	64.71	40.69	33.82	57.84	61.76	68.14	67.65	67.65	68.63	70.10	68.63	75.49
nso_Latn	27.94	30.88	33.33	22.55	21.08	33.82	43.14	46.08	49.02	52.94	51.96	53.92	54.90	53.92
nya_Latn	32.35	34.31	40.69	27.94	23.04	35.29	45.59	49.02	50.98	51.47	52.94	53.92	52.94	58.33
oci_Latn	68.63	56.37	65.69	48.53	34.31	60.29	69.12	65.20	67.65	73.04	73.53	71.57	75.49	76.47
orm_Latn	17.16	18.14	22.06	16.67	20.10	30.39	35.29	41.18	41.67	47.55	41.67	44.12	43.14	51.47
ory_Orya	13.24	13.73	64.22	11.76	24.51	45.10	52.45	57.84	53.92	61.27	57.84	56.86	60.78	60.78
pag_Latn	52.45	49.51	53.92	40.20	31.86	54.90	62.75	60.78	67.65	64.71	70.10	70.10	69.12	69.61
pan_Guru	14.22	11.27	62.25	11.76	33.82	54.90	58.82	63.73	64.22	67.65	67.16	66.67	68.63	67.16
pap_Latn	55.39	50.00	52.94	38.24	30.39	56.86	64.71	66.67	69.61	74.51	69.12	73.53	70.59	75.49
pes_Arab	47.06	58.82	52.94	32.84	39.22	61.27	71.08	63.73	70.59	72.55	72.55	73.53	76.47	76.47
plt_Latn	28.43	32.84	37.25	21.57	29.41	51.96	58.82	57.84	59.31	60.78	60.29	60.29	64.22	60.29
pol_Latn	74.51	60.78	47.06	32.84	36.76	61.76	68.63	69.12	71.08	75.00	74.02	74.02	77.45	75.98
por_Latn	70.10	61.76	65.20	59.31	36.76	64.71	72.06	70.10	74.51	75.00	76.96	75.49	78.43	82.84
prs_Arab	50.49	55.39	49.51	33.33	37.25	60.78	64.22	67.16	69.12	72.55	72.55	73.53	72.55	75.49
pus_Arab	30.39	34.80	38.73	21.08	30.39	47.06	50.98	52.45	54.41	53.92	53.92	55.88	55.88	57.84
quy_Latn	32.84	35.29	40.69	35.29	22.06	36.27	44.12	45.59	49.02	52.45	49.51	49.02	50.98	50.98
ron_Latn	69.12	61.76	57.84	42.65	41.18	61.27	70.10	65.20	70.10	74.51	73.53	75.00	78.92	78.43
run_Latn	25.49	27.94	44.12	25.49	23.53	37.25	46.57	50.49	51.96	59.31	51.96	56.37	57.84	60.29
rus_Cyrl	71.57	63.24	53.43	60.29	38.73	64.22	65.20	69.12	72.06	75.98	75.00	76.47	75.49	78.92
sag_Latn	29.90	27.94	31.37	21.08	20.59	30.88	43.63	47.06	48.53	55.88	52.45	54.41	55.88	58.82
san_Deva	27.94	47.55	54.90	42.65	24.51	48.04	60.29	57.84	62.25	66.67	65.20	61.76	66.18	65.20
scn_Latn	51.96	50.00	53.43	40.69	37.25	63.73	73.04	70.59	74.02	77.45	75.49	75.00	80.39	76.47
sin_Sinh	15.20	10.78	20.10	12.75	29.90	56.37	60.29	65.20	66.18	68.14	64.71	66.67	63.73	67.16
slk_Latn	68.14	60.29	47.55	39.71	34.31	58.33	68.63	66.67	70.59	75.00	70.59	71.57	74.51	75.00
slv_Latn	68.14	60.78	44.12	32.84	38.73	63.24	68.14	68.14	70.59	73.53	73.53	74.51	78.43	76.47
smo_Latn	30.39	25.00	31.86	18.14	29.41	52.45	60.29	62.25	62.25	65.69	67.16	65.20	66.18	69.61
sna_Latn	28.43	29.41	36.27	23.53	24.51	39.71	44.61	45.59	44.61	49.51	45.59	47.55	47.06	50.00
snd_Arab	27.94	37.25	39.22	23.53	27.94	42.65	47.06	50.49	52.45	54.41	54.41	52.94	55.88	56.86
som_Latn	23.53	25.49	27.94	17.16	22.06	36.27	44.61	47.55	51.47	52.94	52.94	53.92	54.41	55.39
sot_Latn	29.41	28.43	33.82	18.63	22.55	36.76	43.14	47.06	50.49	51.96	52.45	55.39	54.41	56.86
spa_Latn	72.55	58.33	67.65	56.37	35.29	64.22	69.61	72.06	74.51	74.02	72.06	76.47	78.43	78.43
srd_Latn	53.92	52.45	50.98	37.25	31.37	60.29	66.18	68.63	75.98	74.02	77.94	77.45	79.41	79.41
srp_Cyrl	63.73	55.39	33.33	39.22	45.59	65.20	70.59	69.61	73.04	76.47	74.02	75.00	77.94	79.41
ssw_Latn	29.41	25.00	31.37	21.57	24.02	44.12	46.57	50.00	52.94	51.96	53.92	56.86	53.92	60.78
sun_Latn	55.39	59.31	63.73	44.61	37.25	60.29	68.63	70.10	71.08	73.53	72.55	73.53	75.00	75.98
swe_Latn	71.08	61.27	52.94	48.04	33.82	53.43	60.29	64.71	64.71	69.12	70.10	69.61	72.55	70.59
swh_Latn	32.35	63.24	61.27	56.86	29.41	50.49	59.31	58.82	62.75	60.29	62.25	68.63	66.67	66.67
szl_Latn	56.86	50.49	45.59	29.41	30.88	51.47	59.80	63.73	64.71	67.16	69.61	68.63	71.08	69.12
tam_Taml	20.59	63.24	67.16	58.82	30.88	50.49	55.88	62.75	63.24	63.73	65.20	69.61	66.67	68.63
tat_Cyrl	37.75	60.29	35.29	28.92	33.33	54.90	64.22	64.71	65.69	74.51	70.10	71.57	71.57	73.53
tel_Telu	18.14	60.78	61.27	59.80	25.98	50.00	52.45	59.80	58.82	63.73	60.78	60.29	63.73	61.76
tgk_Cyrl	26.96	57.84	23.53	17.16	36.76	54.90	60.29	60.78	61.27	69.12	64.71	66.18	68.14	70.59
tgl_Latn	55.88	58.33	49.02	40.20	43.14	64.22	69.61	64.71	70.10	75.49	74.51	77.45	78.43	77.45
tha_Thai	44.61	60.78	23.53	57.35	41.18	63.24	67.16	68.63	70.10	72.06	70.59	70.10	72.06	75.49
tir_Ethi	13.24	16.18	16.18	13.73	21.57	34.80	39.22	41.67	47.06	46.08	45.59	47.06	45.59	47.55
tpi_Latn	63.24	46.57	56.86	33.33	31.86	58.82	65.69	68.14	70.10	72.06	74.51	73.53	75.98	74.51
tsn_Latn	28.92	29.90	32.35	24.51	24.51	40.20	44.12	47.06	47.06	53.43	51.96	50.00	50.49	54.41
tso_Latn	30.88	31.37	36.76	28.92	22.55	35.29	41.18	45.10	46.08	48.04	43.14	43.14	45.59	49.02
tuk_Latn	34.31	46.08	39.22	27.45	24.02	45.59	53.43	58.82	57.84	63.73	63.73	66.18	65.69	66.18
tum_Latn	26.96	34.31	33.82	27.94	21.57	39.22	43.14	45.59	46.08	47.55	44.61	49.02	49.51	49.51
tur_Latn	52.94	62.75	40.20	52.94	36.76	60.78	68.63	70.10	72.06	74.02	75.00	76.47	75.98	76.96
uig_Arab	18.63	18.14	20.10	11.27	21.08	33.33	36.76	39.71	43.14	44.61	43.14	48.53	47.55	48.04
ukr_Cyrl	71.57	63.73	41.18	43.63	39.71	60.29	65.69	66.18	69.12	71.08	75.00	73.53	72.55	75.00
umb_Latn	25.00	26.47	29.90	23.04	21.57	30.88	32.84	36.76	35.78	40.20	38.24	34.80	36.76	35.29
urd_Arab	38.73	53.43	63.24	54.41	36.27	55.39	62.75	64.22	64.22	68.63	65.20	68.63	67.16	67.16
uzb_Latn	30.39	62.75	35.78	23.53	22.06	50.49	56.37	57.84	63.24	72.06	63.73	69.12	72.55	71.57
vec_Latn	65.69	59.80	56.86	52.45	39.22	62.25	66.18	69.61	70.10	69.12	74.51	75.98	75.00	76.47
vie_Latn	67.65	63.24	67.16	60.78	39.71	60.78	67.16	68.63	74.51	75.98	76.47	75.00	78.43	79.90
war_Latn	51.47	51.47	51.47	37.25	37.75	61.27	65.69	65.20	69.61	73.04	71.57	71.57	74.02	74.02
wol_Latn	32.35	34.80	43.14	25.49	23.53	36.76	42.16	45.59	48.53	53.43	47.55	52.94	54.90	53.43
xho_Latn	30.39	29.90	38.24	22.55	25.98	46.57	51.96	56.37	58.33	60.78	61.76	60.29	62.75	64.22
yid_Hebr	23.04	22.06	16.18	12.25	24.02	34.80	39.22	39.71	40.20	46.57	41.18	40.69	44.61	45.10
yor_Latn	21.57	29.41	47.55	21.57	26.47	32.35	41.18	39.22	41.67	48.04	42.16	43.14	44.61	43.14
yue_Hani	75.00	64.71	67.16	55.88	40.69	69.12	71.57	76.47	76.96	81.37	77.94	79.41	81.86	79.41
zsm_Latn	65.69	61.27	64.71	50.00	36.76	60.29	68.14	69.12	67.65	73.53	73.53	76.96	77.45	75.00
zul_Latn	25.00	25.49	35.29	15.20	25.49	51.47	49.02	54.41	56.37	57.84	60.29	61.76	59.31	62.75
all	42.08	45.34	44.63	34.36	30.88	50.71	57.02	58.95	61.20	64.04	63.15	64.13	65.19	66.32

Table 23: Detailed results on SIB200 (Part II). For previous LLMs, 3-shot results are presented.

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MaLA-500
ace_Latn	46.85	47.75	49.55	41.44	48.65
ach_Latn	45.05	37.84	41.44	40.54	36.04
acr_Latn	47.75	51.35	50.45	47.75	50.45
afr_Latn	54.05	38.74	51.35	49.55	55.86
agw_Latn	48.65	45.05	41.44	42.34	49.55
ahk_Latn	43.24	36.04	36.04	35.14	45.05
aka_Latn	42.34	32.43	38.74	42.34	54.95
aln_Latn	34.23	35.14	36.94	35.14	44.14
als_Latn	38.74	36.94	42.34	42.34	47.75
alt_Cyrl	44.14	44.14	45.05	51.35	51.35
alz_Latn	36.94	35.14	31.53	28.83	37.84
aoj_Latn	50.93	37.96	45.37	46.3	49.07
arb_Arab	43.24	45.05	49.55	44.14	50.45
arn_Latn	38.74	42.34	34.23	36.04	43.24
ary_Arab	32.43	33.33	38.74	32.43	44.14
arz_Arab	31.53	39.64	45.05	36.94	46.85
asm_Beng	45.95	42.34	54.95	40.54	54.05
ayr_Latn	47.75	37.84	44.14	44.14	54.05
azb_Arab	39.64	40.54	47.75	45.05	47.75
aze_Latn	45.05	46.85	45.05	43.24	49.55
bak_Cyrl	45.05	52.25	49.55	56.76	56.76
bam_Latn	42.34	37.84	49.55	39.64	47.75
ban_Latn	36.04	41.44	34.23	34.23	42.34
bar_Latn	49.55	46.85	44.14	48.65	53.15
bba_Latn	45.05	32.43	45.95	46.85	46.85
bci_Latn	36.94	35.14	36.94	33.33	44.14
bcl_Latn	42.34	48.65	39.64	39.64	54.95
bel_Cyrl	47.75	45.95	48.65	43.24	57.66
bem_Latn	47.75	37.84	42.34	41.44	51.35
ben_Beng	40.54	41.44	52.25	51.35	47.75
bhw_Latn	37.84	43.24	41.44	46.85	47.75
bim_Latn	38.74	39.64	33.33	36.94	45.05
bis_Latn	44.14	49.55	44.14	39.64	48.65
bqc_Latn	39.64	36.04	34.23	33.33	40.54
bre_Latn	39.64	36.04	35.14	36.04	40.54
btx_Latn	49.55	36.94	42.34	41.44	43.24
bul_Cyrl	45.05	42.34	48.65	45.05	54.95
bum_Latn	42.34	39.64	37.84	37.84	44.14
bzj_Latn	53.15	46.85	47.75	50.45	52.25
cab_Latn	39.64	38.74	37.84	36.94	36.04
cac_Latn	43.24	37.84	40.54	38.74	45.05
cak_Latn	45.95	35.14	44.14	40.54	50.45
caq_Latn	39.64	38.74	38.74	44.14	37.84
cat_Latn	52.25	45.05	46.85	48.65	52.25
cbk_Latn	54.05	40.54	56.76	54.05	55.86
cce_Latn	49.55	45.05	50.45	48.65	48.65
ceb_Latn	44.14	42.34	48.65	45.05	51.35
ces_Latn	44.14	43.24	45.05	46.85	51.35
cfm_Latn	49.55	41.44	49.55	53.15	48.65
che_Cyrl	37.84	33.33	36.94	37.84	38.74
chk_Latn	45.05	41.44	41.44	36.04	45.95
chv_Cyrl	43.24	45.05	45.05	49.55	58.56
ckb_Arab	44.14	36.94	45.05	42.34	51.35
cmn_Hani	48.65	45.05	53.15	48.65	53.15
cnh_Latn	46.85	46.85	46.85	49.55	46.85
crh_Cyrl	49.55	40.54	47.75	54.95	54.05
crs_Latn	52.25	44.14	49.55	55.86	59.46
csy_Latn	47.75	41.44	54.95	53.15	45.95
ctd_Latn	50.45	48.65	56.76	53.15	56.76
ctu_Latn	41.44	35.14	38.74	40.54	43.24
cuk_Latn	42.34	42.34	38.74	39.64	37.84
cym_Latn	39.64	38.74	39.64	43.24	41.44
dan_Latn	53.15	41.44	39.64	38.74	54.95
deu_Latn	45.05	36.04	37.84	38.74	43.24
djk_Latn	42.34	35.14	42.34	46.85	40.54
dln_Latn	48.65	40.54	51.35	54.05	47.75

Table 24: Detailed results on Taxi1500 (Part I). 3-shot results are presented.

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MaLA-500
dtp_Latn	39.64	35.14	42.34	46.85	50.45
dyu_Latn	41.44	39.64	42.34	38.74	46.85
dzo_Tibt	45.05	40.54	41.44	45.05	45.05
efi_Latn	39.64	36.04	38.74	41.44	45.95
ell_Grek	49.55	45.95	49.55	48.65	51.35
eng_Latn	55.86	42.34	58.56	54.05	59.46
enm_Latn	50.45	41.44	56.76	50.45	54.95
epo_Latn	49.55	40.54	47.75	42.34	49.55
est_Latn	46.85	42.34	38.74	52.25	46.85
eus_Latn	38.74	36.04	36.94	39.64	39.64
ewe_Latn	51.35	43.24	50.45	46.85	45.05
fao_Latn	53.15	44.14	52.25	53.15	58.56
fas_Arab	49.55	50.45	57.66	51.35	55.86
fij_Latn	48.65	43.24	41.44	43.24	53.15
fil_Latn	48.65	41.44	46.85	51.35	51.35
fin_Latn	47.75	45.05	41.44	45.95	54.95
fon_Latn	38.74	35.14	37.84	40.54	45.05
fra_Latn	60.36	51.35	62.16	52.25	59.46
fry_Latn	37.84	33.33	36.04	27.03	46.85
gaa_Latn	41.44	33.33	37.84	35.14	40.54
gil_Latn	36.7	31.19	41.28	32.11	41.28
giz_Latn	46.85	44.14	43.24	38.74	45.05
gkn_Latn	38.74	34.23	34.23	36.94	41.44
gkp_Latn	30.63	33.33	41.44	29.73	48.65
gla_Latn	33.33	39.64	44.14	45.05	49.55
gle_Latn	33.33	35.14	36.04	34.23	39.64
glv_Latn	43.24	41.44	37.84	38.74	42.34
gom_Latn	34.23	31.53	33.33	40.54	42.34
gor_Latn	43.24	34.23	43.24	40.54	46.85
guc_Latn	44.14	36.04	37.84	41.44	45.05
gug_Latn	45.05	44.14	42.34	41.44	50.45
guj_Gujr	45.95	37.84	52.25	44.14	56.76
gur_Latn	45.95	45.95	44.14	47.75	48.65
guw_Latn	45.05	37.84	47.75	46.85	48.65
gya_Latn	37.84	37.84	41.44	34.23	42.34
gym_Latn	41.44	39.64	39.64	43.24	50.45
hat_Latn	50.45	43.24	44.14	41.44	56.76
hau_Latn	44.14	37.84	41.44	44.14	48.65
haw_Latn	45.95	39.64	38.74	34.23	49.55
heb_Hebr	38.74	35.14	34.23	36.94	44.14
hif_Latn	42.34	43.24	49.55	47.75	48.65
hil_Latn	49.55	41.44	40.54	36.94	54.95
hin_Deva	51.35	50.45	49.55	46.85	56.76
hmo_Latn	46.85	45.05	46.85	45.05	53.15
hne_Deva	55.86	54.05	54.05	58.56	58.56
hnj_Latn	48.65	45.05	53.15	51.35	60.36
hra_Latn	49.55	41.44	43.24	46.85	45.95
hrv_Latn	55.86	51.35	52.25	54.95	61.26
hui_Latn	51.35	40.54	41.44	45.05	46.85
hun_Latn	46.85	44.14	41.44	43.24	48.65
hus_Latn	32.43	32.43	34.23	37.84	42.34
hye_Armn	45.95	40.54	45.95	49.55	61.26
iba_Latn	49.55	46.85	51.35	48.65	55.86
ibo_Latn	38.74	33.33	43.24	38.74	44.14
ifa_Latn	36.04	30.63	35.14	38.74	43.24
ifb_Latn	34.23	35.14	39.64	34.23	53.15
ikk_Latn	43.24	36.94	39.64	39.64	43.24
ilo_Latn	39.64	36.04	41.44	37.84	43.24
ind_Latn	49.55	50.45	53.15	53.15	54.95
isl_Latn	48.65	44.14	43.24	48.65	54.05
ita_Latn	50.45	49.55	56.76	58.56	54.95
ium_Latn	45.95	44.14	49.55	50.45	45.05
ixl_Latn	42.34	39.64	41.44	43.24	40.54
izz_Latn	38.74	47.75	39.64	42.34	54.95
jam_Latn	41.44	43.24	53.15	50.45	61.26
jav_Latn	41.44	47.75	44.14	37.84	45.95

Table 25: Detailed results on Taxi1500 (Part II). 3-shot results are presented.

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MaLA-500
jpn_Jpan	46.85	46.85	47.75	50.45	51.35
kaa_Latn	43.24	53.15	47.75	51.35	54.05
kab_Latn	27.93	36.04	30.63	34.23	35.14
kac_Latn	44.14	34.23	43.24	42.34	52.25
kal_Latn	41.44	37.84	36.04	35.14	40.54
kan_Knda	48.65	37.84	52.25	45.95	54.05
kat_Geor	41.44	41.44	42.34	46.85	48.65
kaz_Cyrl	49.55	45.05	51.35	53.15	55.86
kbp_Latn	40.54	35.14	36.94	31.53	47.75
kek_Latn	45.95	42.34	45.05	44.14	51.35
khm_Khmr	52.25	38.74	48.65	49.55	64.86
kia_Latn	36.94	36.04	40.54	41.44	48.65
kik_Latn	45.05	43.24	45.05	44.14	50.45
kin_Latn	42.34	37.84	41.44	38.74	50.45
kir_Cyrl	51.35	46.85	47.75	63.06	64.86
kjb_Latn	48.65	46.85	44.14	44.14	48.65
kjh_Cyrl	44.14	41.44	45.05	41.44	45.95
kmm_Latn	45.95	45.05	47.75	51.35	45.95
kmr_Cyrl	39.64	35.14	45.05	42.34	43.24
knv_Latn	44.55	44.55	45.45	42.73	44.55
kor_Hang	48.65	48.65	49.55	51.35	62.16
kpg_Latn	44.14	52.25	51.35	42.34	54.95
krc_Cyrl	45.95	36.04	48.65	48.65	53.15
kri_Latn	49.55	48.65	49.55	51.35	54.95
ksd_Latn	36.94	33.33	40.54	33.33	49.55
kss_Latn	32.43	28.83	34.23	29.73	47.75
ksw_Mymr	44.14	45.95	42.34	37.84	52.25
kua_Latn	41.44	42.34	36.94	35.14	40.54
lam_Latn	43.24	36.94	45.95	43.24	40.54
lao_Laoo	45.05	39.64	46.85	50.45	50.45
lat_Latn	53.15	41.44	53.15	56.76	57.66
lav_Latn	39.64	33.33	36.04	39.64	45.05
ldi_Latn	35.14	32.43	36.94	34.23	36.04
leh_Latn	47.75	37.84	33.33	32.43	41.44
lhu_Latn	27.93	34.23	34.23	37.84	42.34
lin_Latn	47.75	37.84	39.64	39.64	48.65
lit_Latn	42.34	40.54	44.14	48.65	49.55
loz_Latn	45.95	42.34	36.04	44.14	40.54
ltz_Latn	46.85	45.95	47.75	41.44	49.55
lug_Latn	40.54	32.43	39.64	38.74	45.95
luo_Latn	40.54	36.94	34.23	38.74	40.54
lus_Latn	39.64	40.54	42.34	41.44	50.45
lzh_Hani	54.95	48.65	54.05	43.24	56.76
mad_Latn	47.75	52.25	47.75	47.75	53.15
mah_Latn	43.24	36.04	42.34	45.95	45.05
mai_Deva	45.05	41.44	49.55	54.05	51.35
mam_Latn	43.24	33.33	41.44	45.05	45.95
mar_Deva	49.55	44.14	53.15	45.95	56.76
mau_Latn	29.73	29.73	36.94	37.84	32.43
mbb_Latn	44.14	42.34	38.74	39.64	49.55
mck_Latn	40.54	34.23	36.04	39.64	49.55
mcn_Latn	35.14	27.93	33.33	33.33	38.74
mco_Latn	41.44	33.33	43.24	33.33	43.24
mdy_Ethi	39.64	46.85	43.24	43.24	51.35
meu_Latn	53.15	38.74	45.05	48.65	52.25
mfe_Latn	51.35	48.65	52.25	50.45	56.76
mgh_Latn	42.34	33.33	41.44	35.14	38.74
mgr_Latn	39.64	34.23	33.33	41.44	38.74
mhr_Cyrl	47.27	42.73	45.45	42.73	48.18
min_Latn	37.84	45.95	53.15	45.05	53.15
miq_Latn	51.35	46.85	43.24	54.95	49.55
mkd_Cyrl	52.25	48.65	56.76	57.66	66.67
mlg_Latn	35.14	36.04	36.94	37.84	45.95
mlt_Latn	37.84	33.33	42.34	43.24	46.85
mos_Latn	39.64	42.34	39.64	36.04	36.04
mps_Latn	47.75	45.05	42.34	45.05	51.35
mri_Latn	45.05	42.34	38.74	42.34	44.14

Table 26: Detailed results on Taxi1500 (Part III). 3-shot results are presented.

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MaLA-500
mrw_Latn	40.54	39.64	41.44	37.84	49.55
msa_Latn	44.14	41.44	45.95	37.84	46.85
mwm_Latn	36.94	31.53	39.64	38.74	47.75
mxv_Latn	33.33	35.14	39.64	38.74	40.54
mya_Mymr	45.05	48.65	44.14	44.14	46.85
myv_Cyrl	39.64	43.24	40.54	41.44	45.05
mzh_Latn	45.05	45.95	42.34	40.54	44.14
nan_Latn	32.43	35.14	48.65	49.55	44.14
naq_Latn	36.94	36.94	37.84	39.64	41.44
nav_Latn	27.03	28.83	30.63	33.33	38.74
nbl_Latn	21.62	18.02	21.62	25.23	27.93
nch_Latn	37.84	34.23	33.33	40.54	40.54
ncj_Latn	46.85	45.95	42.34	41.44	42.34
ndc_Latn	44.14	36.04	43.24	36.94	49.55
nde_Latn	33.33	29.73	33.33	36.04	41.44
ndo_Latn	41.28	34.86	37.61	33.94	46.79
nds_Latn	41.44	38.74	37.84	34.23	43.24
nep_Deva	45.05	49.55	63.06	51.35	60.36
ngu_Latn	47.75	39.64	43.24	42.34	49.55
nld_Latn	47.75	39.64	47.75	43.24	56.76
nmf_Latn	44.14	40.54	42.34	41.44	44.14
nnb_Latn	45.05	42.34	36.94	44.14	40.54
nno_Latn	56.76	46.85	45.95	52.25	54.95
nob_Latn	52.25	41.44	44.14	45.95	56.76
nor_Latn	50.45	35.14	46.85	47.75	53.15
npi_Deva	51.35	54.95	55.86	45.95	54.95
nse_Latn	38.74	28.83	39.64	38.74	42.34
nso_Latn	45.05	43.24	45.05	45.05	50.45
nya_Latn	48.65	39.64	44.14	42.34	54.95
nyn_Latn	39.64	33.33	37.84	36.94	45.05
nyy_Latn	43.24	42.34	43.24	40.54	47.75
nzi_Latn	36.94	32.43	33.33	32.43	35.14
ori_Orya	43.24	34.23	51.35	46.85	45.95
ory_Orya	44.14	44.14	49.55	46.85	55.86
oss_Cyrl	49.55	49.55	49.55	44.14	54.05
ote_Latn	34.23	31.53	34.23	36.04	49.55
pag_Latn	44.14	48.65	48.65	42.34	50.45
pam_Latn	45.95	36.04	44.14	47.75	45.05
pan_Guru	41.44	33.33	46.85	40.54	47.75
pap_Latn	50.45	44.14	52.25	49.55	53.15
pau_Latn	38.74	45.05	37.84	36.94	46.85
pcm_Latn	58.56	47.75	56.76	53.15	57.66
pdt_Latn	53.15	45.95	45.95	48.65	54.05
pes_Arab	50.91	46.36	59.09	48.18	53.64
pis_Latn	57.66	47.75	50.45	45.95	55.86
pls_Latn	43.24	43.24	43.24	39.64	45.95
plt_Latn	36.94	35.14	37.84	43.24	47.75
poh_Latn	42.34	42.34	45.05	39.64	48.65
pol_Latn	41.44	43.24	46.85	55.86	56.76
pon_Latn	45.95	39.64	43.24	39.64	42.34
por_Latn	56.76	54.95	56.76	54.05	58.56
prk_Latn	44.14	43.24	49.55	40.54	46.85
prs_Arab	50.45	51.35	55.86	56.76	57.66
pxm_Latn	48.65	44.14	41.44	41.44	47.75
qub_Latn	46.85	44.14	43.24	48.65	45.05
quc_Latn	45.05	41.44	43.24	38.74	50.45
qug_Latn	45.95	46.85	50.45	45.05	56.76
quh_Latn	49.55	49.55	46.85	42.34	51.35
quw_Latn	43.24	36.94	45.05	44.14	53.15
quy_Latn	58.56	48.65	54.95	50.45	57.66
quz_Latn	51.35	38.74	60.36	54.95	59.46
qvi_Latn	46.79	46.79	49.54	45.87	47.71
rap_Latn	43.24	35.14	41.44	39.64	46.85
rar_Latn	40.54	32.43	31.53	29.73	45.95
rmy_Latn	37.84	37.84	38.74	40.54	43.24
ron_Latn	45.05	51.35	44.14	47.75	57.66

Table 27: Detailed results on Taxi1500 (Part IV). 3-shot results are presented.

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MaLA-500
rop_Latn	45.95	45.05	42.34	42.34	55.86
rug_Latn	43.24	38.74	46.85	44.14	45.05
run_Latn	46.85	40.54	45.05	40.54	52.25
rus_Cyrl	49.55	41.44	50.45	47.75	53.15
sag_Latn	43.24	43.24	41.44	40.54	47.75
sah_Cyrl	40.54	35.14	44.14	44.14	54.95
sba_Latn	42.34	43.24	45.05	40.54	49.55
seh_Latn	45.05	35.14	40.54	42.34	45.95
sin_Sinh	39.64	38.74	39.64	42.34	45.95
slk_Latn	53.15	50.45	44.14	47.75	53.15
slv_Latn	47.75	45.05	55.86	51.35	49.55
sme_Latn	45.95	45.05	42.34	41.44	48.65
smo_Latn	38.74	40.54	43.24	44.14	53.15
sna_Latn	50.45	30.63	43.24	45.95	60.36
snd_Arab	44.14	45.05	56.76	51.35	56.76
som_Latn	33.33	36.94	35.14	34.23	39.64
sop_Latn	40.54	34.23	40.54	35.14	35.14
sot_Latn	47.75	41.44	40.54	43.24	49.55
spa_Latn	51.35	49.55	51.35	51.35	56.76
sqi_Latn	42.34	43.24	52.25	52.25	57.66
srm_Latn	35.14	41.44	39.64	37.84	45.05
srn_Latn	45.95	53.15	54.05	48.65	51.35
srp_Latn	59.46	48.65	58.56	54.05	58.56
ssw_Latn	38.74	45.05	36.94	40.54	48.65
sun_Latn	43.24	40.54	45.05	44.14	48.65
suz_Deva	46.85	42.34	42.34	43.24	49.55
swe_Latn	58.56	48.65	53.15	54.95	61.26
swh_Latn	46.85	49.55	49.55	48.65	56.76
sxn_Latn	42.34	36.94	44.14	44.14	46.85
tam_Taml	44.14	53.15	59.46	48.65	60.36
tat_Cyrl	47.75	47.75	45.95	48.65	54.05
tbz_Latn	36.04	35.14	34.23	35.14	42.34
tca_Latn	39.64	40.54	43.24	41.44	45.05
tdt_Latn	40.54	38.74	48.65	45.05	52.25
tel_Telu	33.33	45.95	50.45	45.95	49.55
teo_Latn	33.33	37.84	26.13	31.53	41.44
tgk_Cyrl	42.34	44.14	48.65	49.55	57.66
tgl_Latn	48.65	41.44	46.85	51.35	51.35
tha_Thai	43.24	42.34	43.24	37.84	47.75
tih_Latn	43.24	37.84	40.54	36.04	54.05
tir_Ethi	29.73	36.94	27.93	34.23	41.44
tlh_Latn	51.35	45.95	45.95	41.44	53.15
tob_Latn	44.55	43.64	41.82	38.18	50.00
toh_Latn	42.34	39.64	40.54	40.54	42.34
toi_Latn	44.14	45.05	34.23	36.04	45.05
toj_Latn	43.24	40.54	36.94	43.24	42.34
ton_Latn	42.34	42.34	42.34	44.14	52.25
top_Latn	46.85	34.23	37.84	38.74	36.94
tpi_Latn	48.65	44.14	52.25	48.65	49.55
tpm_Latn	37.84	41.44	38.74	32.43	42.34
tsn_Latn	40.54	36.04	38.74	34.23	37.84
tsz_Latn	37.84	32.43	37.84	38.74	46.85
tuc_Latn	45.95	44.14	47.75	44.14	48.65
tui_Latn	42.34	38.74	38.74	37.84	50.45
tuk_Latn	36.04	42.34	45.05	43.24	50.45
tum_Latn	47.75	39.64	46.85	52.25	50.45
tur_Latn	46.79	44.04	40.37	43.12	45.87
twi_Latn	41.44	43.24	41.44	37.84	46.85
tyv_Cyrl	38.74	38.74	43.24	44.14	45.05
tzh_Latn	41.82	36.36	41.82	41.82	38.18
tzo_Latn	39.64	43.24	34.23	29.73	41.44
udm_Cyrl	36.94	38.74	42.34	44.14	47.75
ukr_Cyrl	52.25	48.65	51.35	55.86	53.15

Table 28: Detailed results on Taxi1500 (Part V). 3-shot results are presented.

Lang	LLaMA 2-7B	mGPT-13B	BLOOM-7B1	XGLM-7.5B	MaLA-500
ukr_Cyrl	52.25	48.65	51.35	55.86	53.15
uzb_Latn	45.05	49.55	37.84	46.85	54.05
uzn_Cyrl	45.95	40.54	45.05	45.05	49.55
ven_Latn	45.05	44.14	42.34	41.44	54.05
vie_Latn	53.15	45.95	62.16	45.95	54.95
wal_Latn	35.14	33.33	35.14	35.14	39.64
war_Latn	48.65	39.64	37.84	45.05	54.95
wbm_Latn	48.65	39.64	46.85	46.85	48.65
wol_Latn	36.04	34.23	32.43	34.23	36.94
xav_Latn	50.45	33.33	46.85	44.14	45.95
xho_Latn	43.24	37.84	40.54	39.64	46.85
yan_Latn	45.05	46.85	52.25	41.44	53.15
yao_Latn	42.34	41.44	43.24	44.14	48.65
yap_Latn	38.74	40.54	35.14	32.43	41.44
yom_Latn	35.14	31.53	33.33	25.23	36.94
yor_Latn	41.44	38.74	39.64	44.14	47.75
yua_Latn	41.44	32.43	43.24	41.44	36.04
yue_Hani	43.24	48.65	53.15	38.74	57.66
zai_Latn	45.05	35.14	40.54	43.24	44.14
zho_Hani	47.75	51.35	51.35	44.14	58.56
zlm_Latn	54.05	49.55	57.66	56.76	64.86
zom_Latn	50.45	42.34	44.14	43.24	48.65
zsm_Latn	58.56	59.46	63.96	55.86	66.67
zul_Latn	46.85	42.34	46.85	46.85	51.35
all	44.07	40.98	43.98	43.24	48.89

Table 29: Detailed results on Taxi1500 (Part VI). 3-shot results are presented.

Chapter 4

EMMA-500: Enhancing Massively Multilingual Adaptation of Large Language Models

EMMA-500: Enhancing Massively Multilingual Adaptation of Large Language Models

Shaoxiong Ji*[1], Zihao Li[1], Indraneil Paul[2], Jaakko Paavola[1], Peiqin Lin[3,5], Pinzhen Chen[4], Dayyán O'Brien[4], Hengyu Luo[1], Hinrich Schütze[3,5], Jörg Tiedemann[1], and Barry Haddow[4]

[1]*University of Helsinki* [2]*Technical University of Darmstadt* [3]*University of Munich*

[4]*University of Edinburgh* [5]*Munich Center for Machine Learning*
{*shaoxiong.ji; zihao.li; jaakko.paavola; hengyu.luo; jorg.tiedemann*}*@helsinki.fi*
indraneil.paul@tu-darmstadt.de; linpq@cis.lmu.de; {*pinzhen.chen; dobrien4; bhaddow*}*@ed.ac.uk*

Abstract

In this work, we introduce **EMMA-500**, a large-scale multilingual language model continue-trained on texts across 546 languages designed for enhanced multilingual performance, with a focus on improving language coverage for low-resource languages. To facilitate continual pre-training, we compile **the MaLA corpus**, a comprehensive multilingual dataset and enrich it with curated datasets across diverse domains. Leveraging this corpus, we conduct extensive continual pre-training of the Llama 2 7B model, resulting in EMMA-500, which demonstrates robust performance across a wide collection of benchmarks, including a comprehensive set of multilingual tasks and **PolyWrite**, an open-ended generation benchmark developed in this study. Our results highlight the effectiveness of continual pre-training in expanding large language models' language capacity, particularly for underrepresented languages, demonstrating significant gains in cross-lingual transfer, task generalization, and language adaptability.

Model: huggingface.co/collections/MaLA-LM -¿ EMMA-500
Data: huggingface.co/collections/MaLA-LM -¿ MaLA corpus
PolyWrite: huggingface.co/datasets/MaLA-LM/PolyWrite
Evaluation: github.com/MaLA-LM/emma-500

1 Introduction

Multilingual language models (MLMs) are designed to process and generate text in multiple languages. These models have evolved rapidly over the past decade, fueled by advances in deep learning, e.g., Transformer networks (Vaswani et al., 2017), pre-training techniques, and the availability of large-scale multilingual corpora such as mC4 (Raffel et al., 2020) and ROOTS (Laurençon et al., 2022). The development of models like BERT (Devlin et al., 2019), GPT, and T5 (Raffel et al., 2020) opened the door for multilingual counterparts such as mBERT, XLM-R (Conneau et al., 2020), mGPT (Shliazhko et al., 2022), and mT5 (Xue et al., 2021). These models were trained on massive multilingual corpora, allowing text in dozens of languages to be processed with the same set of model weights. By increasing the scale of both the data and the model, multilingual language models have demonstrated impressive performance on a wide array of tasks, such as text classification, machine translation, and question answering across different languages. A key advantage of multilingual models is their ability to leverage cross-lingual transfer, where knowledge learned from high-resource languages (like English or Chinese) can be applied to other languages. However, despite the remarkable progress in multilingual models, many low-resource languages—those with limited available data—remain underserved. While massive and extensive corpora are readily available for high-resource languages like English, French, and Spanish, languages such as Xhosa and Inuktitut often have only scarce or

*Corresponding author

fragmented data. This disparity between languages tends to make training datasets imbalanced, and multilingual models trained on such imbalanced datasets tend to prioritize high-resource languages, leaving low-resource languages underrepresented.

Recent studies adopt continual pre-training to enhance the language coverage of large language models on low-resource languages. For example, Glot500 (Imani et al., 2023) and MaLA-500 (Lin et al., 2024) use continual pre-training and vocabulary extension using XLM-R and LLaMA, respectively, on the Glot500-c corpus covering 534 languages. xLLMs-100 (Lai et al., 2024) proceeds to multilingual instruction fine-tuning to improve the multilingual performance of LLaMA and BLOOM models on 100 languages, and Aya model (Üstün et al., 2024) applies continual training to the mT5 model (Xue et al., 2021) using their constructed instruction dataset. LLaMAX (Lu et al., 2024) pushes the envelope by focusing on translation tasks via continual pre-training of LLaMA in over 100 languages.

As the field of MLMs evolves, the role of data becomes increasingly critical in enhancing the performance of the models, particularly when it comes to low-resource languages. To address this need for more and better data, we extend existing work, such as MaLA-500, by expanding the corpus for continual pre-training, coupled with large-scale training methods. We emphasize the creation of a massively multilingual corpus that not only increases the quantity of data but also diversifies the types of texts (e.g., code, books, scientific papers, and instructions). This ensures better language adaptation and broader language coverage, thus improving the representation and performance of multilingual language models, especially for underrepresented languages, ultimately creating more inclusive and versatile language models that cater to a broader linguistic diversity. Our contribution is summarized as follows:

- We compile a massively multilingual corpus named MaLA to facilitate continual training of large language models for enhanced language adaptation across a wide range of linguistic contexts.
- We extend the MaLA corpus by integrating multiple curated datasets, creating a comprehensive and diverse data mix specifically for continual pre-training.
- We perform continual pre-training with the Llama 2 7B model[1] on the multilingual corpus with 546 languages, resulting in the new EMMA-500 model. This model is rigorously evaluated on a diverse set of tasks and benchmarks, including PolyWrite, a novel open-ended generation benchmark developed as part of this work.

The MaLA Corpus Our multilingual corpus features the following characteristics:

- It contains 939 languages, 546 of which have more than 100k tokens and are used for training our EMMA-500 model, and 74 billion (B) whitespace delimited tokens in total.
- It has more than 300 languages with over 1 million whitespace delimited tokens and 546 languages with over 100k tokens.
- It comes with four publicly available versions: (1) a noisy version after only basic pre-processing like extraction and harmonization; (2) a cleaned version after data cleaning; (3) a deduplicated version after approximate and exact deduplication; (4) a split version with train and valid splits.
- Our augmentation to the MaLA corpus includes different types of texts such as code, books, scientific papers and instruction data, leading to a data mix with 100B+ whitespace delimited tokens.

Evaluation Results In a comparison with decoder-only LLMs (Section 4) including Llama 2-based continual pre-trained models, LLMs that are designed to be multilingual, and the latest advanced LLMs, our model achieves strong results:

[1]Choosing Llama 2 allows us to compare our model with many other models derived from it using continual pre-training. We plan to continue training models based on Llama 3/3.1 in the future.

- Out of models with parameter sizes from 4.5B to 13B, our model with 7B parameters has the lowest negative log-likelihood according to an intrinsic evaluation.
- Our model remarkably improves the performance of commonsense reasoning, machine translation, and open-ended generation over Llama 2-based models and multilingual baselines, and outperforms the latest advanced models in many cases.
- Our model improves the performance of text classification and natural language inference, outperforming all Llama 2-based models and LLMs designed to be multilingual.
- While math and machine reading comprehension (MRC) tasks are challenging for the Llama 2 7B model and other multilingual LLMs, our model remarkably enhances the Llama 2 base model. Our model yields improved performance on MRC over the base model but still produces quasi-random results similar to other multilingual baselines.
- We demonstrate that massively multilingual continued pre-training does not necessarily lead to regressions in other areas, such as code generation, if the data mix is carefully curated. Our model surpasses the Llama 2 7B base model's code generation abilities.

2 The MaLA Corpus

The MaLA corpus—MaLA standing for **Ma**ssive **L**anguage **A**daptation—is a diverse and extensive compilation of text data encompassing 939 languages sourced from a wide array of datasets. It is developed for continual training of multilingual large language models. The source datasets the MaLA corpus is compiled from exhibit a wide range of variance in various aspects. Examples of such elements include the data quality, the nature of the text content, how the data sources were organized into directory and file tree structures (if distributed as files rather than through an API) and the naming conventions used therein, the data formats and structures in the files or in-memory objects containing the text data, and the logic by which multilingual texts were aligned. This section introduces the efforts made in data extraction, harmonization, pre-processing, cleaning, and deduplication in order to build the corpus.

2.1 Data Pre-processing

To develop the MaLA corpus for training our language model, we establish a processing workflow consisting of the following key steps: (1) loading and curating identified data sources, (2) extracting and harmonizing both textual and metadata from these diverse sources into a unified format—often with tailored filtering, and (3) performing deduplication and further filtering on the textual data. In step (1), source data are either organized and loaded into memory from a file tree structure or, if available, accessed through an API. In step (2), the output is designed to be `JSON` Lines (`JSONL`) files with extracted text data and other desired content. A `JSONL` file contains multiple JSON records for storing data, each separated by a newline character. We selectively process only data annotated as training or development (validation) data, deliberately excluding test data.

2.1.1 Extraction and Harmonization

As mentioned, there is significant variability in the data quality of the source datasets used for compiling the corpus. Many of the datasets exhibit data quality challenges that would have adverse effects on model training if left unaddressed. These dataset-specific challenges are typically addressed during the pre-processing stage. For example, we identify one issue involving text records consisting solely of date and timestamp information in the dataset for Languages of Russia (Corpora and Tools, n.d.), likely resulting from a web scraper failing to differentiate between these elements and actual text. We address this by implementing a logic in the pre-processing script to detect and exclude such records. This issue is resolved by introducing a rule to identify and discard text containing consecutive repeating words.

Due to the extensive volume of data, exhaustive examination of every source for data quality issues is impractical. Instead, we address issues only as we encounter them in our data exploration

and pre-processing pipeline development efforts. This approach likely leaves some data quality problems undetected, since we do not go through the data systematically.

Despite the need for customized handling of certain dataset-specific idiosyncrasies, the core logic and structure of the pre-processing workflow remain consistent across most datasets. We develop a standardized pre-processing script that can be adapted with minor adjustments to accommodate different datasets.

2.1.2 Language Code Normalization

An essential component of our pre-processing pipeline involved converting language codes to the ISO 639-3 standard. This is crucial for ensuring consistent language identification across the source datasets. We rely on the declared language of each dataset and normalize it to the ISO standard without performing additional language identification at this step. This approach helps maintain uniformity while streamlining the pre-processing workflow.

For monolingual data, we primarily use the PyPI library `iso639-lang`[2] or `langcodes`[3]. While converting language codes to ISO 639-3, we encounter several challenges. One issue is that some languages in ISO 639-3 are divided into multiple subvarieties, but our source data does not specify which subvariety is present. Our solution is to retain the original language code from the dataset, even when it does not conform to the ISO 639-3 standard. Another issue arises when certain languages are merged into other languages in the ISO 639-3 standard. In these cases, we update the language code to reflect the merged language.

Additionally, some language names or codes in the source data—referred to as "original language names" or "original language codes"—are not recognized by the conversion libraries. In some cases, the reason behind this is that the original code in fact represents language families or groups of dialects (e.g., the ISO 639-2 codes "ber" for Berber languages and "bih" for Bihari languages), rather than specific languages. If so, we then retain the original codes, despite their non-compliance with ISO 639-3. In other cases, the original language names are spelled differently from the standard recognized by the libraries. To address this mismatch, we implement a logic to detect and correct misspelled language names during pre-processing. All these "corner cases" require careful attention in the pre-processing stage to ensure correct language code identification.

2.1.3 Writing System Recognition

In addition to normalizing language codes, we also identify the script or writing system used in the text data. We use the `GlotScript` library (Kargaran et al., 2024) to recognize writing systems accordant to the ISO 15924 standard. The process begins by sampling 100 random lines from each dataset (or the full dataset if it contains fewer than 100 lines). If `GlotScript` fails to identify a script from this sample, we attempt identification using just the first line of the sample. If this still does not yield a result, we set the script as "None". It is worth noting that we choose not to classify a dataset into multiple scripts, even when code-mixing (i.e., the use of multiple scripts) is present.

If script identification is unsuccessful after the initial steps, we assume the script matched a previously detected one for that language. In cases where no previous script information exists, we refer to a mapping of languages and their default scripts provided by the Glot500 corpora collection. Through this multi-step process, we are able to determine the script for every dataset without exceptions.

During script identification, we encounter several challenges. One issue is determining an appropriate length for the text chunk used for script recognition. A chunk that is too short could lead to incorrect identification if the text contains quotes or foreign language fragments using a different writing system. Conversely, using a chunk that is too long could result in excessive resource usage, slowing down processing or even causing memory exhaustion. Another consideration is whether to assume that a single file or dataset might contain multiple scripts. Such an assumption would require identifying the script at a more granular level, such as paragraph by paragraph or even sentence by sentence. Alternatively, we could assume that each file or dataset contained only one "main" script. This assumption would allow us to identify the script from a representative sample of the text for the whole file or dataset. We adopt the latter approach, recognizing a single dominant script for each

[2] https://pypi.org/project/iso639-lang/

[3] https://pypi.org/project/langcodes

dataset. The output of this process is a label in the format language_Script, e.g., eng_Latn, where "Language" represents the ISO 639-3 language code and ''Script" represents the ISO 15924 script code.

2.1.4 List of Data Sources

The MaLA corpus harvests a wide range of datasets in multiple domains. Table 19 in the appendix lists the corpora and collections we used as monolingual data sources. The major ones include AfriBERTa (Ogueji et al., 2021), Bloom library (Leong et al., 2022), CC100 (Conneau et al., 2020; Wenzek et al., 2020) CulturaX (Nguyen et al., 2023), CulturaY (Thuat Nguyen and Nguyen, 2024), the Curse of Multilinguality (Chang et al., 2023), Evenki Life (Life, 2014), Glot500 (Imani et al., 2023), GlotSparse (Kargaran et al., 2023), monoHPLT of HPLT v1.2 (de Gibert et al., 2024) from the HPLT project (Aulamo et al., 2023), Indigenous Languages Corpora (EdTeKLA, 2022), Indo4B (Wilie et al., 2020), Lacuna Project (Masakhane, 2023), Languages of Russia (Corpora and Tools, n.d.), MADLAD-400 (Kudugunta et al., 2024), Makerere Radio Speech Corpus (Mukiibi et al., 2022), masakhane-ner1.0 (Ifeoluwa Adelani et al., 2021), MC2 (Zhang et al., 2024a), mC4 (Raffel et al., 2020), multilingual-data-peru (Grupo de Inteligencia Artificial PUCP, n.d.), OSCAR 2301 (OSCAR, 2023) from the OSCAR project [4], Tatoeba challenge monolingual collection (Tiedemann, 2020), The Leipzig Corpora (Goldhahn et al., 2012), Tigrinya Language Modeling (Gaim et al., 2021), Wikipedia 20231101 (Wikimedia Foundation, n.d.) and Wikisource 20231201 (Wikimedia Foundation, n.d.). We exclude high-resource languages in CulturaX, HPLT, MADLAD-400, CC100, mC4, and OSCAR 2301. We exclude Gahuza and Pidgin in the AfriBERTa dataset. We filter out texts that mainly contain a date or timestamp in the Languages of Russia dataset. For Glot500-c, we filter out texts, which may come from train or test tests from datasets for machine translation, such as Flores200, Tatoeba, and mtdata. Despite the translation data being split into source and target languages in the Glot500-c corpus, we decide to filter them to avoid potential data leakage, especially since we use Flores200 as the evaluation benchmark.

2.2 Data Cleaning

Most source data has already undergone data cleaning to different extents. Nonetheless, different cleaning processes have been adopted. We continue to clean the data to ensure consistency and accuracy for monolingual and bilingual texts.

Following the pipeline used by BigScience's pipeline for ROOTS corpus (Laurençon et al., 2022), we further adopt some necessary data cleaning to filter out text samples that might have undesirable quality. We first perform document modification for monolingual texts. The first step is whitespace standardization: all types of whitespace in a document are converted into a single, consistent space character. We split documents by newline characters, tabs, and spaces, strip words, and reconstruct the documents to remove very long words. However, these two steps do not apply to languages without whitespace word delimiters like Chinese, Japanese, Korean, Thai, Lao, Burmese, etc. We also remove words containing certain patterns, e.g., "http" and ".com", which are likely to be links and page source code. We then perform document filtering, including word count filtering, character repetition filtering, word repetition filtering, special characters filtering, stop words filtering, and flag words filtering. We re-identify the languages that are supported by the pre-trained `fastText`-based language identification model (Joulin et al., 2016a,b). For other languages, we assume the language identification of the original data source and language code conversion are reliable.

2.3 Data Deduplication

As we collect data from different sources, we deduplicate the data to remove the overlap between different sources.

MinHash Deduplication For each language's dataset in each writing system, we start by using the MinHashLSH algorithm (Rajaraman and Ullman, 2011) to filter out similar documents. It is a near-deduplication technique that builds on MinHash (Broder, 1997), utilizing multiple hash functions for n-grams and the Jaccard similarity, and incorporates Locality-Sensitive Hashing to

[4] https://oscar-project.org/

Table 1: Data statistics of the MaLA corpus and comparison to other multilingual corpora. The number of documents and tokens is in millions.

Dataset	N Lang	N Lang Counted	N Docs	N Tokens	Avg Tokens/Doc
Glot500-c	534	454	1,815	35,449	19.53
MADLAD	419	414	1,043	645,111	618.51
CulturaX	167	161	2,141	1,029,810	480.99
CC100	116	101	2,557	52,201	20.41
MaLA	939	546	824	74,255	90.12

enhance efficiency. We use the implementation by `text-dedup` repository (Mou et al., 2023), applying 5-grams and a similarity threshold of 0.7 to identify similar documents based on the Jaccard index.

Exact Deduplication We further deploy exact deduplication using the `text-dedup` repository again with precise matching. It takes each document through a hash function, i.e., MD5 (Rivest, 1992) in our choice, and the hash values of all documents are compared to identify duplicates.

2.4 Key Statistics

This section presents the final MaLA corpus obtained after data sourcing, pre-processing, cleaning, and deduplication. Table 1 first shows some basic data statistics and compares them with other multilingual corpora for pre-training language models or language adaptation. Additional statistics are presented in Section B in the appendix. The token counts are based on white-space delimitation, though it might not be accurate for languages like Chinese, Japanese and Korean since the entire clause is counted as one token. Glot500-c (Imani et al., 2023) has 534 languages in total, in which 454 languages are directly distributed on Huggingface [5]. We also omit high-resource languages in the other three datasets, i.e., MADLAD (Kudugunta et al., 2024), CulturaX (Nguyen et al., 2023), and CC100 (Wenzek et al., 2020), as our main focus is continual pre-training for language adaptation. The final MaLA corpus consists of 939 languages, 546 of which have more than 100k tokens and are used for training our EMMA-500 model. Counting languages with more than 1 million tokens, the MaLA corpus and Glot500-c have more than 300, while MADLAD and CulturaX have 200 and 100 respectively. Compared with Glot500-c, the MaLA corpus contains documents with significantly higher sequence lengths, with an average token count of 90 versus 19. This higher sequence length is advantageous for continually training LLMs because it provides more context within each training example, allowing the model to better capture long-range dependencies and patterns in the data. As a result, MaLA is more effective for language adaptation.

3 Data Mixing and Model Training

Incorporating a diverse data mix—spanning various languages, domains, document lengths and styles—is crucial for continual training of large language models to enhance their versatility, generalization ability, and robustness across a wide range of tasks and domains. We augment the MaLA corpus with diverse data to mitigate issues such as over-fitting to specific styles or topics or underperforming on tasks outside the training distribution.

3.1 Data Mixing

Curated Data We enhance the corpus with high-quality curated data, specifically high-resource languages in the monolingual part. We use texts from scientific papers as these provide a structured, information-dense corpus that can improve the model's ability to handle technical language and domain-specific content. They are (1) CSL (Li et al., 2022), a large-scale Chinese Scientific Literature dataset, that contains titles, abstracts, keywords and academic fields of 396,209 papers; (2) pes2o (Soldaini and Lo, 2023), a collection of full-text open-access academic papers derived from the Semantic Scholar Open Research Corpus (S2ORC) (Lo et al., 2020). We further add free e-books from the Gutenberg project[6] compiled by Faysse (2023). These texts enhance the range of literary styles

[5] `https://huggingface.co/datasets/cis-lmu/Glot500`
[6] `https://www.gutenberg.org/`

Table 2: Data mix for continual training. Code and reasoning-related data are counted by Llama 2 tokenizer and others are counted as whitespace delimited; 'inst' stands for instruction and 'mono' stands for monolingual texts.

Data	Original Counts	Sample Rate	Final Counts	Percentage
inst high	42,121,055,562	0.1	4,212,105,556	3.08%
inst medium-high+	6,486,592,274	0.2	1,297,318,455	0.95%
inst medium-high	30,651,187,534	0.5	15,325,593,767	11.21%
inst medium	1,444,764,863	1.0	1,444,764,863	1.06%
inst medium-low	47,691,495	5.0	238,457,475	0.17%
inst low	3,064,796	20.0	61,295,920	0.04%
inst code/reasoning	612,208,775	1.0	612,208,775	0.45%
code	221,003,976,266	0.1	20,786,882,764	15.20%
curated (EN pes2o)	56,297,354,921	0.2	11,241,574,489	8.22%
curated (ZH CSL & wiki)	61,787,372	1.0	61,787,372	0.05%
curated (Gutenberg)	5,173,357,710	1.0	5,173,357,710	3.78%
mono high EN	3,002,029,817	0.1	300,202,982	0.22%
mono high	40,411,201,964	0.5	20,205,600,982	14.78%
mono medium-high	27,515,227,962	1.0	27,515,227,962	20.12%
mono medium	2,747,484,380	5.0	13,737,421,900	10.05%
mono medium-low	481,935,633	20.0	9,638,712,660	7.05%
mono low	97,535,696	50.0	4,876,784,800	3.57%

and narrative forms, thus enhancing the model's versatility. Adding high-resource languages into the pre-training corpora also mitigates the forgetting in model training.

Instruction Data We further augmented the training corpus by incorporating instruction-based datasets, inspired by Li et al. (2023); Nakamura et al. (2024); Taylor et al. (2022). We mix two instruction data into our training corpus. They are: (1) xp3x (Crosslingual Public Pool of Prompts eXtended)[7], a multitask instruction collection in 277 languages (Muennighoff et al., 2022); (2) the Aya collection [8] that contains both human-curated and machine translated instructions in 101 languages (Singh et al., 2024). For both instruction datasets, we use their training set.

Code We additionally enrich the training corpus by sourcing code data from The Stack (Kocetkov et al., 2023). This is done following existing work that demonstrates the value of code data in improving the reasoning ability of language models (Ma et al., 2024; Zhang et al., 2024b) while also mitigating any catastrophic forgetting of the base model's programming knowledge.

We subsample The Stack at an effective rate of 15.2%, prioritizing high-quality source files and data science code [9]. We retain the 32 most important non-data programming languages by prevalence while also adding in all `llvm` code following prior work detailing its importance in multi-lingual code generation (Paul et al., 2024; Szafraniec et al., 2023). We also source from data-heavy formats but follow precedent (Lozhkov et al., 2024) and subsample them more aggressively. For a more detailed read on filtering heuristics, we direct the reader to Appendix A.2.

Data Mix Our final data mix for continual training is listed in Table 2. The resource categorization refers to Section B.1 in the appendix and `inst medium-high+` is a separate category with languages with more than 500 million but less than 1B tokens. For monolingual text, we also have a seperate category for English. In continual learning, where new data is introduced to an existing model, there is a risk of "catastrophic forgetting", where the model loses knowledge from earlier training stages. Although our work's primary focus is in a low-resource regime, we enhance the training corpus with a wide range of data types, including books and scientific papers in high-resource languages, code, and instruction data in our data mix. We downsample texts in high-resource languages and upsample text in low-resource languages using different sample rates according to how resourceful the language is. We make our data mix diverse and balanced towards different resource groups of languages in order to retain the prior knowledge of the model while learning new information, especially in medium- and low-resource languages, thus maintaining high performance across both previously seen and new languages. The final data mix has around 136B tokens.

[7] https://huggingface.co/datasets/CohereForAI/xP3x

[8] https://huggingface.co/datasets/CohereForAI/aya_collection_language_split

[9] https://huggingface.co/datasets/AlgorithmicResearchGroup/arxiv_research_code

Table 3: Evaluation statistics. Sample/Lang: average number of test samples per language; N Lang: number of languages covered; NLL: negative log-likelihood; ACC: accuracy.

Tasks	Dataset	Metric	Samples/Lang	N Lang	Domain
Intrinsic Evaluation	Glot500-c test (Imani et al., 2023)	NLL	1000	534	Misc
	PBC (Mayer and Cysouw, 2014)	NLL	500	370	Bible
Text Classification	SIB200 (Adelani et al., 2023)	ACC	204	205	Misc
	Taxi1500 (Ma et al., 2023)	ACC	111	1507	Bible
Commonsense Reasoning	XCOPA (Ponti et al., 2020)	ACC	600	11	Misc
	XStoryCloze (Lin et al., 2022)	ACC	1870	11	Misc
	XWinograd (Tikhonov and Ryabinin, 2021)	ACC	741.5	6	Misc
Natural Language Inference	XNLI (Conneau et al., 2018)	ACC	2490	15	Misc
Machine Translation	FLORES-200 (Costa-jussà et al., 2022)	BLEU, chrF++	1012	204	Misc
Open-Ended Generation	Aya (Singh et al., 2024)	BLEU, Self-BLEU	215	119	Misc
	PolyWrite (Ours)	Self-BLEU	149	240	Misc
Summarization	XL-Sum (Hasan et al., 2021)	ROUGE-L, BERTScore	2537	44	News
Math	MGSM direct (Shi et al., 2022)	ACC	250	10	Misc
	MGSM CoT (Shi et al., 2022)	ACC	250	10	Misc
Machine Comprehension	BELEBELE (Bandarkar et al., 2023)	ACC	900	122	Misc
	ARC multilingual (Lai et al., 2023)	ACC	1170	31	Misc
Code Generation	Multipl-E (Cassano et al., 2022)	Pass@k	164	7	Misc

3.2 Model Training

We employ continual training using the causal language modelling objective for the decoder-only Llama model and exposing the pre-trained model to new data to develop our EMMA-500 model. We adopt efficient training strategies combining optimization, memory management, precision handling, and distributed training techniques. Our EMMA-500 model is trained on the Leonardo supercomputer[10], occupying 256 Nvidia A100 GPUs, using the GPT-NeoX framework (Andonian et al., 2023). During training, we set a global batch size of 4096 and worked with sequences of 4096 tokens. The training process ran for 12,000 steps, resulting in a total of 200 billion Llama 2 tokens. We use the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 0.0001, betas of [0.9, 0.95], and an epsilon of 1e-8. We use a cosine learning rate scheduler with a warm-up of 500 iterations. To reduce memory consumption, activation checkpointing is employed. Precision is managed through mixed-precision techniques, using bfloat16 for computational efficiency and maintaining FP32 for gradient accumulation.

4 Evaluation

4.1 Tasks, Benchmarks, and Baselines

Tasks and Benchmarks We conduct a comprehensive evaluation to validate the usability of our processed data and data mixing for massively multilingual language adaptation. We perform the intrinsic evaluation of the models' performance on next-word prediction and evaluate the model's performance on downstream tasks. Table 3 lists the datasets we used as downstream evaluation datasets in this work. For math tasks, we perform direct prompting and Chain-of-Thoughts prompting on the MGSM benchmark and evaluate the performance using both exact and flexible matches. For code generation tasks, we perform test-case-based execution-tested evaluations using the `pass@k` metric (Chen et al., 2021). We benchmark for `k` values of 1,10, and 25 using a generation pool of 50 samples per problem.

Baselines We compare our model with three groups of decoder-only models. They are (1) Llama 2 models (Touvron et al., 2023) and continual pre-trained models based on Llama 2, such as CodeLlama (Roziere et al., 2023), MaLA-500 (Lin et al., 2024), LLaMAX (Lu et al., 2024), Tower (Alves et al., 2024), and YaYi[11]; (2) other LLMs and continual pre-trained LLMs designed to be massively

[10] `https://leonardo-supercomputer.cineca.eu`

[11] `https://huggingface.co/wenge-research/yayi-7b-llama2`

multilingual, including BLOOM (Scao et al., 2022), mGPT (Shliazhko et al., 2022), XGLM (Lin et al., 2022), and Occiglot[12]; and (3) recent LLMs with superior English capabilities like Llama 3 (Dubey et al., 2024), Llama 3.1[13], Qwen 2 (Yang et al., 2024a), and Gemma 2 (Team et al., 2024). There are also some other LLMs such as OpenAI's API models [14] and xLLMs-100 (Lai et al., 2024). However, they do not release the model weights or they limit access to them through commercial API, so we did not include them. The MADLAD model (Kudugunta et al., 2024) that uses the decoder-only T5 architecture is not supported by inference engines such as the HuggingFace transformers (Wolf et al., 2019). We do not compare them in this work.

Evaluation Software We use the Language Model Evaluation Harness (`lm-evaluation-harness`) framework (Gao et al., 2023) for benchmarking test sets that are already ingested in the framework. For other benchmarks, we use in-house developed evaluation scripts and other open-source implementations. In text classification tasks such as SIB-200 and Taxi-1500, the evaluation protocol involves calculating the probability of the next output token for each candidate category. These probabilities are then sorted in descending order, with the category having the highest probability being selected as the model's prediction. This process is implemented using the Transformers (Wolf et al., 2019) library. In tasks like Machine Translation and Open-Ended Generation, the vLLM library (Kwon et al., 2023) is used to accelerate inference, providing significant speed improvements. Similarly, for code generation tasks, we use a VLLM-enabled evaluation harness package (`vllm-code-harness`)[15] for the execution-tested evaluations. For massively multilingual benchmarks with more than 100 languages, we categorize languages into five groups as listed in Table 20 in the appendix.

4.2 Intrinsic Evaluation

For intrinsic evaluation, we compute the negative log-likelihood (NLL) of the test text given by the tested LLMs instead of using length-normalized perplexity due to different text tokenization schemes across models. We concatenate the test set as a single input and run a sliding-window approach.[16] To make a comparison across models, we use the Glot500-c test set, which covers 534 evaluated languages. We also test on the Parallel Bible Corpus (PBC) which could yield NLL more comparable across languages. Table 4 and Table 5 show the intrinsic evaluation results on Glot500-c test and PBC, respectively. As shown, our model attains lower NLL compared to all other models on both test sets and languages with different resource availability. This suggests that the test sets are more similar to the underlying training data of our model and it can be interpreted that EMMA-500 has learned to compress massive multilingual text more efficiently.

4.3 Commonsense Reasoning

We assess the models' commonsense reasoning ability using three multilingual benchmarks. XCOPA (Ponti et al., 2020) is a dataset covering 11 languages that focuses on causal commonsense reasoning across multiple languages. XStoryCloze (Lin et al., 2022) is derived from the English StoryCloze dataset (Mostafazadeh et al., 2017) and translated into 10 non-English languages, testing commonsense reasoning within a story. In this task, the system must choose the correct ending for a four-sentence narrative. XWinograd (Tikhonov and Ryabinin, 2021) is a multilingual collection of Winograd Schemas (Levesque et al., 2012) available in six languages, aimed at evaluating cross-lingual commonsense reasoning. We perform zero-shot evaluations. Accuracy is used as the evaluation metric. We categorize languages into different resource groups based on language availability, accessibility, and possible corpus size. For XCOPA, we have three groups, i.e., high-resource (Italian, Turkish, Vietnamese, and Chinese), medium-resource (Swahili due to its regional influence, Estonian, Haitian Creole, Indonesian, Thai, and Tamil), and low-resource languages (Cusco-Collao Quechua).[17] For XStoryCloze, the resource group is high-resource (Arabic, English, Spanish, Russian, and Chinese), medium (Hindi, Indonesian, Swahili, and Telugu), and low (Basque and Burmese). For XWinograd,

[12] https://huggingface.co/occiglot/occiglot-7b-eu5

[13] https://llama.meta.com/docs/model-cards-and-prompt-formats/llama3_1

[14] https://platform.openai.com/docs/models

[15] https://github.com/iNeil77/vllm-code-harness

[16] https://huggingface.co/docs/transformers/en/perplexity

[17] Note that there is no perfect categorization for language resource groups.

Table 4: NLL on Glot500-c test. EMMA-500 Llama 2 7B has better average performance than all baselines.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	190.58	146.43	176.30	205.86	210.41	196.45
Llama 2 7B Chat	218.87	173.67	204.78	239.18	240.97	223.86
CodeLlama 2 7B	193.96	146.43	180.20	210.09	212.70	200.60
LLaMAX Llama 2 7B	187.37	108.58	142.84	197.74	212.22	203.83
LLaMAX Llama 2 7B Alpaca	169.12	94.84	123.90	173.54	193.83	187.34
MaLA-500 Llama 2 10B v1	155.62	127.51	153.93	173.02	166.01	158.12
MaLA-500 Llama 2 10B v2	151.25	123.82	147.69	167.46	161.20	155.13
TowerBase Llama 2 7B	192.98	150.41	180.19	209.12	212.18	198.89
TowerInstruct Llama 2 7B	199.44	157.33	186.42	216.92	218.82	204.93
Occiglot Mistral 7B v0.1	191.11	159.48	185.53	209.26	207.67	194.07
Occiglot Mistral 7B v0.1 Instruct	193.83	162.31	188.20	212.41	210.81	196.58
BLOOM 7B	202.95	160.33	195.01	216.15	220.46	206.89
BLOOMZ 7B	217.32	178.12	210.99	235.53	235.60	220.65
mGPT	340.37	311.14	337.29	388.97	367.45	343.14
XGLM 7.5B	205.07	199.08	210.22	225.53	214.92	205.67
Yayi 7B	226.67	181.48	217.34	243.42	246.58	231.65
Llama 3 8B	156.36	102.78	129.36	153.11	167.26	173.56
Llama 3.1 8B	154.59	101.23	127.57	150.87	164.81	172.05
Gemma 7B	692.25	583.39	721.77	817.40	729.61	689.60
Gemma 2 9B	320.81	348.26	351.08	380.49	338.00	303.62
Qwen 2 7B	188.55	132.47	171.50	200.26	210.21	196.78
Qwen 1.5 7B	195.52	141.37	181.41	212.10	217.16	202.46
EMMA-500 Llama 2 7B	112.20	81.78	100.89	122.53	99.28	109.25

Table 5: NLL on PBC. EMMA-500 Llama 2 7B has better average performance than all baselines.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	122.10	91.30	99.41	112.31	133.08	135.34
Llama 2 7B Chat	139.14	108.40	115.78	129.79	149.54	152.82
CodeLlama 2 7B	123.98	93.27	101.83	113.47	134.65	137.52
LLaMAX Llama 2 7B	117.39	69.41	79.06	103.74	131.90	138.03
LLaMAX Llama 2 7B Alpaca	107.81	60.05	69.36	93.39	122.44	128.77
MaLA-500 Llama 2 10B v1	103.20	94.04	98.60	100.53	105.04	107.65
MaLA-500 Llama 2 10B v2	101.67	92.42	96.30	98.88	103.34	106.62
TowerBase Llama 2 7B	123.70	93.64	101.44	114.72	134.41	136.77
TowerInstruct Llama 2 7B	127.30	98.21	105.17	118.65	137.53	140.28
Occiglot Mistral 7B v0.1	121.64	95.15	101.86	114.37	131.49	132.93
Occiglot Mistral 7B v0.1 Instruct	123.18	96.86	103.41	115.88	132.98	134.48
BLOOM 7B	129.55	96.62	111.22	115.33	138.19	143.03
BLOOMZ 7B	137.72	107.03	119.89	125.85	145.27	150.95
mGPT	225.14	211.35	203.84	219.75	229.91	239.82
XGLM 7.5B	131.31	116.86	117.69	125.15	136.47	140.53
Yayi 7B	143.80	108.79	123.20	130.60	152.37	158.73
Llama 3 8B	102.55	64.20	71.82	85.22	114.79	121.29
Llama 3.1 8B	101.43	62.98	70.68	83.83	113.36	120.33
Gemma 7B	460.86	399.22	427.14	468.66	463.11	483.29
Gemma 2 9B	197.41	200.76	192.07	197.88	196.56	202.69
Qwen 2 7B	120.44	83.34	94.87	107.84	133.38	136.52
Qwen 1.5 7B	124.02	89.36	100.55	113.08	135.54	139.13
EMMA-500 Llama 2 7B	68.11	50.12	55.62	64.78	60.53	65.68

Table 6: 0-shot results (ACC) on commonsense reasoning: XCOPA, XStoryCloze, and XWinograd. EMMA-500 Llama 2 7B has better average performance than Llama 2 models and multilingual LLMs on XCOPA, XStoryCloze, and comparable performance on XWinograd.

Model	XCOPA				XStoryCloze				XWinograd
	Avg	High	Medium	Low	Avg	High	Medium	Low	Avg
Llama 2 7B	0.5667	0.6210	0.5390	0.5160	0.5755	0.6338	0.5445	0.4921	0.7247
Llama 2 7B Chat	0.5585	0.6125	0.5313	0.5060	0.5841	0.6480	0.5477	0.4974	0.6945
CodeLlama 2 7B	0.5469	0.5870	0.5253	0.5160	0.5568	0.6068	0.5233	0.4990	0.7092
LLaMAX Llama 2 7B	0.5438	0.5550	0.5413	0.5140	0.6036	0.6434	0.5882	0.5347	0.6749
LLaMAX Llama 2 7B Alpaca	0.5660	0.5980	0.5517	0.5240	0.6383	0.6908	0.6201	0.5433	0.6986
MaLA-500 Llama 2 10B v1	0.5309	0.5355	0.5327	0.5020	0.5307	0.5815	0.4922	0.4808	0.6589
MaLA-500 Llama 2 10B v2	0.5309	0.5355	0.5327	0.5020	0.5307	0.5815	0.4922	0.4808	0.6589
YaYi Llama 2 7B	0.5671	0.6210	0.5413	0.5060	0.5842	0.6498	0.5491	0.4904	0.7450
TowerBase Llama 2 7B	0.5633	0.6250	0.5290	0.5220	0.5778	0.6435	0.5367	0.4957	0.7429
TowerInstruct Llama 2 7B	0.5705	0.6290	0.5400	0.5200	0.5924	0.6683	0.5453	0.4970	0.7400
Occiglot Mistral 7B v0.1	0.5667	0.6280	0.5337	0.5200	0.5810	0.6518	0.5328	0.5003	0.7461
Occiglot Mistral 7B v0.1 Instruct	0.5655	0.6285	0.5297	0.5280	0.5939	0.6694	0.5433	0.5063	0.7293
BLOOM 7B	0.5689	0.5995	0.5587	0.5080	0.5930	0.6199	0.5905	0.5308	0.7013
BLOOMZ 7B	0.5487	0.5635	0.5460	0.5060	0.5712	0.6114	0.5582	0.4967	0.6795
mGPT	0.5504	0.5710	0.5440	0.5060	0.5443	0.5496	0.5453	0.5291	0.5969
mGPT 13B	0.5618	0.5975	0.5513	0.4820	0.5644	0.5776	0.5635	0.5331	0.6359
XGLM 7.5B	0.6064	0.6400	0.6037	0.4880	0.6075	0.6242	0.6036	0.5738	0.6884
YaYi 7B	0.5664	0.5955	0.5550	0.5180	0.6067	0.6490	0.5940	0.5265	0.6979
Llama 3 8B	0.6171	0.6835	0.5903	0.5120	0.6341	0.6850	0.6203	0.5344	0.7684
Llama 3.1 8B	0.6171	0.6930	0.5880	0.4880	0.6358	0.6866	0.6209	0.5387	0.7552
Gemma 7B	0.6364	0.7035	0.6143	0.5000	0.6501	0.6946	0.6449	0.5493	0.7741
Gemma 2 9B	0.6633	0.7340	0.6427	0.5040	0.6767	0.7247	0.6669	0.5764	0.8007
Qwen 2 7B	0.6031	0.6865	0.5640	0.5040	0.6146	0.6945	0.5697	0.5050	0.7644
Qwen 1.5 7B	0.5944	0.6685	0.5590	0.5100	0.5985	0.6662	0.5604	0.5056	0.7259
EMMA-500 Llama 2 7B	0.6311	0.6660	0.6257	0.5240	0.6638	0.6892	0.6573	0.6132	0.7280

there is only one group for high-resource languages. Table 6 shows the evaluation results of zero-shot commonsense reasoning.

Compared with Llama 2-based models on XCOPA, our model improves the average performance by a large margin—up to a 5% increase when compared with the best-performing TowerInstruct based on Llama 2 7B.[18] Our model also outperforms all the multilingual LLMs. Recent LLMs such as Gemma and Llama 3 have stronger performance than Llama 2 models, and Gemma 2 9B performs the best. However, our model achieves better performance than Qwen, Llama 3, and 3.1. We gain similar results on XStoryCloze, outperforming all the models except Gemma 2 9B. As for XWinograd, a multilingual benchmark with high-resource only, our model achieves improved performance than Llama 2, despite not being as good as Tower models, which target high-resource languages. However, our model is comparable to other multilingual LLMs. For low-resource languages, our model outperforms all the compared LLMs except LLaMAX Llama 2 7B Alpaca on XCOPA, where we achieve the same accuracy as it.

4.4 Machine Translation

FLORES-200 is an evaluation benchmark for translation tasks with 204 language pairs involving English and thus 408 translation directions, with a particular focus on low-resource languages. We assess all language models by adopting a 3-shot evaluation approach with the prompt:

```
Translate the following sentence from {src_lang} to {tgt_lang}
[{src_lang}]: {src_sent}
[{tgt_lang}]:
```

The performance is measured by BLEU (Papineni et al., 2002) and chrF++ (Popović, 2015) implemented in `sacrebleu` (Post, 2018). The BLEU score is calculated with the `flores200` tokenizer applied to the texts and chrF++ uses word order 2. The choice of `flores200` tokenization ensures that languages that do not have a whitespace delimiter can be evaluated at the (sub-)word level. For

[18]The biggest improvements are on medium-resource languages. However, we note that the resource categorization is not perfect.

Table 7: 3-shot results on FLORES-200 (X-Eng, BLEU/chrF++). EMMA-500 Llama 2 7B has better average performance than all baselines.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	12.93/ 30.32	19.91/ 39.04	17.56/ 35.84	12.49/ 29.81	8.27/ 24.35	6.96/ 23.36
Llama 2 7B Chat	12.28/ 31.72	18.98/ 39.65	17.06/ 37.03	11.74/ 31.1	7.79/ 26.34	6.18/ 25.03
CodeLlama 2 7B	10.82/ 28.57	17.39/ 37.43	15.27/ 33.94	10.39/ 28.05	6.45/ 22.85	5.04/ 21.48
LLaMAX Llama 2 7B	1.99/ 13.66	3.68/ 22.18	2.95/ 18.15	1.83/ 12.84	0.67/ 7.2	1.01/ 9.04
LLaMAX Llama 2 7B Alpaca	22.29/ 42.27	32.83/ 54.56	30.04/ 51.25	21.7/ 41.94	13.06/ 31.32	14.24/ 32.88
MaLA-500 Llama 2 10B v1‡	2.29/ 13.6	4.64/ 15.95	3.18/ 14.64	2.68/ 14.23	1.24/ 12.58	0.33/ 11.18
MaLA-500 Llama 2 10B v2‡	2.87/ 15.44	5.58/ 18.65	3.81/ 16.33	3.55/ 16.29	1.63/ 14.2	0.55/ 12.76
Yayi Llama 2 7B	12.98/ 31.38	19.48/ 39.58	17.55/ 36.71	12.47/ 30.79	8.54/ 25.63	7.22/ 24.84
TowerBase Llama 2 7B	13.74/ 31.47	21.76/ 40.96	18.92/ 37.27	13.15/ 30.9	8.3/ 25.05	7.21/ 24.1
TowerInstruct Llama 2 7B	4.81/ 25.43	9.18/ 34.4	6.66/ 30.01	4.62/ 25.22	2.64/ 20.24	1.8/ 18.69
Occiglot Mistral 7B v0.1	13.12/ 31.13	19.53/ 38.93	17.57/ 36.27	13.07/ 31.2	9.03/ 26.15	6.86/ 23.83
Occiglot Mistral 7B v0.1 Instruct	11.61/ 31.65	16.72/ 39.28	15.06/ 36.48	11.7/ 31.73	8.48/ 26.88	6.54/ 24.7
BLOOM 7B	9.57/ 27.84	15.75/ 36.65	9.65/ 28.19	9.42/ 27.81	6.81/ 23.95	8.61/ 25.89
BLOOMZ 7B†	20.22/ 34.74	32.23/ 47.03	19.2/ 34.08	20.09/ 34.49	16.25/ 30.58	18.54/ 32.63
mGPT	5.29/ 20.69	9.37/ 26.64	8.28/ 25.29	3.41/ 17.87	2.43/ 16.07	2.84/ 17.28
mGPT-13B	7.42/ 24.58	12.61/ 31.95	11.11/ 30.16	5.72/ 22.49	3.57/ 18.16	4.11/ 20.04
Yayi 7B	4.82/ 21.36	5.69/ 25.18	4.53/ 19.97	4.41/ 21.52	3.71/ 19.18	6.13/ 23.12
Llama 3 8B	23.78/ 43.72	33.71/ 55.36	30.31/ 51.3	24.75/ 44.91	15.18/ 33.65	16.01/ 34.65
Llama 3.1 8B	24.19/ 44.1	34.15/ 55.7	30.79/ 51.7	24.98/ 45.26	15.89/ 34.24	16.13/ 34.85
Gemma 2 9B	23.15/ 38.87	33.11/ 51.36	30.81/ 48.53	25.58/ 41.23	15.37/ 30.03	11.73/ 24.15
Gemma 7B	23.79/ 43.68	34.23/ 55.77	29.87/ 50.95	24.0/ 44.25	16.16/ 34.36	16.03/ 34.58
Qwen 1.5 7B	15.58/ 35.87	24.07/ 46.29	19.92/ 40.74	15.76/ 36.27	9.74/ 28.81	9.77/ 29.13
Qwen 2 7B	17.39/ 37.61	27.63/ 50.06	22.48/ 43.28	18.13/ 38.63	9.89/ 28.54	10.64/ 29.99
EMMA-500 Llama 2 7B	25.37/ 45.78	32.24/ 53.74	31.39/ 52.85	25.72/ 46.16	20.32/ 39.96	17.18/ 36.15

reproducibility, we attach the BLEU and chrF++ signatures.[19,20]

Table 7 presents the average X-to-English (X-Eng) translation results.[21] Our EMMA-500 model outperforms all other models on average. We achieve the best performance across all language settings, except for high-resource languages where our model slightly lags behind Llama 3/3.1, Gemma 7B, and LlaMAX 7B Alpaca. In the English-to-X (Eng-X) translation direction, as shown in Table 8, the advantage of EMMA-500 is even more pronounced. We outperform all other models even in high-resource languages, and the advantage becomes more significant in lower-resource languages. Overall, we note that our model outperforms Tower models which are explicitly adjusted to perform translation tasks in high-resource languages. Further, the much larger margin between EMMA-500 and other models in Eng-X compared with X-Eng indicates that our EMMA-500 model is particularly good at generating non-English texts.

4.5 Text Classification

SIB-200 (Adelani et al., 2023) and Taxi1500 (Ma et al., 2023) are two prominent topic classification datasets. SIB-200 encompasses seven categories: science/technology, travel, politics, sports, health, entertainment, and geography. Taxi1500 spans 1507 languages, involving six classes: Recommendation, Faith, Description, Sin, Grace, and Violence.

We use 3-shot prompting, drawing demonstrations from the development set and testing models on the test split. The prompt template for SIB-200 is as follows:

```
Topic Classification: science/technology, travel, politics, sports, health,
    entertainment, geography.
{examples}
The topic of the news "${text}" is
```

For Taxi-1500, the prompt template is as follows:

```
Topic Classification: Recommendation, Faith, Description, Sin, Grace,
    Violence.
```

[19]BLEU: nrefs:1—case:mixed—eff:no—tok:flores200—smooth:exp—version:2.4.2

[20]chrF++: nrefs:1—case:mixed—eff:yes—nc:6—nw:2—space:no—version:2.4.2

[21]We mark BLOOMZ with a † because it has used FLORES in its instruction tuning data; we mark MaLA-500 with a ‡ because it has used FLORES in its training data but with source and target sides split. Besides, as a remark, Tower, LLaMAX, and our EMMA-500 have intentionally used parallel data (not FLORES) in the training stage.

Table 8: 3-shot results on FLORES-200 (Eng-X, BLEU/chrF++). EMMA-500 Llama 2 7B has better average performance than all baselines.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	4.62/ 15.13	10.77/ 24.38	8.56/ 21.4	2.55/ 13.72	0.74/ 8.72	0.7/ 7.92
Llama 2 7B Chat	4.95/ 16.95	10.87/ 24.51	8.54/ 22.69	3.25/ 15.5	1.52/ 12.08	0.94/ 10.03
CodeLlama 2 7B	4.27/ 14.94	10.04/ 23.48	7.79/ 20.79	2.57/ 14.2	0.71/ 9.27	0.58/ 7.49
LLaMAX Llama 2 7B	0.8/ 7.42	1.85/ 12.06	1.2/ 9.74	0.54/ 6.55	0.22/ 4.52	0.38/ 4.81
LLaMAX Llama 2 7B Alpaca	12.51/ 28.35	24.8/ 41.76	18.69/ 38.42	10.1/ 27.27	3.79/ 16.53	6.68/ 18.15
MaLA-500 Llama 2 10B v1‡	0.6/ 6.08	1.51/ 9.0	1.13/ 8.19	0.35/ 5.99	0.07/ 4.5	0.02/ 2.9
MaLA-500 Llama 2 10B v2‡	0.54/ 6.38	1.4/ 9.19	1.02/ 8.42	0.24/ 5.99	0.07/ 5.14	0.02/ 3.27
Yayi Llama 2 7B	4.41/ 14.87	10.49/ 24.0	8.21/ 21.27	2.52/ 13.57	0.6/ 8.49	0.53/ 7.42
TowerBase Llama 2 7B	4.83/ 16.03	11.89/ 24.15	8.33/ 21.46	2.57/ 14.49	1.38/ 11.6	0.74/ 8.9
TowerInstruct Llama 2 7B	3.23/ 15.64	7.22/ 22.65	4.99/ 20.0	2.2/ 14.9	1.62/ 12.31	0.73/ 8.97
Occiglot Mistral 7B v0.1	4.32/ 16.1	10.5/ 23.74	6.95/ 20.91	2.87/ 15.44	1.47/ 12.0	0.79/ 9.09
Occiglot Mistral 7B v0.1 Instruct	3.99/ 15.8	9.46/ 23.17	6.46/ 20.73	2.68/ 15.29	1.31/ 11.31	0.84/ 9.04
BLOOM 7B	2.81/ 11.8	7.53/ 19.0	3.12/ 13.36	2.05/ 11.48	0.85/ 8.0	2.09/ 9.22
BLOOMZ 7B†	7.44/ 16.1	23.64/ 32.22	7.46/ 16.62	6.98/ 16.05	1.28/ 9.99	4.17/ 11.77
mGPT	2.59/ 12.56	5.24/ 17.04	4.75/ 16.92	1.14/ 9.75	0.78/ 9.24	0.84/ 9.08
mGPT-13B	3.88/ 14.57	8.32/ 21.7	6.84/ 20.55	2.06/ 12.23	0.9/ 8.4	1.33/ 9.58
Yayi 7B	4.37/ 13.5	13.72/ 26.28	4.68/ 14.31	3.35/ 12.89	0.91/ 8.51	2.55/ 10.08
Llama 3 8B	9.93/ 24.08	20.38/ 36.87	14.95/ 32.05	8.89/ 24.28	2.83/ 14.26	4.2/ 14.29
Llama 3.1 8B	10.11/ 24.69	20.82/ 37.39	15.3/ 32.82	8.85/ 24.85	2.9/ 14.83	4.23/ 14.81
Gemma 2 9B	12.09/ 26.48	24.62/ 40.69	17.82/ 35.51	10.68/ 26.58	3.38/ 15.02	5.94/ 15.98
Gemma 7B	9.05/ 23.05	17.58/ 34.5	13.62/ 30.16	7.96/ 22.85	2.64/ 14.11	4.47/ 14.82
Qwen 1.5 7B	5.87/ 17.77	14.05/ 28.6	8.88/ 23.57	3.85/ 17.07	1.7/ 10.85	2.35/ 10.21
Qwen 2 7B	5.56/ 17.17	13.22/ 27.65	8.21/ 22.36	4.15/ 16.93	1.56/ 10.47	2.19/ 10.11
EMMA-500 Llama 2 7B	15.58/ 33.25	26.37/ 42.4	21.96/ 41.98	13.4/ 32.06	9.15/ 27.99	7.92/ 21.25

```
{examples}
The topic of the verse "${text}" is
```

The outcomes are tabulated in Table 9 and Table 10 for SIB-200 and TAXI-1500 respectively. For SIB-200, our EMMA-500 model outperforms all Llama2-based models, with particularly notable gains in languages with medium or fewer resources—seeing an average improvement of 47.5%. Taxi-1500 could be a more challenging task since it is in the religious domain, but our model still surpasses all Llama2-based models except for MalA-500. However, despite these improvements in both classification tasks, our models lag behind the latest models such as Llama3 and 3.1, especially in high-resource languages.

4.6 Open-Ended Generation

Aya Evaluation We choose the two subsets `aya-human-annotated` and `dolly-machine-translated` from the Aya evaluation suite (Singh et al., 2024), which have both inputs and targets for subsequent evaluation. To quantitatively assess the quality of the generated text by the models, we employ two metrics: BLEU (Papineni et al., 2002) and Self-BLEU (Zhu et al., 2018). BLEU is crucial for assessing the linguistic accuracy and relevance of the generated text in comparison to the expected human-like text present in the dataset. Self-BLEU is used to evaluate the diversity of the text generated by a model. It measures how similar different texts from the same model are to each other by treating one generated text as the "candidate" and others as the "reference" texts. This metric is useful in scenarios where high degrees of variation are desirable, as it helps identify models that might be overfitting to particular styles or patterns of text.

The results are presented in Table 11. The BLEU metric is an indicator that measures how generated texts are close to the references. However, in open-ended generation, LLMs cannot generate identical texts to references, leading to low BLEU scores. The EMMA-500 model obtains remarkably high BLEU scores in both high-resource and low-resource settings when compared with baselines. Its performance in low-resource settings is particularly noteworthy, as it not only sustains high BLEU scores but also exhibits a Self-BLEU score of 5.09, the highest among all models evaluated. This high Self-BLEU score indicates less diversity in the generated text, suggesting that while EMMA-500 maintains consistency, it may produce less varied outputs. When compared to other high-performing models like Qwen 2 7B and Llama 3.1 8B, the EMMA-500 model exhibits a superior balance between accuracy and linguistic creativity. Unlike Qwen 2 7B, which shows a spike

Table 9: 3-shot results on SIB-200 (ACC). EMMA-500 Llama 2 7B has better average performance than Llama 2 models and comparable performance with multilingual LLMs.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	0.2241	0.2664	0.2469	0.2205	0.1968	0.1900
Llama 2 7B Chat	0.2558	0.2972	0.2811	0.2501	0.2303	0.2191
CodeLlama 2 7B	0.2335	0.2606	0.2542	0.2310	0.2142	0.2037
LLaMAX Llama 2 7B	0.1061	0.1242	0.1160	0.1001	0.0945	0.0954
LLaMAX Llama 2 7B Alpaca	0.2789	0.3309	0.3212	0.2716	0.2338	0.2282
MaLA-500 Llama 2 10B v1	0.2325	0.2330	0.2364	0.2288	0.2276	0.2358
MaLA-500 Llama 2 10B v2	0.1930	0.1893	0.2105	0.1949	0.1755	0.1846
Yayi Llama 2 7B	0.2457	0.2904	0.2717	0.2442	0.2144	0.2069
TowerBase Llama 2 7B	0.1934	0.2200	0.2092	0.1874	0.1790	0.1693
TowerInstruct Llama 2 7B	0.2053	0.2321	0.2196	0.2026	0.1915	0.1804
Occiglot Mistral 7B v0.1	0.3269	0.3880	0.3582	0.3174	0.2892	0.2836
Occiglot Mistral 7B v0.1 Instruct	0.3431	0.3948	0.3716	0.3336	0.3147	0.3008
BLOOM 7B	0.1781	0.2313	0.1805	0.1717	0.1576	0.1702
BLOOMZ 7B	0.2973	0.3039	0.2963	0.2980	0.2953	0.2970
mGPT	0.2711	0.2858	0.2799	0.2673	0.2589	0.2648
mGPT-13B	0.3320	0.3669	0.3427	0.3448	0.2939	0.3226
XGLM 7.5B	0.3181	0.3528	0.3512	0.3169	0.2696	0.2996
Yayi 7B	0.3576	0.4057	0.3620	0.3563	0.3472	0.3318
Llama 3 8B	0.6369	0.7345	0.7025	0.6462	0.5316	0.5696
Llama 3.1 8B	0.6142	0.7070	0.6790	0.6199	0.5146	0.5475
Gemma 7B	0.5821	0.6806	0.6455	0.5816	0.4886	0.5112
Gemma 2 9B	0.4625	0.5177	0.4900	0.4692	0.4304	0.4045
Qwen 1.5 7B	0.4795	0.5600	0.5181	0.4825	0.4156	0.4286
Qwen 2 7B	0.5495	0.6637	0.6014	0.5517	0.4519	0.4925
EMMA-500 Llama 2 7B	0.3127	0.3275	0.3328	0.3099	0.3083	0.2760

Table 10: 3-shot results on Taxi-1500 (ACC). EMMA-500 Llama 2 7B has comparable performance with the compared LLMs.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	0.1754	0.1950	0.1949	0.1847	0.1746	0.1737
Llama 2 7B Chat	0.1544	0.1873	0.1766	0.1661	0.1559	0.1522
CodeLlama 2 7B	0.1703	0.1745	0.1741	0.1720	0.1705	0.1700
LLaMAX Llama 2 7B	0.2352	0.2320	0.2340	0.2376	0.2356	0.2352
LLaMAX Llama 2 7B Alpaca	0.1509	0.1870	0.1688	0.1599	0.1500	0.1491
MaLA-500 Llama 2 10B v1	0.2527	0.2390	0.2402	0.2476	0.2457	0.2543
MaLA-500 Llama 2 10B v2	0.2339	0.2136	0.2230	0.2132	0.2172	0.2367
Yayi Llama 2 7B	0.1773	0.1874	0.1846	0.1819	0.1789	0.1765
TowerBase Llama 2 7B	0.1773	0.1849	0.1881	0.1867	0.1810	0.1761
TowerInstruct Llama 2 7B	0.1729	0.2017	0.1960	0.1808	0.1740	0.1709
Occiglot Mistral 7B v0.1	0.2226	0.2464	0.2291	0.2299	0.2233	0.2215
Occiglot Mistral 7B v0.1 Instruct	0.1876	0.2430	0.2090	0.1941	0.1918	0.1848
BLOOM 7B	0.1476	0.1558	0.1489	0.1456	0.1511	0.1473
BLOOMZ 7B	0.1696	0.1693	0.1698	0.1696	0.1699	0.1695
mGPT	0.1072	0.0867	0.0844	0.0992	0.1029	0.1093
mGPT-13B	0.1723	0.1798	0.1644	0.1588	0.1610	0.1738
XGLM 7.5B	0.2041	0.2421	0.2369	0.2125	0.2105	0.2010
Yayi 7B	0.1612	0.1665	0.1638	0.1645	0.1583	0.1611
Llama 3 8B	0.2173	0.3184	0.2708	0.2560	0.2261	0.2105
Llama 3.1 8B	0.2020	0.2751	0.2521	0.2443	0.2097	0.1959
Gemma 2 9B	0.1805	0.2868	0.2855	0.2499	0.2028	0.1689
Gemma 7B	0.1383	0.2429	0.2413	0.1874	0.1497	0.1283
Qwen 1.5 7B	0.0729	0.1265	0.1145	0.0878	0.0730	0.0692
Qwen 2 7B	0.2187	0.2737	0.2557	0.2401	0.2233	0.2147
EMMA-500 Llama 2 7B	0.1982	0.2366	0.2333	0.2277	0.2200	0.1930

Table 11: Results on Aya (BLEU/Self-BLEU). EMMA-500 Llama 2 7B has higher average BLEU scores than all baselines.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	1.24/0.74	1.27/0.57	1.47/0.60	0.86/0.37	1.17/0.45	0.77/1.87
Llama 2 7B Chat	1.17/1.29	1.46/1.15	1.36/1.15	0.69/1.03	1.14/1.14	0.54/2.23
CodeLlama 2 7B	1.22/1.21	1.31/1.19	1.40/1.00	0.85/0.84	1.23/0.56	0.78/2.57
LLaMAX Llama 2 7B	1.72/1.70	1.80/1.37	2.03/1.48	1.30/1.24	1.65/0.89	0.98/3.74
LLaMAX Llama 2 7B Alpaca	1.68/1.67	1.82/1.22	1.96/1.42	1.28/1.14	1.55/1.04	1.00/3.94
MaLA-500 Llama 2 10B v1	0.40/2.29	0.42/2.53	0.49/2.79	0.32/1.22	0.31/2.42	0.18/1.16
MaLA-500 Llama 2 10B v2	0.41/2.31	0.42/2.01	0.50/2.65	0.32/1.42	0.33/1.02	0.18/3.09
Yayi Llama 2 7B	1.65/0.61	1.84/0.82	1.88/0.62	1.26/0.62	1.53/0.45	1.02/0.41
TowerBase Llama 2 7B	1.44/0.83	1.45/0.64	1.67/0.56	1.19/0.49	1.46/0.38	0.88/2.54
TowerInstruct Llama 2 7B	1.55/0.93	1.80/0.85	1.82/0.78	1.15/0.65	1.21/0.54	0.85/1.97
Occiglot Mistral 7B v0.1	1.53/2.43	1.57/0.91	1.78/2.89	1.17/1.73	1.61/1.31	0.95/4.27
Occiglot Mistral 7B v0.1 Instruct	0.75/2.81	0.82/2.38	0.89/3.10	0.43/2.72	0.79/1.17	0.45/3.50
BLOOM 7B	0.85/1.17	0.92/1.20	0.96/1.32	0.73/1.05	0.78/1.27	0.55/0.72
BLOOMZ 7B	0.12/0.61	0.07/0.33	0.17/0.62	0.08/1.00	0.06/0.89	0.08/0.51
mGPT	1.24/0.55	1.22/0.64	1.47/0.59	0.91/0.48	1.21/0.60	0.84/0.29
mGPT-13B	1.42/0.57	1.42/0.80	1.63/0.58	1.00/0.48	1.53/0.44	1.03/0.36
Yayi 7B	1.05/0.39	1.22/0.38	1.18/0.41	0.76/0.42	1.01/0.54	0.67/0.22
Llama 3 8B	1.59/0.94	1.03/0.60	1.31/0.53	2.96/0.54	1.11/0.28	2.43/3.47
Llama 3.1 8B	1.85/1.08	1.41/0.95	1.60/0.64	3.11/0.52	1.33/0.39	2.52/3.52
Gemma 2 9B	1.55/0.82	1.59/0.94	1.73/0.93	1.38/0.70	1.33/0.47	1.21/0.65
Gemma 7B	1.29/0.57	1.41/0.62	1.39/0.66	1.16/0.40	1.26/0.50	0.93/0.40
Qwen 1.5 7B	1.93/1.13	1.66/0.85	1.64/0.65	3.14/0.79	1.51/0.58	2.45/3.69
Qwen 2 7B	1.99/1.13	1.84/0.94	1.69/0.70	3.29/0.72	1.36/0.44	2.47/3.53
EMMA-500 Llama 2 7B	2.93/1.54	2.90/1.29	2.75/0.83	3.79/0.82	2.86/0.95	2.87/5.09

in performance primarily in medium-low resource settings, EMMA-500 maintains a consistently high performance across varying levels of resource availability.

PolyWrite This is a novel multilingual benchmark composed in this work for evaluating open-ended generation in 240 languages. We use ChatGPT to generate different prompts in English and use Google Translate to translate them into different languages for models to generate creative content. This benchmark consists of 31 writing tasks, such as storytelling and email writing, and 155 prompts in total. We back-translate the multilingual prompts to English, calculate the BLEU scores between original English prompts and back-translation, and filter out translated prompts with BLEU scores below 20, and the entire dataset contains a total of 35,751 prompts. We release the PolyWrite dataset on Huggingface[22]. The details of PolyWrite are described in Section C.1.

We use Self-BLEU (Zhu et al., 2018) to evaluate the diversity of generated texts in the PolyWrite benchmark, as presented in Table 12. A lower Self-BLEU score indicates more diverse generation, but does not means a better generation quality. Our EMMA-500 model demonstrates comparable performance across various languages. Compared to other models like Llama 3/3.1 and Qwen 1.5/2, particularly in medium-low and low-resource languages, EMMA-500 has higher Self-BLEU scores, indicating lower diversity in its generated content.

Evaluating open-ended generation poses significant challenges, as it goes beyond simply measuring accuracy or correctness. Metrics like BLEU or Self-BLEU, while useful for assessing similarity to reference texts or the diversity of given texts, often fail to capture more nuanced aspects. Subjective factors like cultural relevance and the appropriateness of responses in low-resource languages are difficult to quantify. This makes it challenging to create evaluation benchmarks and metrics that fully capture the strengths and weaknesses of models like EMMA-500 in diverse, real-world scenarios.

4.7 Summarization

XL-Sum (Hasan et al., 2021) is a large-scale multilingual abstractive summarization dataset that covers 44 languages. To evaluate the quality of the generated summaries, we use ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2019) as evaluation metrics. ROUGE-L measures the longest common subsequence (LCS) between the reference and generated summaries. Recall and precision are calculated by dividing the LCS length by the reference length and the generated summary length, respectively. An F-score is then used to combine these two aspects into a single metric. BERTScore

[22] https://huggingface.co/datasets/MaLA-LM/PolyWrite

Table 12: Results on PolyWrite (Self-BLEU).

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	0.5358	0.4282	0.6545	0.3769	0.4766	0.6587
Llama 2 7B Chat	1.1550	0.8640	0.8902	1.1877	1.4435	1.4167
CodeLlama 2 7B	1.0313	1.2052	1.2883	0.9191	0.9092	0.6798
LLaMAX Llama 2 7B	0.9564	1.0066	1.1655	0.8786	0.8363	0.7709
LLaMAX Llama 2 7B Alpaca	0.8086	0.7321	0.9517	0.8322	0.8351	0.4981
MaLA-500 Llama 2 10B v1	3.7079	3.7902	3.5066	2.4541	4.8052	3.5163
MaLA-500 Llama 2 10B v2	4.0059	3.1737	4.0148	3.0476	5.0652	3.8916
Yayi Llama 2 7B	0.6274	0.6207	0.6921	0.4169	0.6813	0.6352
TowerBase Llama 2 7B	0.4938	0.7268	0.4736	0.3945	0.4417	0.5396
TowerInstruct Llama 2 7B	0.7124	0.9565	0.7651	0.6492	0.5998	0.6615
Occiglot Mistral 7B v0.1	0.9647	0.8818	0.7975	0.9543	0.9563	1.4157
Occiglot Mistral 7B v0.1 Instruct	3.9033	5.3884	4.7140	3.7555	2.8444	2.9568
BLOOM 7B	1.3892	1.2845	1.6705	1.9685	1.1513	0.6600
BLOOMZ 7B	0.0024	0.0000	0.0005	0.0093	0.0000	0.0049
mGPT	0.7560	0.9222	0.6291	0.4534	0.8156	1.1134
mGPT-13B	0.7479	0.8483	0.7091	0.4230	0.7962	1.0201
Yayi 7B	0.5151	0.4266	0.3574	0.3283	0.6706	0.8574
Llama 3 8B	0.5796	0.5753	0.5921	0.7141	0.5586	0.4440
Llama 3.1 8B	0.7995	0.6805	0.8253	0.5044	0.6965	1.3585
Gemma 2 9B	1.1736	1.2347	1.2913	1.1616	0.9261	1.3240
Gemma 7B	1.0541	0.9629	1.1222	0.9275	1.1112	1.0284
Qwen 1.5 7B	0.6441	0.9398	0.7457	0.3797	0.4164	0.8738
Qwen 2 7B	0.5709	0.7763	0.6135	0.3695	0.4186	0.7989
EMMA-500 Llama 2 7B	0.9879	0.9833	0.7058	0.8465	1.3130	1.1702

computes the semantic similarity between summaries by leveraging contextual embeddings from pre-trained language models. Specifically, we use the `bert-base-multilingual-cased`[23] model for BERTScore to accommodate multiple languages.

The evaluation results are presented in Table 13, we observe that our model, EMMA-500, performs comparably to other Llama2-based models like TowerInstruct Llama 2 7B. While EMMA-500 shows strength in certain language categories, it does not significantly outperform these models overall. This suggests that, although EMMA-500 is effective in multilingual summarization tasks, there is room for improvement to achieve more consistent performance across all languages. It is also important to note that this test set includes only 44 languages, limiting the evaluation of EMMA-500's capabilities in low-resource languages.

4.8 Natural Language Inference

We evaluate on the XNLI (Cross-lingual Natural Language Inference) benchmark (Conneau et al., 2018) where sentence pairs in different languages need to be classified as entailment, contradiction, or neutral. We categorize the languages in XNLI into 3 groups, i.e., high-resource (German, English, Spanish, French, Russian, and Chinese), medium-resource (Arabic, Bulgarian, Greek, Hindi, Turkish, and Vietnamese), and low-resource (Swahili, Thai, and Urdu).[24] Table 14 shows the aggregated accuracy. Our model outperforms most baselines including Llama 2-based models and multilingual LLMs. We achieve the second-best average accuracy, slightly behind the Llama 3.1 8B model. On the low-resource end, we perform the second-best, slightly behind the Gemma 2 9B model.

4.9 Math

We evaluate the ability of LLMs to solve grade-school math problems across multiple languages on the MGSM (Multilingual Grade School Math) benchmark (Shi et al., 2022). MGSM is an extension of the GSM8K (Grade School Math 8K) dataset (Cobbe et al., 2021) by translating 250 of the original GSM8K problems into ten languages. We also split these ten languages into three groups, i.e., high-resource (Spanish, French, German, Russian, Chinese, and Japanese), medium-resource (Thai, Swahili, and Bengali), and low-resource (Telugu). Table 15 shows the results for 3-shot prompting with a flexible match to obtain the answers in model generation. We evaluate all the models by directly prompting the questions (denoted as direct) and the questions with answers followed by Chain-of-Thoughts

[23] `https://huggingface.co/google-bert/bert-base-multilingual-cased`

[24] Again, the categorization is not perfect.

Table 13: Results on XL-Sum (ROUGE-L/BERTScore).

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	0.07/0.67	0.07/0.67	0.07/0.66	0.06/0.65	0.10/0.62	0.08/0.71
Llama 2 7B Chat	0.09/0.68	0.09/0.68	0.08/0.68	0.08/0.67	0.12/0.64	0.10/0.74
CodeLlama 2 7B	0.07/0.66	0.07/0.65	0.07/0.66	0.06/0.63	0.10/0.64	0.08/0.71
LLaMAX Llama 2 7B	0.05/0.65	0.05/0.65	0.05/0.65	0.05/0.61	0.07/0.62	0.06/0.69
LLaMAX Llama 2 7B Alpaca	0.10/0.69	0.10/0.70	0.09/0.69	0.09/0.68	0.13/0.65	0.12/0.74
MaLA-500 Llama 2 10B v1	0.05/0.64	0.06/0.64	0.05/0.64	0.05/0.61	0.07/0.64	0.06/0.68
MaLA-500 Llama 2 10B v2	0.05/0.64	0.06/0.64	0.05/0.64	0.05/0.61	0.07/0.64	0.06/0.69
Yayi Llama 2 7B	0.08/0.67	0.09/0.67	0.07/0.67	0.07/0.66	0.11/0.61	0.09/0.71
TowerBase Llama 2 7B	0.08/0.67	0.08/0.67	0.07/0.67	0.07/0.65	0.10/0.63	0.09/0.71
TowerInstruct Llama 2 7B	0.09/0.68	0.09/0.69	0.09/0.69	0.08/0.66	0.12/0.64	0.09/0.73
Occiglot Mistral 7B v0.1	0.07/0.66	0.08/0.66	0.07/0.66	0.07/0.64	0.10/0.63	0.09/0.71
Occiglot Mistral 7B v0.1 Instruct	0.08/0.67	0.09/0.67	0.08/0.67	0.08/0.64	0.10/0.64	0.09/0.73
BLOOM 7B	0.07/0.65	0.07/0.65	0.07/0.65	0.06/0.62	0.10/0.58	0.08/0.71
BLOOMZ 7B	0.11/0.70	0.14/0.69	0.09/0.70	0.10/0.69	0.12/0.66	0.15/0.74
mGPT	0.04/0.60	0.05/0.60	0.04/0.60	0.02/0.56	0.07/0.59	0.04/0.63
mGPT-13B	0.05/0.62	0.06/0.62	0.05/0.63	0.04/0.59	0.08/0.59	0.05/0.67
Yayi 7B	0.12/0.70	0.14/0.69	0.10/0.69	0.11/0.69	0.12/0.66	0.15/0.75
Llama 3 8B	0.08/0.67	0.08/0.67	0.08/0.67	0.08/0.65	0.10/0.65	0.11/0.72
Llama 3.1 8B	0.09/0.67	0.08/0.66	0.08/0.67	0.08/0.65	0.09/0.66	0.10/0.72
Gemma 2 9B	0.07/0.65	0.07/0.64	0.07/0.66	0.07/0.63	0.09/0.64	0.09/0.71
Gemma 7B	0.07/0.65	0.07/0.63	0.06/0.65	0.07/0.67	0.07/0.60	0.08/0.63
Qwen 1.5 7B	0.10/0.69	0.11/0.69	0.09/0.69	0.10/0.68	0.11/0.65	0.11/0.74
Qwen 2 7B	0.10/0.69	0.12/0.69	0.09/0.69	0.10/0.68	0.11/0.65	0.11/0.74
EMMA-500 Llama 2 7B	0.09/0.67	0.08/0.66	0.08/0.67	0.08/0.69	0.11/0.66	0.10/0.67

Table 14: 0-shot results on XNLI (ACC).

Model	Avg	High	Medium	Low
Llama 2 7B	0.4019	0.4526	0.3772	0.3497
Llama 2 7B Chat	0.3858	0.4277	0.3675	0.3387
CodeLlama 2 7B	0.4019	0.4627	0.3729	0.3386
LLaMAX Llama 2 7B	0.4427	0.4653	0.4264	0.4303
LLaMAX Llama 2 7B Alpaca	0.4509	0.4847	0.4280	0.4289
MaLA-500 Llama 2 10B v1	0.3811	0.4210	0.3585	0.3465
MaLA-500 Llama 2 10B v2	0.3811	0.4210	0.3585	0.3465
YaYi Llama 2 7B	0.4128	0.4732	0.3841	0.3494
TowerBase Llama 2 7B	0.3984	0.4608	0.3633	0.3439
TowerInstruct Llama 2 7B	0.4036	0.4707	0.3692	0.3379
Occiglot Mistral 7B v0.1	0.4235	0.4990	0.3839	0.3519
Occiglot Mistral 7B v0.1 Instruct	0.4081	0.4758	0.3718	0.3452
BLOOM 7B	0.4160	0.4513	0.3969	0.3838
BLOOMZ 7B	0.3713	0.4002	0.3556	0.3451
mGPT	0.4051	0.4297	0.3965	0.3730
XGLM 7.5B	0.4375	0.4572	0.4216	0.4300
YaYi 7B	0.3987	0.4385	0.3824	0.3515
Llama 3 8B	0.4497	0.4882	0.4384	0.3956
Llama 3.1 8B	0.4562	0.4961	0.4404	0.4083
Gemma 7B	0.4258	0.4644	0.4100	0.3801
Gemma 2 9B	0.4674	0.4850	0.4511	0.4649
Qwen 2 7B	0.4277	0.4731	0.4135	0.3653
Qwen 1.5 7B	0.3947	0.4095	0.3880	0.3783
EMMA-500 Llama 2 7B	0.4514	0.4609	0.4440	0.4471

prompt in the same languages as the subset being evaluated (denoted as CoT) (Shi et al., 2022). The results show that Llama 2 7B is a weak model in the MGSM math task. The base model and its variants failed in most settings. Multilingual LLMs such as BLOOM, XGLM, and YaYi are also subpar at this task. Recent LLMs like Llama 3, Qwen, and Gemma obtain reasonable performance, and Qwen series models are the best. Our model improves the Llama 2 7B model remarkably in both prompt strategies. For direct prompting, our model has an average accuracy of 0.1702, 7% higher than the Llama 2 7B chat model. For CoT-based prompting, our model increases the score of the Llama 2 7B chat model from 0.1 to 0.3 (20% higher) and slightly outperforms Llama 3 and 3.1 8B models.

Table 15: 3-shot results (ACC) on MGSM obtained by direct and CoT prompting.

Model	Direct Prompting				CoT Prompting			
	Avg	High	Medium	Low	Avg	High	Medium	Low
Llama 2 7B	0.0669	0.0807	0.0213	0.0120	0.0636	0.0760	0.0213	0.0080
Llama 2 7B Chat	0.1022	0.1373	0.0213	0.0080	0.1091	0.1353	0.0280	0.0160
CodeLlama 2 7B	0.0593	0.0707	0.0293	0.0120	0.0664	0.0873	0.0267	0.0200
LLaMAX Llama 2 7B	0.0335	0.0400	0.0200	0.0080	0.0362	0.0433	0.0227	0.0240
LLaMAX Llama 2 7B Alpaca	0.0505	0.0520	0.0400	0.0160	0.0635	0.0807	0.0413	0.0080
MaLA-500 Llama 2 10B v1	0.0091	0.0133	0.0027	0.0000	0.0073	0.0127	0.0000	0.0000
MaLA-500 Llama 2 10B v2	0.0091	0.0133	0.0027	0.0000	0.0073	0.0127	0.0000	0.0000
TowerBase Llama 2 7B	0.0615	0.0833	0.0173	0.0080	0.0616	0.0860	0.0240	0.0080
TowerInstruct Llama 2 7B	0.0724	0.0953	0.0173	0.0200	0.0824	0.1047	0.0187	0.0120
Occiglot Mistral 7B v0.1	0.1331	0.1687	0.0453	0.0160	0.1407	0.1880	0.0360	0.0120
Occiglot Mistral 7B v0.1 Instruct	0.2276	0.2980	0.0747	0.0280	0.2216	0.3040	0.0787	0.0280
BLOOM 7B	0.0287	0.0260	0.0280	0.0360	0.0229	0.0220	0.0147	0.0200
BLOOMZ 7B	0.0255	0.0267	0.0240	0.0120	0.0215	0.0167	0.0307	0.0200
mGPT	0.0135	0.0167	0.0053	0.0000	0.0142	0.0193	0.0067	0.0000
mGPT 13B	0.0131	0.0180	0.0067	0.0000	0.0153	0.0167	0.0120	0.0000
XGLM 7.5B	0.0102	0.0120	0.0067	0.0200	0.0116	0.0107	0.0120	0.0280
Yayi 7B	0.0276	0.0293	0.0147	0.0240	0.0302	0.0293	0.0240	0.0200
Llama 3 8B	0.2745	0.2787	0.2613	0.0560	0.2813	0.2853	0.2667	0.0520
Llama 3.1 8B	0.2836	0.2900	0.2613	0.0440	0.2731	0.2727	0.2547	0.0840
Gemma 7B	0.3822	0.3660	0.3827	0.2720	0.3578	0.3467	0.3707	0.2680
Gemma 2 9B	0.3295	0.2800	0.3573	0.3080	0.4469	0.3607	0.5200	0.4720
Qwen 2 7B	0.4895	0.5440	0.3880	0.1480	0.4469	0.3607	0.5200	0.4720
Qwen 1.5 7B	0.3156	0.4000	0.1600	0.0400	0.5147	0.5893	0.3907	0.1440
EMMA-500 Llama 2 7B	0.1702	0.1920	0.1187	0.0240	0.3036	0.4060	0.1480	0.0240

4.10 Machine Reading Comprehension

BELEBELE (Bandarkar et al., 2023) is a machine reading comprehension dataset covering 122 languages including high- and low-resource languages. Each question offers four multiple-choice answers based on a short passage from the FLORES-200 dataset. This benchmark is very challenging, even the English version of it presents remarkable challenges for advanced models. Table 16 shows the zero-shot results in different resource groups. Our continual pre-training improves the Llama 2 7B base model. But Llama 2 7 B-based models mostly fail in this task and get quasi-random results. Recent advanced models like Llama 3.1 and Qwen 2 get reasonable results.

Table 16: 0-shot results (ACC) on BELEBELE.

Model	Avg	High	Medium-High	Medium	Medium-Low	Low
Llama 2 7B	0.2627	0.2676	0.2635	0.2607	0.2641	0.2627
Llama 2 7B Chat	0.2905	0.3184	0.2997	0.2895	0.2947	0.2909
CodeLlama 2 7B	0.2738	0.2737	0.2733	0.2730	0.2730	0.2738
LLaMAX Llama 2 7B	0.2309	0.2323	0.2315	0.2307	0.2310	0.2308
LLaMAX Llama 2 7B Alpaca	0.2448	0.2546	0.2482	0.2441	0.2460	0.2449
MaLA-500 Llama 2 10B v1	0.2296	0.2302	0.2298	0.2297	0.2298	0.2297
MaLA-500 Llama 2 10B v2	0.2296	0.2302	0.2298	0.2297	0.2298	0.2297
YaYi Llama 2 7B	0.2832	0.2964	0.2867	0.2811	0.2837	0.2826
TowerBase Llama 2 7B	0.2636	0.2743	0.2685	0.2629	0.2648	0.2634
TowerInstruct Llama 2 7B	0.2793	0.2988	0.2851	0.2757	0.2819	0.2792
Occiglot Mistral 7B v0.1	0.3016	0.3225	0.3094	0.3002	0.3040	0.3015
Occiglot Mistral 7B v0.1 Instruct	0.3205	0.3414	0.3262	0.3174	0.3240	0.3208
BLOOM 7B	0.2411	0.2425	0.2452	0.2412	0.2411	0.2408
BLOOMZ 7B	0.3932	0.4543	0.4367	0.4151	0.4008	0.3951
mGPT	0.2396	0.2401	0.2381	0.2400	0.2394	0.2398
XGLM 7.5B	0.2485	0.2552	0.2517	0.2489	0.2490	0.2485
YaYi 7B	0.3797	0.4437	0.4271	0.4049	0.3872	0.3809
Llama 3 8B	0.4073	0.4607	0.4292	0.4103	0.4187	0.4088
Llama 3.1 8B	0.4519	0.5250	0.4801	0.4565	0.4674	0.4534
Gemma 7B	0.4337	0.5263	0.4783	0.4482	0.4543	0.4394
Gemma 2 9B	0.5449	0.6410	0.5890	0.5583	0.5685	0.5505
Qwen 2 7B	0.4931	0.5762	0.5220	0.5004	0.5116	0.4948
Qwen 1.5 7B	0.4183	0.4886	0.4479	0.4218	0.4300	0.4178
EMMA-500 Llama 2 7B	0.2675	0.2832	0.2818	0.2758	0.2714	0.2694

We then move to a more challenging task, the ARC multilingual test, which is a machine-translated benchmark (Lai et al., 2023) from the ARC dataset (Clark et al., 2018) that contains English science

exam questions for multiple grade levels. We test the five-shot performance, with results shown in Table 17. The evaluation results show a similar pattern to BELEBELE that EMMA-500 improves Llama 2 but the 7B model is not capable of this challenging task, all Llama2 7B-based models obtain close to random results, and recent advances like Llama 3, Gemma, and Qwen get much better results.

Table 17: 5-shot results (ACC) on ARC multilingual.

Model	Avg	High	Medium	Low
Llama 2 7B	0.2756	0.3312	0.2731	0.2102
Llama 2 7B Chat	0.2802	0.3369	0.2779	0.2129
CodeLlama 2 7B	0.2523	0.2886	0.2464	0.2165
LLaMAX Llama 2 7B	0.2609	0.3000	0.2592	0.2148
LLaMAX Llama 2 7B Alpaca	0.3106	0.3689	0.3185	0.2249
MaLA-500 Llama 2 10B v1	0.2116	0.2192	0.2048	0.2132
MaLA-500 Llama 2 10B v2	0.2116	0.2192	0.2048	0.2132
YaYi Llama 2 7B	0.2840	0.3430	0.2835	0.2111
TowerBase Llama 2 7B	0.2794	0.3532	0.2682	0.2051
TowerInstruct Llama 2 7B	0.3010	0.3888	0.2885	0.2116
Occiglot Mistral 7B v0.1	0.2977	0.3839	0.2851	0.2103
Occiglot Mistral 7B v0.1 Instruct	0.3088	0.4029	0.2965	0.2113
BLOOM 7B	0.2365	0.2627	0.2272	0.2189
BLOOMZ 7B	0.2395	0.2694	0.2274	0.2218
mGPT	0.2024	0.2011	0.1965	0.2137
mGPT 13B	0.2176	0.2299	0.2114	0.2123
XGLM 7.5B	0.2221	0.2461	0.2105	0.2110
YaYi 7B	0.2444	0.2796	0.2329	0.2191
Llama 3 8B	0.3480	0.4243	0.3553	0.2406
Llama 3.1 8B	0.3493	0.4243	0.3589	0.2400
Gemma 7B	0.3868	0.4646	0.4047	0.2606
Gemma 2 9B	0.4415	0.5459	0.4618	0.2782
Qwen 2 7B	0.3382	0.4388	0.3264	0.2317
Qwen 1.5 7B	0.2893	0.3555	0.2814	0.2192
EMMA-500 Llama 2 7B	0.2953	0.3410	0.2982	0.2334

4.11 Code Generation

We conduct code generation evaluations on the Multipl-E (Cassano et al., 2022) benchmark in the interest of measuring the effects of massively multilingual continual pre-training on a model's code generation utility and detecting if any catastrophic forgetting (Luo et al., 2023) has occurred on this front. Importantly, this also has implications for a model's reasoning (Yang et al., 2024b) and entity-tracking abilities (Kim et al., 2024).

Table 18 outlines comparisons against strong, openly available multilingual models such as Qwen (Yang et al., 2024a) and Aya (Üstün et al., 2024) along with other continually pre-trained models. Our main takeaway is that by carefully curating high-quality code data as part of the data mix, it is possible to not only avoid the catastrophic forgetting that has plagued existing continually pre-trained models (MaLA-500, LLaMAX and TowerBase) but also surpass the base model itself. However, the pre-training phase is still the most important, and closing the gap with stronger base models like Qwen and Gemma remains elusive.

5 Related Work

Multilingual LLMs Multilingual large language models (LLMs) have made significant progress in processing and understanding multiple languages within a unified framework. Models like mT5 (Xue et al., 2021) and XGLM (Lin et al., 2022) leverage both monolingual and multilingual datasets to perform tasks such as translation and text summarization across a wide spectrum of languages. However, the predominant focus on English has led to disparities in performance, particularly for low-resource languages. Recent work on multilingual LLMs, such as BLOOM (Scao et al., 2022), has shown that adapting these English-centric models through vocabulary extension based on multilingual corpora and continual pre-training (CPT) can improve performance across languages, especially low-resource ones. These models highlight the importance of efficient tokenization and adaptation, which can bridge the performance gap between high-resource and low-resource languages.

Table 18: Results on Multipl-E. For language-level breakdowns, refer to Tables 24 to 26 in the appendix.

Model	Avg Pass@1	Avg Pass@10	Avg Pass@25
Llama 2 7B	8.92%	19.45%	25.68%
CodeLlama 2 7B	28.43%	50.83%	63.92%
LLaMAX 2 7B	0.35%	1.61%	2.67%
MaLA-500 Llama 2 10B V2	0.0%	0.0%	0.0%
TowerBase Llama 2 7B	3.61%	6.65%	8.97%
Occiglot Mistral 7B v0.1	21.26%	31.37%	45.86%
Bloom 7B	5.34%	10.49%	14.65%
BloomZ 2 7B	5.85%	11.40%	15.76%
Aya23 8B	9.19%	23.52%	32.09%
Mistral 7B v0.3	26.10%	48.68%	59.05%
Llama 3 8B	30.09%	53.82%	64.01%
LLaMAX 3 8B	3.00%	7.23%	10.67%
Gemma 7B	28.55%	54.27%	64.75%
CodeGemma 7B	31.51%	63.13%	72.65%
Qwen 1.5 7B	21.05%	37.19%	47.63%
Qwen 2 7B	38.68%	62.63%	71.55%
EMMA-500 Llama 2 7B	11.38%	19.02%	26.16%

Multilingual Corpora The availability and use of multilingual corpora play a crucial role in training multilingual LLMs. CC100 Corpus (Conneau et al., 2020), launched in 2020, encompasses hundreds of billions of tokens and over 100 languages. Further, CC100-XL Corpus (Lin et al., 2022), created for the training of XGLM, extends across 68 Common Crawl Snapshots and 134 languages, aiming to balance language presentation and improve performance in few-shot and zero-shot tasks. The ROOTS Corpus (Laurençon et al., 2022), released in July 2022, supports BLOOM with approximately 341 billion tokens across 46 natural languages. It emphasizes underrepresented languages such as Swahili and Catalan, drawing from diverse sources including web crawls, books and academic publications. Besides, Occiglot Fineweb [25], which began to be released in early 2024, consists of around 230 million documents in 10 European languages. It combines curated and cleaned web data to support efficient training for both high- and low-resource European languages. Additionally, recent efforts parallel corpus construction from web crawls, such as ParaCrawl (Bañón et al., 2020a) and CCMatrix (Schwenk et al., 2021b), have contributed to large-scale multilingual training too.

Continual Pre-training Continual pre-training has become a popular strategy to adapt LLMs to new languages and domains without retraining from scratch. The process involves updating model parameters incrementally using a relatively small amount of new data from target languages. Recent studies, such as those by Tejaswi et al. (2024), have demonstrated the effectiveness of CPT in improving the adaptability of LLMs in low-resource language settings. Techniques such as vocabulary augmentation and targeted domain-specific pre-training have been shown to significantly improve both efficiency and performance, particularly when large multilingual models are adapted to new languages. Despite its benefits, CPT can lead to catastrophic forgetting, where models lose previously learned information. To address the potential degraded performance issue, Ibrahim et al. (2024) presents a simple yet effective approach to continual pre-train models, demonstrating that with a combination of learning rate re-warming, re-decaying, and replay of previous data, it is possible to match the performance of fully re-trained models. Also, recent studies have delved into the effectiveness of continual pre-training with parallel data. As highlighted in the study by Gilabert et al. (2024), the use of a Catalan-centric parallel dataset has enabled the training of models good at trasnalting in various directions. Besides, the research by Kondo et al. (2024), proposed a two-phase continual training approach with parallel data. In the first phase, a pre-trained LLM is continually pre-trained on parallel data, followed by a second phase of supervised fine-tuning with a small amount of high-quality parallel data. Their experiments with a 3.8B-parameter model across various data formats revealed that alternating between source and target sentences during continual pre-training is crucial for enhancing translation accuracy in the corresponding direction.

[25] https://occiglot.eu/posts/occiglot-fineweb/

6 Conclusion

This paper addresses critical advancements and challenges in adapting language models to more than 500 languages, focusing on enhancing their performance across diverse languages. We compile the MaLA corpus, a multilingual dataset for continual pre-training of multilingual language models. By expanding and augmenting existing corpora, we train the EMMA-500 model. It demonstrates notable improvements in a range of tasks such as next-token prediction, commonsense reasoning, machine translation, text classification, and open-ended generation, showing remarkable improvements in low-resource languages. Our results show that a well-curated, massively multilingual corpora can advance model capabilities. This work sets a new benchmark for inclusive and effective multilingual language models and paves the way for future research to address the disparities between high-resource and low-resource languages.

Limitations and Outlooks

Multilingual language models are typically designed to serve users of different languages, which reflect the diverse cultural and linguistic backgrounds of their speakers. However, many available multilingual benchmarks, including some used in this study, have been created through human or machine translation. They tend to feature topics and knowledge primarily from English-speaking communities and carry imperfections due to translation, which affects the integrity of LLM evaluation (Chen et al., 2024). We highlight this discrepancy and call for collaborative efforts to develop large-scale natively-created multilingual test sets that more accurately assess the breadth of languages and cultures these models aim to cater to.

Our EMMA-500 model demonstrates enhanced multilingual performance relative to its base model, Llama 2 7B, as well as other continually pre-trained variants. However, it does not surpass some of the latest models, such as Llama 3, Gemma 2, and Qwen 2, on certain benchmarks. These models are likely to be a better starting point for continual pre-training compared to Llama 2 due to their larger pre-training data. We intend to pursue multilingual extension based on these newer models in the future. On the other hand, it would be practical to assess the released corpus directly in pre-training from scratch. Moreover, multilingual instruction tuning to prepare these models for better task performance and natural interactions would be useful.

Acknowledgment

The work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070350 and from UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10052546], and funding from UTTER's Financial Support for Third Parties under the EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631). The authors wish to acknowledge CSC – IT Center for Science, Finland, the Leonardo and LUMI supercomputers, owned by the EuroHPC Joint Undertaking, for providing computational resources.

References

Solomon Teferra Abate, Michael Melese, Martha Yifiru Tachbelie, Million Meshesha, Solomon Atinafu, Wondwossen Mulugeta, Yaregal Assabie, Hafte Abera, Binyam Ephrem, Tewodros Abebe, Wondimagegnhue Tsegaye, Amanuel Lemma, Tsegaye Andargie, and Seifedin Shifaw. 2018. Parallel corpora for bi-lingual English-Ethiopian languages statistical machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*.

Ahmed Abdelali, Hamdy Mubarak, Younes Samih, Sabit Hassan, and Kareem Darwish. 2021. QADI: Arabic dialect identification in the wild. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. Shami: A corpus of Levantine Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

David Adelani, Dana Ruiter, Jesujoba Alabi, Damilola Adebonojo, Adesina Ayeni, Mofe Adeyemi, Ayodele Esther Awokoya, and Cristina España-Bonet. 2021. The effect of domain and diacritics in Yoruba–English neural machine translation. In *Proceedings of Machine Translation Summit XVIII: Research Track*.

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2023. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. *CoRR*.

Rodrigo Agerri, Xavier Gómez Guinovart, German Rigau, and Miguel Anxo Solla Portela. 2018. Developing new linguistic resources and tools for the Galician language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Israa Alsarsour, Esraa Mohamed, Reem Suwaileh, and Tamer Elsayed. 2018. DART: A large dataset of dialectal Arabic tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*.

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Jason Phang, Shivanshu Purohit, Hailey Schoelkopf, Dashiell Stander, Tri Songz, Curt Tigges, Benjamin Thérien, Phil Wang, and Samuel Weinbach. 2023. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch.

Mikko Aulamo, Nikolay Bogoychev, Shaoxiong Ji, Graeme Nail, Gema Ramírez-Sánchez, Jörg Tiedemann, Jelmer Van Der Linde, and Jaume Zaragoza. 2023. Hplt: High performance language technologies. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*.

Niyati Bafna. 2022. Empirical models for an indic language continuum.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The BELEBELE benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. 2020a. Paracrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020b. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Marta Bañón, Miquel Esplà-Gomis, Mikel L. Forcada, Cristian García-Romero, Taja Kuzman, Nikola Ljubesic, Rik van Noord, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Peter Rupnik, Vít Suchomel, Antonio Toral, Tobias van der Werff, and Jaume Zaragoza. 2022. Macocu: Massive collection and curation of monolingual and bilingual data: focus on under-resourced languages. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation, EAMT 2022, Ghent, Belgium, June 1-3, 2022*.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*.

José Camacho-Collados, Claudio Delli Bovi, Alessandro Raganato, and Roberto Navigli. 2016. A large-scale multilingual disambiguation of glosses. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, et al. 2022. Multipl-e: A scalable and extensible approach to benchmarking neural code generation. *arXiv preprint arXiv:2208.08227*.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2023. When is multilinguality a curse? Language modeling for 250 high- and low-resource languages. *arXiv preprint*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Pinzhen Chen, Simon Yu, Zhicheng Guo, and Barry Haddow. 2024. Is it good data for multilingual instruction tuning or just bad multilingual evaluation for large language models? *arXiv preprint arXiv:2406.12822*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard 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*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Corpora and Tools. n.d. Languages of Russia: Collections of texts in small languages.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, et al. 2024. A new massive multilingual dataset for high-performance language technologies. In *Proceedings of LREC-COLING*.

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*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jonathan Dunn. 2020. Mapping languages: the corpus of global language use. *Lang. Resour. Evaluation*.

EdTeKLA. 2022. Indigenous languages corpora.

Mahmoud El-Haj. 2020. Habibi - a multi dialect multi national Arabic song lyrics corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*.

Mahmoud El-Haj, Paul Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Manuel Faysse. 2023. Dataset card for "project gutenberg".

Fitsum Gaim, Wonsuk Yang, and Jong C. Park. 2021. Tlmd: Tigrinya language modeling dataset (1.0.0). Dataset.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. 2023. A framework for few-shot language model evaluation. Zenodo.

Javier García Gilabert, Carlos Escolano, Aleix Sant Savall, Francesca De Luca Fornaciari, Audrey Mash, Xixian Liao, and Maite Melero. 2024. Investigating the translation capabilities of large language models trained on parallel data only. *Preprint*, arXiv:2406.09140.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*.

Santiago Góngora, Nicolás Giossa, and Luis Chiruzzo. 2021. Experiments on a Guarani corpus of news and social media. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*.

Santiago Góngora, Nicolás Giossa, and Luis Chiruzzo. 2022. Can we use word embeddings for enhancing Guarani-Spanish machine translation? In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*.

Thamme Gowda, Zhao Zhang, Chris Mattmann, and Jonathan May. 2021. Many-to-English machine translation tools, data, and pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*.

Grupo de Inteligencia Artificial PUCP. n.d. Monolingual and parallel corpora of peruvian languages.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. 2024. Simple and scalable strategies to continually pre-train large language models. *Preprint*, arXiv:2403.08763.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. 2021. Masakhaner: Named entity recognition for african languages. *arXiv e-prints*.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André F. T. Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. *arXiv preprint*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

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

Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schütze. 2023. GlotLID: Language identification for low-resource languages. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Amir Hossein Kargaran, François Yvon, and Hinrich Schütze. 2024. Glotscript: A resource and tool for low resource writing system identification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*.

Najoung Kim, Sebastian Schuster, and Shubham Toshniwal. 2024. Code pretraining improves entity tracking abilities of language models. *CoRR*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2023. The stack: 3 TB of permissively licensed source code. *Trans. Mach. Learn. Res.*

Minato Kondo, Takehito Utsuro, and Masaaki Nagata. 2024. Enhancing translation accuracy of large language models through continual pre-training on parallel data. *Preprint*, arXiv:2407.03145.

Fajri Koto and Ikhwan Koto. 2020. Towards computational linguistics in Minangkabau language: Studies on sentiment analysis and machine translation. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. 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*.

Wen Lai, Mohsen Mesgar, and Alexander Fraser. 2024. LLMs beyond English: Scaling the multilingual capability of LLMs with cross-lingual feedback. In *Findings of the Association for Computational Linguistics ACL 2024*.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Colin Leong, Joshua Nemecek, Jacob Mansdorfer, Anna Filighera, Abraham Owodunni, and Daniel Whitenack. 2022. Bloom library: Multimodal datasets in 300+ languages for a variety of downstream tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Shenggui Li, Hongxin Liu, Zhengda Bian, Jiarui Fang, Haichen Huang, Yuliang Liu, Boxiang Wang, and Yang You. 2023. Colossal-ai: A unified deep learning system for large-scale parallel training. In *Proceedings of the 52nd International Conference on Parallel Processing*.

Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. 2022. CSL: A large-scale chinese scientific literature dataset. In *Proceedings of the 29th International Conference on Computational Linguistics*.

Evenki Life. 2014. Evenki life newspaper.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André FT Martins, and Hinrich Schütze. 2024. MaLA-500: Massive language adaptation of large language models. *arXiv preprint arXiv:2401.13303*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. 2024. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*.

Yinquan Lu, Wenhao Zhu, Lei Li, Yu Qiao, and Fei Yuan. 2024. LLaMAX: Scaling linguistic horizons of LLM by enhancing translation capabilities beyond 100 languages. *arXiv preprint arXiv:2407.05975*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *CoRR*.

Chunlan Ma, Ayyoob ImaniGooghari, Haotian Ye, Ehsaneddin Asgari, and Hinrich Schütze. 2023. Taxi1500: A multilingual dataset for text classification in 1500 languages. *Preprint*, arXiv:2305.08487.

Yingwei Ma, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. 2024. At which training stage does code data help llms reasoning? In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Martin Majliš. 2011. W2C – web to corpus – corpora. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Masakhane. 2023. Lacuna project.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*.

Jamshidbek Mirzakhalov, Anoop Babu, Duygu Ataman, Sherzod Kariev, Francis Tyers, Otabek Abduraufov, Mammad Hajili, Sardana Ivanova, Abror Khaytbaev, Antonio Laverghetta Jr., Bekhzodbek Moydinboyev, Esra Onal, Shaxnoza Pulatova, Ahsan Wahab, Orhan Firat, and Sriram Chellappan. 2021. A large-scale study of machine translation in Turkic languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Steven Moran, Christian Bentz, Ximena Gutierrez-Vasques, Olga Pelloni, and Tanja Samardzic. 2022. TeDDi sample: Text data diversity sample for language comparison and multilingual NLP. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*.

Chenghao Mou, Chris Ha, Kenneth Enevoldsen, and Peiyuan Liu. 2023. Chenghaomou/text-dedup: Reference snapshot.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Jonathan Mukiibi, Andrew Katumba, Joyce Nakatumba-Nabende, Ali Hussein, and Joshua Meyer. 2022. The makerere radio speech corpus: A luganda radio corpus for automatic speech recognition. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.

Taishi Nakamura, Mayank Mishra, Simone Tedeschi, Yekun Chai, Jason T Stillerman, Felix Friedrich, Prateek Yadav, Tanmay Laud, Vu Minh Chien, Terry Yue Zhuo, et al. 2024. Aurora-m: The first open source multilingual language model red-teamed according to the us executive order. *arXiv preprint arXiv:2404.00399*.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Raj Dabre, Shohei Higashiyama, Shantipriya Parida, Anoop Kunchukuttan, Makoto Morishita, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2022. Overview of the 9th workshop on Asian translation. In *Proceedings of the 9th Workshop on Asian Translation*.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *Preprint*, arXiv:2309.09400.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*.

OSCAR. 2023. Oscar (open super-large crawled aggregated corpus) 2301.

Chester Palen-Michel, June Kim, and Constantine Lignos. 2022. Multilingual open text release 1: Public domain news in 44 languages. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.

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*.

Indraneil Paul, Goran Glavas, and Iryna Gurevych. 2024. Ircoder: Intermediate representations make language models robust multilingual code generators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

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*.

Anand Rajaraman and Jeffrey D Ullman. 2011. *Mining of massive datasets*. Autoedicion.

Ronald Rivest. 1992. The md5 message-digest algorithm.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*.

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*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. 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)*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *Preprint*, arXiv:2210.03057.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mGPT: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

Anil Kumar Singh. 2008. Named entity recognition for south and south East Asian languages: Taking stock. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, et al. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Luca Soldaini and Kyle Lo. 2023. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI. ODC-By, `https://github.com/allenai/pes2o`.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*.

Marc Szafraniec, Baptiste Rozière, Hugh Leather, Patrick Labatut, François Charton, and Gabriel Synnaeve. 2023. Code translation with compiler representations. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Atula Tejaswi, Nilesh Gupta, and Eunsol Choi. 2024. Exploring design choices for building language-specific llms. *Preprint*, arXiv:2406.14670.

Huu Nguyen Thuat Nguyen and Thien Nguyen. 2024. Culturay: A large cleaned multilingual dataset of 75 languages.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*.

Jörg Tiedemann. 2020. The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*.

Alexey Tikhonov and Max Ryabinin. 2021. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

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*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*.

Wikimedia Foundation. n.d. Wikimedia downloads.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, X. Li, Zhi Yuan Lim, S. Soleman, R. Mahendra, Pascale Fung, Syafri Bahar, and A. Purwarianti. 2020. Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint*.

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*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Ke Yang, Jiateng Liu, John Wu, Chaoqi Yang, Yi R. Fung, Sha Li, Zixuan Huang, Xu Cao, Xingyao Wang, Yiquan Wang, Heng Ji, and Chengxiang Zhai. 2024b. If LLM is the wizard, then code is the wand: A survey on how code empowers large language models to serve as intelligent agents. *CoRR*.

Rodolfo Zevallos, John Ortega, William Chen, Richard Castro, Núria Bel, Cesar Toshio, Renzo Venturas, Hilario Aradiel, and Nelsi Melgarejo. 2022. Introducing QuBERT: A large monolingual corpus and BERT model for Southern Quechua. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*.

Chen Zhang, Mingxu Tao, Quzhe Huang, Jiuheng Lin, Zhibin Chen, and Yansong Feng. 2024a. Mc^2: Towards transparent and culturally-aware nlp for minority languages in china. *arXiv preprint arXiv:2311.08348*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xinlu Zhang, Zhiyu Zoey Chen, Xi Ye, Xianjun Yang, Lichang Chen, William Yang Wang, and Linda Ruth Petzold. 2024b. Unveiling the impact of coding data instruction fine-tuning on large language models reasoning. *CoRR*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*.

A Data Sources

A.1 Monolingual Data

Table 19 lists the corpora and collections we use as monolingual data sources in this work. Monolingual data sources simply contain text data in a single language.

Metadata For the monolingual data sources, we define the contents of the JSONL output file from the pre-processing workflow to consist of the fields `url`, `text`, `collection`, `source`, and `original_code`. The contents of these fields are as follows. The field `text` contains the language data in a granularity specific to the given corpus. If the granularity was sentence-level, then we could expect the sentences in the corpus to generally be independent of each other, while only parts of the sentences—such as phrases, clauses, and words—exhibit serial dependence. If the granularity was paragraph-level, then we could expect the sentences within paragraphs to have serial dependence, while paragraphs to largely be independent of each other. The field `url` contains a URL indicating the web address from which the text data has been extracted, if available. The field `collection` contains the name of the collection, i.e., a corpus or a set of corpora, which the text is extracted from, whereas the field `source` contains the name of a more specific part of the collection, such as the name of an individual corpus or a file in the collection the text was extracted from. Lastly, the field `original_code` contains the language code of the text data as it is designated in the data source, e.g., in the directory structure, the filenames, or the data object returned by an API call.

Table 19: Datasets used as monolingual source data.

Name	Languages	Domains	URL	Year
AfriBERTa (Ogueji et al., 2021)	10	news	https://huggingface.co/datasets/castorini/afriberta-corpus	2021
Bloom library (Leong et al., 2022)	363	religious, books	https://huggingface.co/datasets/sil-ai/bloom-lm	2022
CC100 (Conneau et al., 2020)	100	web	https://huggingface.co/datasets/cc100	2020
CulturaX (Nguyen et al., 2023)	167	web	https://huggingface.co/datasets/uonlp/CulturaX	2023
CulturaY (Thuat Nguyen and Nguyen, 2024)	75	web	https://huggingface.co/datasets/ontocord/CulturaY	2024
curse-of-multilinguality (Chang et al., 2023)	200	misc	https://github.com/tylerachang/curse-of-multilinguality	2023
Evenki Life (Life, 2014)	1	newspapers	https://drive.google.com/file/d/1he2q6RncA_NKHPIJjSzlkK-2qgEFTiCG/view	2014
Glot500 (Imani et al., 2023)	511	misc	https://huggingface.co/datasets/cis-lmu/Glot500	2023
GlotSparse (Kargaran et al., 2023)	10	news	https://huggingface.co/datasets/cis-lmu/GlotSparse	2023
HPLT v1.2 (de Gibert et al., 2024)	75	web	https://hplt-project.org/datasets/v1.2	2024
Indigenous Languages Corpora (EdTeKLA, 2022)	1	UNK	https://github.com/EdTeKLA/IndigenousLanguages_Corpora	2022
Indo4B (Wilie et al., 2020)	1	misc	https://github.com/IndoNLP/indonlu?tab=readme-ov-file	2020
Lacuna Project (Masakhane, 2023)	20	UNK	https://github.com/masakhane-io/lacuna_pos_ner	2023
Languages of Russia (Corpora and Tools, n.d.)	46	social media, web	http://web-corpora.net/wsgi3/minorlangs/download	UNK
Madlad-400 (Kudugunta et al., 2024)	419	web	https://huggingface.co/datasets/allenai/MADLAD-400	2023
Makerere Radio Speech Corpus (Mukiibi et al., 2022)	1	transcription	https://zenodo.org/records/5855017	2022
masakhane-ner1.0 (Ifeoluwa Adelani et al., 2021)	12	UNK	https://github.com/masakhane-io/masakhane-ner	2021
MC2 (Zhang et al., 2024a)	4	web	https://huggingface.co/datasets/pkupie/mc2_corpus	2024
mC4 (Raffel et al., 2020)	101	web	https://huggingface.co/datasets/allenai/c4	2020
multilingual-data-peru (Grupo de Inteligencia Artificial PUCP, n.d.)	4	UNK	https://github.com/iapucp/multilingual-data-peru	2020
OSCAR 2301 (OSCAR, 2023)	152	web	https://huggingface.co/datasets/oscar-corpus/OSCAR-2301	2023
The Leipzig Corpora (Goldhahn et al., 2012)	136	newspapers, wikipedia	https://huggingface.co/datasets/imvladikon/leipzig_corpora_collection	2012
Tigrinya Language Modeling (Gaim et al., 2021)	1	news, blogs, books	https://zenodo.org/records/5139094	2021
Wikipedia 20231101 (Wikimedia Foundation, n.d.)	323	wikipedia	https://huggingface.co/datasets/wikimedia/wikipedia	2023
Wikisource 20231201 (Wikimedia Foundation, n.d.)	73	books	https://huggingface.co/datasets/wikimedia/wikisource	2023
Tatoeba challenge monolingual (Tiedemann, 2020)	280	wikimedia	https://github.com/Helsinki-NLP/Tatoeba-Challenge/blob/master/data/MonolingualData-v2020-07-28.md	2020

Glot500-c uses the following datasets: AI4Bharat,[26] AIFORTHAI-LotusCorpus,[27] Add (El-Haj et al., 2018), AfriBERTa (Ogueji et al., 2021), AfroMAFT (Adelani et al., 2022; Xue et al., 2021), Anuvaad,[28] AraBench (Sajjad et al., 2020), AUTSHUMATO,[29] Bloom (Leong et al., 2022), CC100 (Conneau et al., 2020), CCNet (Wenzek et al., 2020), CMU_Haitian_Creole,[30] CORP.NCHLT,[31] Clarin,[32] DART (Alsarsour et al., 2018), Earthlings (Dunn, 2020), FFR,[33] Flores200 (Costa-jussà et al., 2022), GiossaMedia (Góngora et al., 2021, 2022), Glosses (Camacho-Collados et al., 2016), Habibi (El-Haj, 2020), HinDialect (Bafna, 2022), HornMT,[34] IITB (Kunchukuttan et al., 2018), IndicNLP (Nakazawa et al., 2021), Indiccorp (Kakwani et al., 2020), isiZulu,[35] JParaCrawl (Morishita et al., 2020), KinyaSMT,[36] Leipzig-

[26] https://ai4bharat.org/

[27] https://github.com/korakot/corpus/releases/download/v1.0/AIFORTHAI-LotusCorpus.zip

[28] https://github.com/project-anuvaad/anuvaad-parallel-corpus

[29] https://autshumato.sourceforge.net/

[30] http://www.speech.cs.cmu.edu/haitian/text/

[31] https://repo.sadilar.org/handle/20.500.12185/7

[32] https://www.clarin.si/

[33] https://github.com/bonaventuredossou/ffr-v1/tree/master/FFR-Dataset

[34] https://github.com/asmelashteka/HornMT

[35] https://zenodo.org/record/5035171

[36] https://github.com/pniyongabo/kinyarwandaSMT

Data (Goldhahn et al., 2012), Lindat,[37] Lingala_Song_Lyrics,[38] Lyrics,[39] MC4 (Raffel et al., 2020), MTData (Gowda et al., 2021), MaCoCu (Bañón et al., 2022), Makerere MT Corpus,[40] Masakhane community,[41] Mburisano_Covid,[42] Menyo20K (Adelani et al., 2021), Minangkabau corpora (Koto and Koto, 2020), MoT (Palen-Michel et al., 2022), NLLB_seed (Costa-jussà et al., 2022), Nart/abkhaz,[43] OPUS (Tiedemann, 2012), OSCAR (Suárez et al., 2019), ParaCrawl (Bañón et al., 2020b), Parallel Corpora for Ethiopian Languages (Abate et al., 2018), Phontron (Neubig, 2011), QADI (Abdelali et al., 2021), Quechua-IIC (Zevallos et al., 2022), SLI_GalWeb.1.0 (Agerri et al., 2018), Shami (Abu Kwaik et al., 2018), Stanford NLP,[44] StatMT,[45] TICO (Anastasopoulos et al., 2020), TIL (Mirzakhalov et al., 2021), Tatoeba,[46] TeDDi (Moran et al., 2022), Tilde (Rozis and Skadiņš, 2017), W2C (Majliš, 2011), WAT (Nakazawa et al., 2022), WikiMatrix (Schwenk et al., 2021a), Wikipedia,[47] Workshop on NER for South and South East Asian Languages (Singh, 2008), XLSum (Hasan et al., 2021). We filter out Flores200 when processing the Glot500-c dataset. Glot500-c includes texts in languages, i.e., Azerbaijani, Gujarati, Igbo, Oromo, Rundi, Tigrinya and Yoruba, from the XLSum dataset.

A.2 Code

All the programming language splits are filtered using the following conditions:

- For files forked more than 25 times, we retain them if the average line length is less than 120, the maximum line length is less than 300, and the alphanumeric fraction is more than 30%.
- For files forked between 15 and 25 times, we retain them if the average line length is less than 90, the maximum line length is less than 150, and the alphanumeric fraction is more than 40%.
- For files forked less than 15 times, we retain them if the average line length is less than 80, the maximum line length is less than 120, and the alphanumeric fraction is more than 45%.

Subsequently, an aggressive MinHash deduplication pipeline with a threshold of 0.5 and a shingle size of 20 is applied. Finally, the resultant language splits are then capped at 5 million samples each.

B Additional Statistics of MaLA Corpus

B.1 Supported Languages

Table 20 shows the languages codes of MaLA corpus, where "unseen" means the languages are not used for training EMMA-500. The classification system for token counts categorizes language resources based on their size into five distinct tiers: "high" for resources exceeding 1 billion tokens, indicating a vast amount of data; "medium-high" for those with more than 100 million tokens, reflecting a substantial dataset; "medium" for resources that contain over 10 million tokens, representing a moderate size; "medium-low" for datasets with over "1 million tokens", indicating a smaller yet significant amount of data; and finally, "low" for resources containing less than 1 million tokens, which suggests a minimal data presence. This hierarchy helps in understanding the scale and potential utility of the language resources available. Figure 1 shows the number of texts and tokens in different resource groups.

[37] https://lindat.cz/faq-repository
[38] https://github.com/espoirMur/songs_lyrics_webscrap
[39] https://lyricstranslate.com/
[40] https://zenodo.org/record/5089560
[41] https://github.com/masakhane-io/masakhane-community
[42] https://repo.sadilar.org/handle/20.500.12185/536
[43] https://huggingface.co/datasets/Nart/abkhaz_text
[44] https://nlp.stanford.edu/
[45] https://statmt.org/
[46] https://tatoeba.org/en/
[47] https://huggingface.co/datasets/wikipedia

Category	Languages	Language Codes
high	27	fra_Latn, mon_Cyrl, kat_Geor, tgk_Cyrl, kaz_Cyrl, glg_Latn, hbs_Latn, kan_Knda, mal_Mlym, rus_Cyrl, cat_Latn, hye_Armn, guj_Gujr, slv_Latn, fil_Latn, bel_Cyrl, isl_Latn, nep_Deva, mlt_Latn, pan_Guru, afr_Latn, urd_Arab, mkd_Cyrl, aze_Latn, deu_Latn, eng_Latn, ind_Latn
low	210	prs_Arab, nqo_Nkoo, emp_Latn, pfl_Latn, teo_Latn, gpe_Latn, izz_Latn, shn_Mymr, hak_Latn, pls_Latn, evn_Cyrl, djk_Latn, toj_Latn, nog_Cyrl, ctu_Latn, tca_Latn, jiv_Latn, ach_Latn, mrj_Latn, ajp_Arab, apc_Arab, tab_Cyrl, hvn_Latn, tls_Latn, bak_Latn, ndc_Latn, trv_Latn, top_Latn, kjh_Cyrl, guh_Latn, mni_Mtei, csy_Latn, noa_Latn, dov_Latn, bho_Deva, kon_Latn, hne_Deva, kcg_Latn, mni_Beng, hus_Latn, pau_Latn, jbo_Latn, dtp_Latn, kmb_Latn, hau_Arab, pdc_Latn, nch_Latn, acf_Latn, bim_Latn, ixl_Latn, dty_Deva, kas_Arab, lrc_Arab, alz_Latn, lez_Cyrl, lld_Latn, tdt_Latn, acm_Arab, bih_Deva, mzh_Latn, guw_Latn, rop_Latn, rwo_Latn, ahk_Latn, qub_Latn, kri_Latn, gub_Latn, laj_Latn, sxn_Latn, luo_Latn, tly_Latn, pwn_Latn, mag_Deva, xav_Latn, bum_Latn, ubu_Latn, roa_Latn, mah_Latn, tsg_Latn, gcr_Latn, arn_Latn, csb_Latn, guc_Latn, bat_Latn, knj_Latn, cre_Latn, bus_Latn, anp_Deva, aln_Latn, nah_Latn, zai_Latn, kpv_Cyrl, enq_Latn, gvl_Latn, wal_Latn, fiu_Latn, swh_Latn, crh_Latn, nia_Latn, bqc_Latn, map_Latn, atj_Latn, npi_Deva, bru_Latn, din_Latn, pis_Latn, gur_Latn, cuk_Latn, zne_Latn, cdo_Latn, lhu_Latn, pcd_Latn, mas_Latn, bis_Latn, ncj_Latn, ibb_Latn, tay_Latn, bts_Latn, tzj_Latn, bzj_Latn, cce_Latn, jvn_Latn, ndo_Latn, rug_Latn, koi_Cyrl, mco_Latn, fat_Latn, olo_Latn, inb_Latn, mkn_Latn, qvi_Latn, mak_Latn, ktu_Latn, nrm_Latn, kua_Latn, san_Latn, nbl_Latn, kik_Latn, dyu_Latn, sgs_Latn, msm_Latn, mnw_Latn, zha_Latn, sja_Latn, xal_Cyrl, rmc_Latn, ami_Latn, sda_Latn, tdx_Latn, yap_Latn, tzh_Latn, sus_Latn, ikk_Latn, bas_Latn, nde_Latn, dsb_Latn, seh_Latn, knv_Latn, amu_Latn, dwr_Latn, iku_Cans, uig_Latn, bxr_Cyrl, tcy_Knda, mau_Latn, aoj_Latn, gor_Latn, cha_Latn, fip_Latn, chr_Cher, mdf_Cyrl, arb_Arab, quw_Latn, shp_Latn, spp_Latn, frp_Latn, ape_Latn, cbk_Latn, mnw_Mymr, mfe_Latn, jam_Latn, lad_Latn, awa_Deva, mad_Latn, ote_Latn, shi_Latn, btx_Latn, maz_Latn, ppk_Latn, smn_Latn, twu_Latn, blk_Mymr, msi_Latn, naq_Latn, tly_Arab, wuu_Hani, mos_Latn, cab_Latn, zlm_Latn, gag_Latn, suz_Deva, ksw_Mymr, gug_Latn, nij_Latn, nov_Latn, srm_Latn, jac_Latn, nyu_Latn, yom_Latn, gui_Latn
medium	68	tha_Thai, kat_Latn, lim_Latn, tgk_Arab, che_Cyrl, lav_Latn, xho_Latn, war_Latn, nan_Latn, grc_Grek, orm_Latn, zsm_Latn, cnh_Latn, yor_Latn, arg_Latn, tgk_Latn, azj_Latn, tel_Latn, slk_Latn, pap_Latn, zho_Hani, sme_Latn, tgl_Latn, uzn_Cyrl, als_Latn, san_Deva, azb_Arab, ory_Orya, lmo_Latn, bre_Latn, mvf_Mong, fao_Latn, oci_Latn, sah_Cyrl, sco_Latn, tuk_Latn, aze_Arab, hin_Deva, haw_Latn, glk_Arab, oss_Cyrl, lug_Latn, tet_Latn, tsn_Latn, hrv_Latn, gsw_Latn, arz_Arab, vec_Latn, mon_Latn, ilo_Latn, ctd_Latn, ben_Beng, roh_Latn, kal_Latn, asm_Beng, srp_Latn, bod_Tibt, hif_Latn, rus_Latn, nds_Latn, lus_Latn, ido_Latn, lao_Laoo, tir_Ethi, chv_Cyrl, wln_Latn, kaa_Latn, pnb_Arab
medium-high	79	div_Thaa, som_Latn, jpn_Japn, hat_Latn, sna_Latn, heb_Hebr, bak_Cyrl, nld_Latn, tel_Telu, kin_Latn, msa_Latn, gla_Latn, bos_Latn, dan_Latn, smo_Latn, ita_Latn, mar_Deva, pus_Arab, srp_Cyrl, spa_Latn, lat_Latn, hmn_Latn, sin_Sinh, zul_Latn, bul_Cyrl, amh_Ethi, ron_Latn, tam_Taml, khm_Khmr, nno_Latn, cos_Latn, fin_Latn, ori_Orya, uig_Arab, hbs_Cyrl, gle_Latn, cym_Latn, vie_Latn, kor_Hang, lit_Latn, yid_Hebr, ara_Arab, sqi_Latn, pol_Latn, tur_Latn, swa_Latn, hau_Latn, ceb_Latn, eus_Latn, kir_Cyrl, mlg_Latn, jav_Latn, snd_Arab, sot_Latn, por_Latn, uzb_Cyrl, fas_Arab, nor_Latn, est_Latn, hun_Latn, ibo_Latn, ltz_Latn, swe_Latn, tat_Cyrl, ast_Latn, mya_Mymr, uzb_Latn, sun_Latn, ell_Grek, ces_Latn, mri_Latn, ckb_Arab, kur_Latn, kaa_Cyrl, nob_Latn, ukr_Cyrl, fry_Latn, epo_Latn, nya_Latn
medium-low	162	aym_Latn, rue_Cyrl, rom_Latn, dzo_Tibt, poh_Latn, sat_Olck, ary_Arab, fur_Latn, mbt_Latn, bpy_Beng, iso_Latn, pon_Latn, glv_Latn, new_Deva, gym_Latn, bgp_Latn, kac_Latn, abt_Latn, quc_Latn, otq_Latn, sag_Latn, cak_Latn, avk_Latn, pam_Latn, meo_Latn, tum_Latn, bam_Latn, kha_Latn, syr_Syrc, kom_Cyrl, nhe_Latn, bal_Arab, srd_Latn, krc_Cyrl, lfn_Latn, bar_Latn, rcf_Latn, nav_Latn, nnb_Latn, sdh_Arab, aka_Latn, bew_Cyrl, bbc_Latn, meu_Latn, zza_Latn, ext_Latn, yue_Hani, ekk_Latn, xmf_Geor, nap_Latn, mzn_Arab, pcm_Latn, lij_Latn, myv_Cyrl, scn_Latn, dag_Latn, ban_Latn, twi_Latn, udm_Cyrl, som_Arab, nso_Latn, pck_Latn, crs_Latn, acr_Latn, tat_Latn, afb_Arab, uzs_Arab, hil_Latn, mgh_Latn, tpi_Latn, ady_Cyrl, pag_Latn, kiu_Latn, ber_Latn, iba_Latn, ksh_Latn, plt_Latn, lin_Latn, chk_Latn, tzo_Latn, tlh_Latn, ile_Latn, lub_Latn, hui_Latn, min_Latn, bjn_Latn, szl_Latn, kbp_Latn, inh_Cyrl, que_Latn, ven_Latn, vls_Latn, kbd_Cyrl, run_Latn, wol_Latn, ace_Latn, ada_Latn, kek_Latn, yua_Latn, tbz_Latn, gom_Latn, ful_Latn, mrj_Cyrl, abk_Cyrl, tuc_Latn, stq_Latn, mwl_Latn, tvl_Latn, quh_Latn, gom_Deva, mhr_Cyrl, fij_Latn, grn_Latn, zap_Latn, mam_Latn, mps_Latn, tiv_Latn, ksd_Latn, ton_Latn, bik_Latn, vol_Latn, ava_Cyrl, tso_Latn, szy_Latn, ngu_Latn, hyw_Armn, fon_Latn, skr_Arab, kos_Latn, tyz_Latn, kur_Arab, srn_Latn, tyv_Cyrl, bci_Latn, vep_Latn, crh_Cyrl, kpg_Latn, hsb_Latn, ssw_Latn, zea_Latn, ewe_Latn, ium_Latn, diq_Latn, ltg_Latn, nzi_Latn, guj_Deva, ina_Latn, pms_Latn, bua_Cyrl, lvs_Latn, eml_Latn, hmo_Latn, kum_Cyrl, kab_Latn, chm_Cyrl, cor_Latn, cfm_Latn, alt_Cyrl, bcl_Latn, ang_Latn, frr_Latn, mai_Deva
unseen	393	rap_Latn, pmf_Latn, lsi_Latn, dje_Latn, bkx_Latn, ipk_Latn, syw_Deva, ann_Latn, bag_Latn, bat_Cyrl, chu_Cyrl, gwc_Arab, adh_Latn, szy_Hani, shi_Arab, njy_Latn, pdu_Latn, buo_Latn, cuv_Latn, udg_Mlym, bax_Latn, tio_Latn, kjb_Latn, taj_Deva, lez_Latn, olo_Cyrl, rnl_Latn, bri_Latn, inh_Latn, kas_Cyrl, wni_Latn, anp_Latn, tsc_Latn, mgg_Latn, udi_Cyrl, mdf_Latn, agr_Latn, xty_Latn, llg_Latn, nge_Latn, gan_Latn, tuv_Latn, stk_Latn, nut_Latn, thy_Thai, lgr_Latn, hnj_Latn, dar_Cyrl, aia_Latn, lwl_Thai, tnl_Latn, tvs_Latn, jra_Khmr, tay_Hani, gal_Latn, ybi_Deva, snk_Arab, gag_Cyrl, tuk_Cyrl, trv_Hani, ydd_Hebr, kea_Latn, gbm_Deva, kwi_Latn, hro_Latn, rki_Latn, quy_Latn, tdg_Deva, zha_Hani, pcg_Mlym, tom_Latn, nsn_Latn, quf_Latn, jmx_Latn, kqr_Latn, mrn_Latn, bxa_Latn, abc_Latn, mve_Arab, lfa_Latn, qup_Latn, yin_Latn, roo_Latn, mrw_Latn, nxa_Latn, yrk_Cyrl, bem_Latn, kvt_Latn, csw_Cans, bjr_Latn, mgm_Latn, ngn_Latn, pib_Latn, quz_Latn, awb_Latn, myk_Latn, otq_Arab, ino_Latn, tkd_Latn, bef_Latn, bug_Bugi, aeu_Latn, nlv_Latn, dty_Latn, bkc_Latn, mmu_Latn, hak_Hani, sea_Latn, mlk_Latn, cbr_Latn, lmp_Latn, tnn_Latn, qvz_Latn, pbt_Arab, cjs_Cyrl, mlw_Latn, mnf_Latn, bfm_Latn, dig_Latn, thk_Latn, zxx_Latn, lkb_Latn, chr_Latn, pnt_Latn, vif_Latn, fli_Latn, got_Latn, hbb_Latn, tll_Latn, bug_Latn, kxp_Arab, qaa_Latn, krr_Khmr, kjg_Laoo, isu_Latn, kmu_Latn, gof_Latn, sdk_Latn, mne_Latn, baw_Latn, idt_Latn, xkg_Latn, mgo_Latn, dtr_Latn, kms_Latn, ffm_Latn, hna_Latn, nxl_Latn, bfd_Latn, odk_Arab, miq_Latn, mhx_Latn, kam_Latn, yao_Latn, pnt_Grek, kby_Latn, kpv_Latn, kbx_Latn, cim_Latn, qvo_Latn, pih_Latn, nog_Latn, nco_Latn, rmy_Cyrl, clo_Latn, dmg_Latn, aaa_Latn, rel_Latn, ben_Latn, loh_Latn, thl_Deva, chd_Latn, cni_Latn, cjs_Latn, lbe_Latn, ybh_Deva, zxx_Zyyy, awa_Latn, gou_Latn, xmm_Latn, nqo_Latn, rut_Cyrl, kbq_Latn, tkr_Latn, dwr_Ethi, ckt_Cyrl, ady_Latn, yea_Mlym, nhx_Latn, niv_Cyrl, bwt_Latn, xmg_Latn, chy_Latn, mfj_Latn, hre_Latn, bbk_Latn, shn_Latn, lrc_Latn, qvc_Latn, muv_Mlym, mdr_Latn, luy_Latn, lzh_Hani, fuh_Latn, mle_Latn, brx_Deva, pex_Latn, kau_Latn, yrk_Latn, hin_Latn, ekm_Latn, msb_Latn, unr_Orya, cac_Latn, chp_Cans, ckt_Latn, bss_Latn, lts_Latn, bbj_Latn, ttt_Cyrl, kwu_Latn, smn_Cyrl, kpy_Cyrl, tod_Latn, wbm_Latn, tcy_Latn, arc_Syrc, nst_Latn, tuz_Latn, bob_Latn, bfn_Latn, pli_Deva, snl_Latn, kwd_Latn, lgg_Latn, nza_Latn, wbr_Deva, lan_Latn, kmz_Latn, bzi_Thai, hao_Latn, nla_Latn, qxr_Latn, ken_Latn, tbj_Latn, blk_Latn, ybb_Latn, nwe_Latn, gan_Hani, snk_Latn, kak_Latn, tpl_Latn, hla_Latn, tks_Arab, pea_Latn, bya_Latn, enc_Latn, jgo_Latn, tnp_Latn, aph_Deva, bgf_Latn, brv_Laoo, nod_Thai, niq_Latn, nwi_Latn, xmd_Latn, gbj_Orya, syr_Latn, ify_Latn, xal_Latn, bra_Deva, cgc_Latn, bhs_Latn, pwg_Latn, ang_Runr, oki_Latn, qve_Latn, qvm_Latn, bkm_Latn, bkh_Latn, niv_Latn, zuh_Latn, mry_Latn, fiu_Cyrl, ssn_Latn, rki_Mymr, sox_Latn, yav_Latn, nyo_Latn, dag_Arab, qxh_Latn, bze_Latn, myx_Latn, zaw_Latn, ddg_Latn, wnk_Latn, bwx_Latn, mqy_Latn, lad_Hebr, boz_Latn, lue_Latn, ded_Latn, pli_Latn, avk_Cyrl, wms_Latn, sgd_Latn, azn_Latn, ajz_Latn, psp_Latn, jra_Latn, smt_Latn, ags_Latn, csw_Latn, wtk_Latn, emp_Cyrl, koi_Latn, tkr_Cyrl, amp_Latn, ymp_Latn, mfh_Latn, tdb_Deva, omw_Latn, khb_Talu, doi_Deva, gld_Cyrl, ava_Latn, chu_Latn, dnw_Latn, azo_Latn, dug_Latn, bce_Latn, kmr_Latn, kpy_Armn, abq_Cyrl, trp_Latn, ewo_Latn, the_Deva, hig_Latn, pkb_Latn, mxu_Latn, oji_Latn, tnt_Latn, mzm_Latn, mns_Cyrl, lbe_Cyrl, qvh_Latn, kmg_Latn, sps_Latn, brb_Khmr, tah_Latn, sxb_Latn, mkz_Latn, mgq_Latn, got_Goth, lns_Latn, arc_Latn, akb_Latn, skr_Latn, nsk_Cans, sml_Latn, pce_Mymr, eee_Thai, lhm_Deva, yux_Cyrl, bqm_Latn, bcc_Arab, nas_Latn, agq_Latn, xog_Latn, tsb_Latn, fub_Latn, mqj_Latn, nsk_Latn, bxr_Latn, dln_Latn, ozm_Latn, rmy_Latn, cre_Cans, kim_Cyrl, cuh_Latn, ngl_Latn, yas_Latn, bud_Latn, miy_Latn, ame_Latn, pnz_Latn, raj_Deva, enb_Latn, cmo_Khmr, saq_Latn, tpu_Khmr, eve_Cyrl, cdo_Hani

Table 20: Languages by resource groups

B.2 Data Analysis

We examine the Unicode block distribution of each language, which counts the percentage of tokens falling into the Unicode block of each language. This aims to check whether language code conversion and writing system recognition are reasonably good. Figure 2 shows the Unicode block distribution. The result aligns with our observation of the presence of code-mixing, but in general, the majority of languages have tokens falling in their own Unicode blocks.

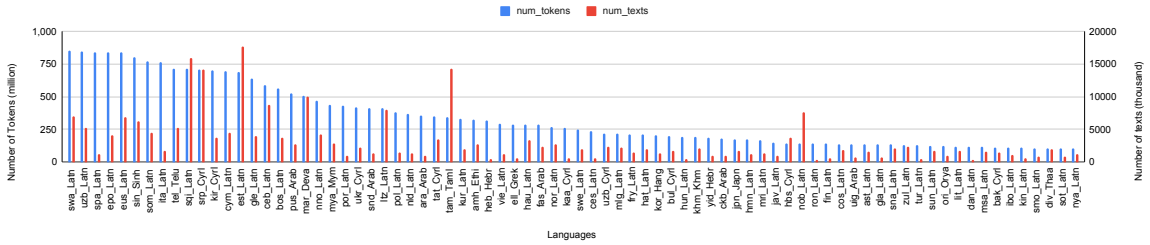
We also check the data source distribution as shown in Figure 3 to see where the texts in the MaLA corpus come from. The main source is common crawl, e.g., CC 2018, CC, OSCAR, and CC 20220801. A large number of documents come from Earthlings which comes from Glot500-c (Imani et al., 2023). For web-crawled data with a URL in the original metadata of corpora like CulturaX (Nguyen et al., 2023) and HPLT (de Gibert et al., 2024), we extract the domain. Thus, the final corpus has many sources with a small portion.

C Evaluation Benchmarks

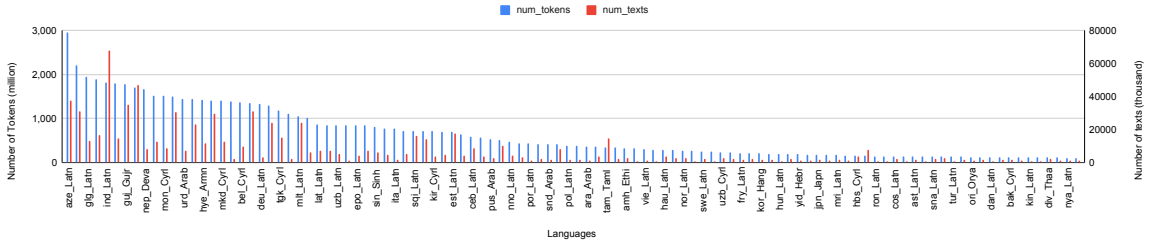
C.1 PolyWrite

The PolyWrite dataset has 51 writing tasks with the number of prompts per task shown in Figure 4. We use Google Translate to translate the English prompts into 240 languages and back-translate for

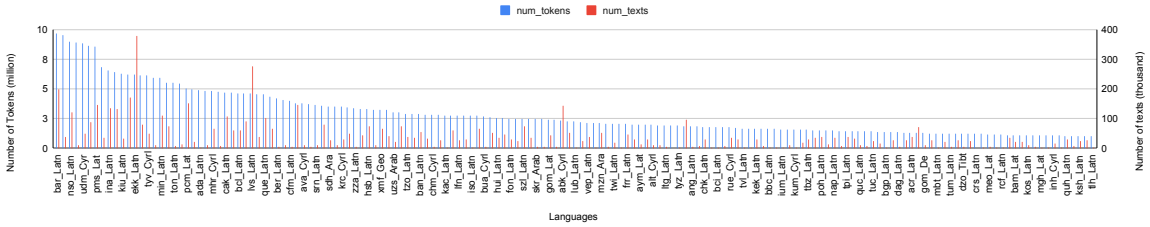
(a) High



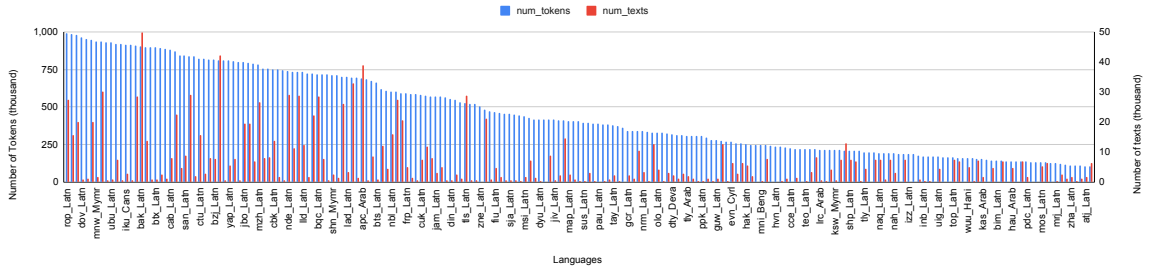
(b) Medium-high



(c) Medium



(d) Medium-low



(e) Low

Figure 1: The number of texts and tokens of MaLA corpus in different resource groups.

translation quality assessment.

The metadata of PolyWrite includes several key fields. The `category` specifies the task type. The `name` field typically holds the specific identifier for each prompt, while `prompt_en` contains the English version of the prompt. `lang_script` identifies the language and script used, ensuring correct language processing. The `prompt_translated` field holds the translated prompt in the target language, and `prompt_backtranslated` contains the back-translated version to assess translation

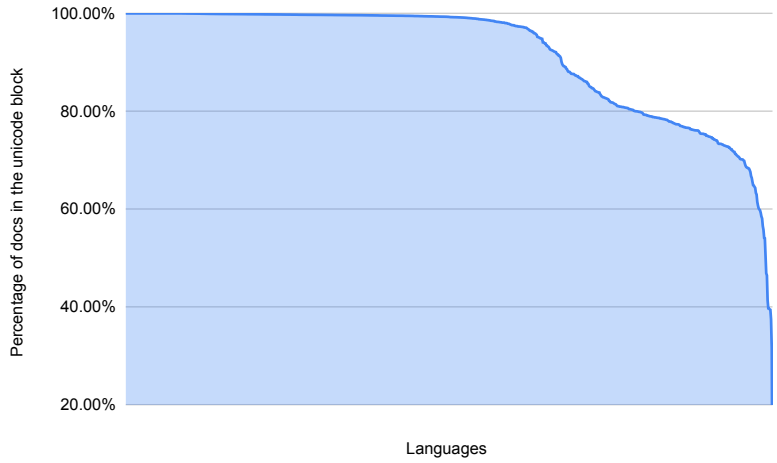


Figure 2: Unicode block distribution that measures the percentage of token counts falling into the Unicode block of each language

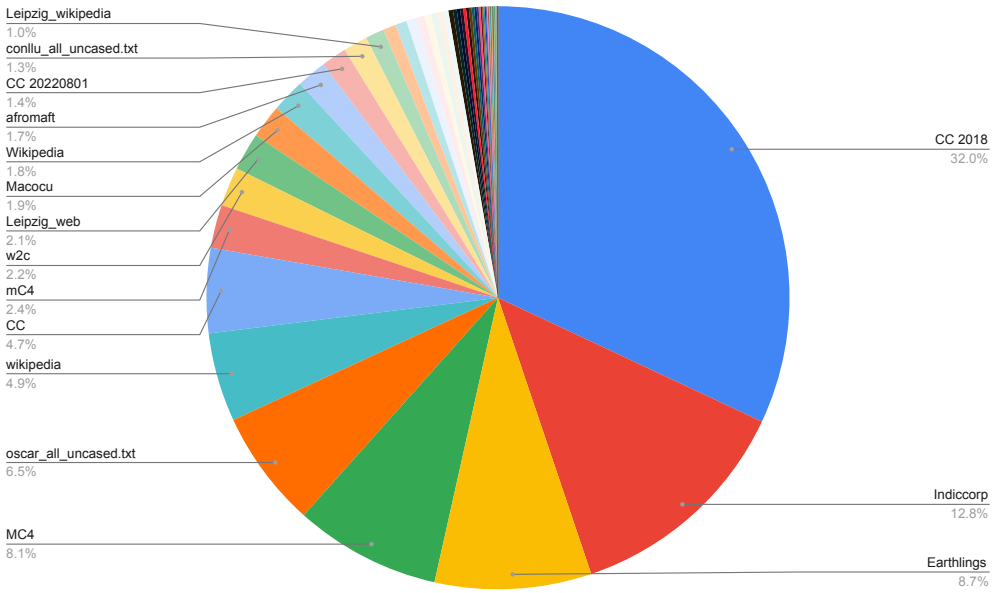


Figure 3: Data source distribution of MaLA corpus calculated by the number of documents

quality. Both `bleu` and `chrf++` fields provide numeric evaluation metrics, with BLEU and chrF++ scores measuring the quality of the generated text. Finally, the `uuid` ensures a unique identifier for each dataset entry, allowing for precise reference and tracking of individual prompts. To mitigate errors introduced by the machine translation process, we filter out prompts with a BLEU score of less than 20. Figure 5 shows the average BLEU score of each language in the final dataset.

D Detailed Results

This section presents detailed results of the evaluation. For benchmarks consisting of multiple languages, we host the results on GitHub[48]. Those benchmarks include ARC multilingual, BELEBELE, Glot500-c test set, PBC, SIB-200, Taxi-1500, FLORES200, XLSum, Aya evaluation suite, and PolyWrite. For FLORES200, XLSum, Aya evaluation suite, and PolyWrite, we also release the generated texts of

[48] `https://github.com/MaLA-LM/emma-500/tree/main/evaluation_results`

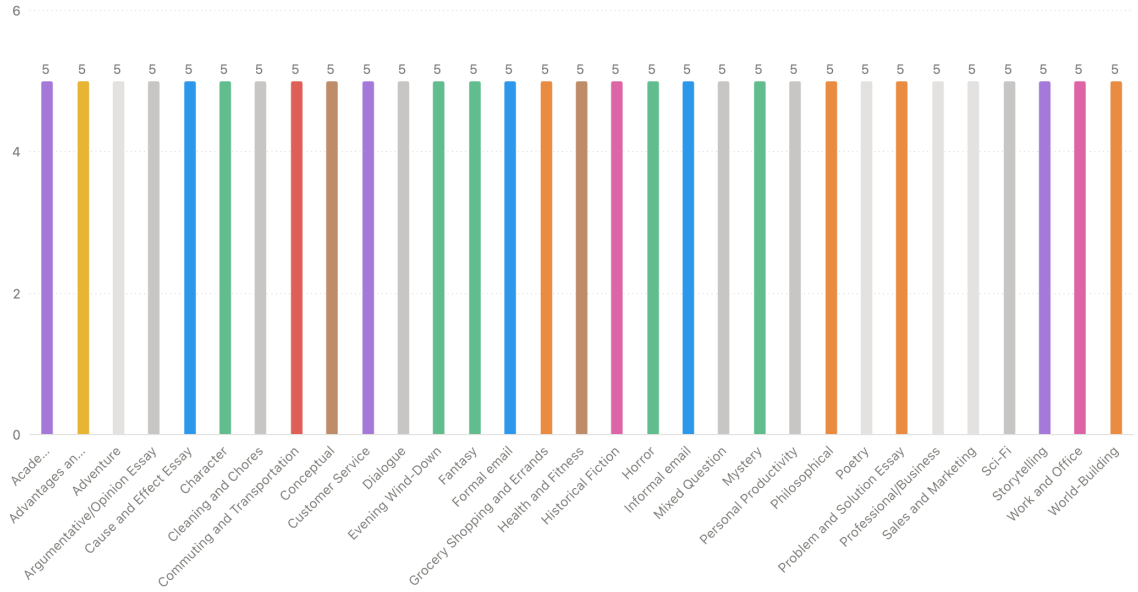


Figure 4: Writing tasks in the PolyWrite dataset.

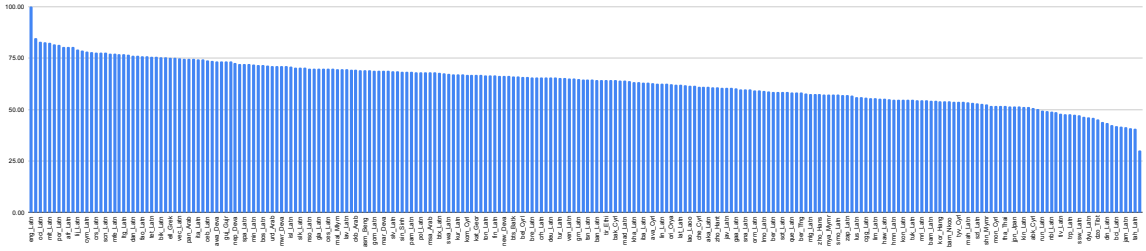


Figure 5: Mean BLEU scores per language in the PolyWrite dataset.

all compared models.

D.1 Commonsense Reasoning

Table 21: 0-shot results (ACC) on XCOPA in all languages

Model	Avg	et-acc	stderr	ht-acc	stderr	id-acc	stderr	it-acc	stderr	qu-acc	stderr	sw-acc	stderr	ta-acc	stderr	th-acc	stderr	tr-acc	stderr	vi-acc	stderr	zh-acc	stderr
Llama 2 7B	0.5667	0.4860	0.0224	0.5060	0.0224	0.6240	0.0217	0.6580	0.0212	0.5160	0.0224	0.5220	0.0224	0.5340	0.0223	0.5620	0.0222	0.5480	0.0223	0.6280	0.0216	0.6500	0.0214
Llama 2 7B Chat	0.5585	0.4780	0.0224	0.5080	0.0224	0.6240	0.0217	0.6720	0.0210	0.5060	0.0224	0.5220	0.0224	0.5060	0.0224	0.5500	0.0223	0.5520	0.0223	0.6120	0.0218	0.6140	0.0218
CodeLlama 2 7B	0.5469	0.4680	0.0223	0.5180	0.0224	0.5740	0.0221	0.6300	0.0216	0.5160	0.0224	0.4880	0.0224	0.5500	0.0223	0.5540	0.0223	0.5380	0.0223	0.5580	0.0222	0.6220	0.0217
LLaMAX Llama 2 7B	0.5438	0.4920	0.0224	0.5260	0.0224	0.5380	0.0223	0.5260	0.0224	0.5140	0.0224	0.5400	0.0223	0.5800	0.0221	0.5720	0.0221	0.5300	0.0223	0.5300	0.0223	0.6340	0.0216
LLaMAX Llama 2 7B Alpaca	0.5660	0.5120	0.0224	0.5420	0.0223	0.5720	0.0221	0.6100	0.0218	0.5240	0.0224	0.5500	0.0223	0.5700	0.0222	0.5640	0.0222	0.5520	0.0223	0.5520	0.0223	0.6780	0.0209
MaLA-500 Llama 2 10B v1	0.5309	0.4860	0.0224	0.5340	0.0223	0.5300	0.0223	0.5940	0.0220	0.5020	0.0224	0.5280	0.0223	0.5760	0.0221	0.5420	0.0223	0.5160	0.0224	0.5240	0.0224	0.5080	0.0224
MaLA-500 Llama 2 10B v2	0.5309	0.4860	0.0224	0.5340	0.0223	0.5300	0.0223	0.5940	0.0220	0.5020	0.0224	0.5280	0.0223	0.5760	0.0221	0.5420	0.0223	0.5160	0.0224	0.5240	0.0224	0.5080	0.0224
YaYi Llama 2 7B	0.5671	0.4880	0.0224	0.5080	0.0224	0.6260	0.0217	0.6700	0.0210	0.5060	0.0224	0.5320	0.0223	0.5520	0.0223	0.5420	0.0223	0.5540	0.0223	0.6320	0.0216	0.6280	0.0216
TowerBase Llama 2 7B	0.5633	0.4600	0.0223	0.5020	0.0224	0.6020	0.0219	0.7080	0.0204	0.5220	0.0224	0.5060	0.0224	0.5440	0.0223	0.5600	0.0222	0.5380	0.0223	0.5920	0.0220	0.6620	0.0212
TowerInstruct Llama 2 7B	0.5705	0.4880	0.0224	0.5160	0.0224	0.6200	0.0217	0.7100	0.0203	0.5200	0.0224	0.5100	0.0224	0.5420	0.0223	0.5640	0.0222	0.5460	0.0223	0.5860	0.0220	0.6740	0.0210
Occiglot Mistral 7B v0.1	0.5667	0.4720	0.0223	0.5140	0.0224	0.5700	0.0222	0.7460	0.0195	0.5200	0.0224	0.5160	0.0224	0.5720	0.0221	0.5580	0.0222	0.5440	0.0223	0.5520	0.0223	0.6700	0.0210
Occiglot Mistral 7B v0.1 Instruct	0.5655	0.4680	0.0223	0.5100	0.0224	0.5840	0.0221	0.7380	0.0197	0.5280	0.0223	0.5000	0.0224	0.5660	0.0222	0.5500	0.0223	0.5620	0.0222	0.5580	0.0222	0.6560	0.0213
BLOOM 7B	0.5689	0.4820	0.0224	0.5080	0.0224	0.6980	0.0206	0.5280	0.0223	0.5080	0.0224	0.5180	0.0224	0.5920	0.0220	0.5540	0.0223	0.5100	0.0224	0.7080	0.0204	0.6520	0.0213
BLOOMZ 7B	0.5487	0.4920	0.0224	0.5400	0.0223	0.6060	0.0219	0.5140	0.0224	0.5060	0.0224	0.5340	0.0223	0.5740	0.0221	0.5300	0.0223	0.5220	0.0224	0.5980	0.0219	0.6200	0.0217
mGPT	0.5504	0.5300	0.0223	0.4980	0.0224	0.5880	0.0220	0.5820	0.0221	0.5060	0.0224	0.5640	0.0222	0.5320	0.0223	0.5520	0.0223	0.5600	0.0222	0.6020	0.0219	0.5400	0.0223
mGPT 13B	0.5618	0.5080	0.0224	0.5180	0.0224	0.6260	0.0217	0.6080	0.0219	0.4820	0.0224	0.5800	0.0221	0.5480	0.0223	0.5280	0.0223	0.5680	0.0222	0.6340	0.0216	0.5800	0.0221
XGLM 7.5B	0.6064	0.6140	0.0218	0.5740	0.0221	0.6940	0.0206	0.6360	0.0215	0.4880	0.0224	0.6000	0.0219	0.5460	0.0223	0.5940	0.0220	0.5840	0.0221	0.7020	0.0205	0.6380	0.0215
YaYi 7B	0.5664	0.5040	0.0224	0.5300	0.0223	0.6340	0.0216	0.5180	0.0224	0.5180	0.0224	0.5540	0.0223	0.5620	0.0222	0.5460	0.0223	0.5200	0.0224	0.6640	0.0211	0.6800	0.0209
Llama 3 8B	0.6171	0.5340	0.0223	0.5280	0.0223	0.7140	0.0202	0.7160	0.0202	0.5120	0.0224	0.5800	0.0221	0.6000	0.0219	0.5860	0.0220	0.6240	0.0217	0.7160	0.0202	0.6780	0.0209
Llama 3.1 8B	0.6171	0.5300	0.0223	0.5380	0.0223	0.7160	0.0202	0.7260	0.0200	0.4880	0.0224	0.5540	0.0223	0.6120	0.0218	0.5780	0.0221	0.6180	0.0218	0.7240	0.0200	0.7040	0.0204
Gemma 7B	0.6364	0.5920	0.0220	0.5480	0.0223	0.7200	0.0201	0.7280	0.0199	0.5000	0.0224	0.6060	0.0219	0.6160	0.0218	0.6040	0.0219	0.6560	0.0213	0.7420	0.0196	0.6880	0.0207
Gemma 2 9B	0.6633	0.6380	0.0215	0.5280	0.0223	0.7780	0.0186	0.7580	0.0192	0.5040	0.0224	0.6380	0.0215	0.6360	0.0215	0.6380	0.0215	0.6740	0.0210	0.7660	0.0190	0.7380	0.0197
Qwen 2 7B	0.6031	0.5080	0.0224	0.5080	0.0224	0.7060	0.0204	0.7140	0.0202	0.5040	0.0224	0.5240	0.0224	0.5280	0.0223	0.6100	0.0218	0.5720	0.0221	0.6940	0.0206	0.7660	0.0190
Qwen 1.5 7B	0.5944	0.5200	0.0224	0.5300	0.0223	0.6440	0.0214	0.6520	0.0213	0.5100	0.0224	0.5260	0.0224	0.5580	0.0222	0.5760	0.0221	0.5880	0.0220	0.6920	0.0207	0.7420	0.0196
EMMA-500 Llama 2 7B	0.6311	0.6140	0.0218	0.5800	0.0221	0.7420	0.0196	0.6940	0.0206	0.5240	0.0224	0.6620	0.0212	0.6000	0.0219	0.5560	0.0222	0.6200	0.0217	0.7020	0.0205	0.6480	0.0214

Table 22: 0-shot results (ACC) on XStoryCloze in all languages

Model	Avg	ar-acc	stderr	en-acc	stderr	es-acc	stderr	eu-acc	stderr	hi-acc	stderr	id-acc	stderr	my-acc	stderr	ru-acc	stderr	sw-acc	stderr	te-acc	stderr	zh-acc	stderr
Llama 2 7B	0.5755	0.4990	0.0129	0.7704	0.0108	0.6737	0.0121	0.5036	0.0129	0.5374	0.0128	0.5923	0.0126	0.4805	0.0129	0.6300	0.0124	0.5050	0.0129	0.5433	0.0128	0.5956	0.0126
Llama 2 7B Chat	0.5841	0.5050	0.0129	0.7869	0.0105	0.6711	0.0121	0.5083	0.0129	0.5407	0.0128	0.5963	0.0126	0.4864	0.0129	0.6552	0.0122	0.5202	0.0129	0.5334	0.0128	0.6221	0.0125
CodeLlama 2 7B	0.5568	0.5010	0.0129	0.7148	0.0116	0.6340	0.0124	0.5043	0.0129	0.4970	0.0129	0.5586	0.0128	0.4937	0.0129	0.5923	0.0126	0.5003	0.0129	0.5374	0.0128	0.5917	0.0126
LLaMAX Llama 2 7B	0.6036	0.5890	0.0127	0.7551	0.0111	0.6525	0.0123	0.5447	0.0128	0.5817	0.0127	0.6062	0.0126	0.5248	0.0129	0.6122	0.0125	0.5718	0.0127	0.5930	0.0126	0.6082	0.0126
LLaMAX Llama 2 7B Alpaca	0.6383	0.6036	0.0126	0.8147	0.0100	0.7068	0.0117	0.5486	0.0128	0.6214	0.0125	0.6645	0.0122	0.5381	0.0128	0.6744	0.0121	0.6016	0.0126	0.5930	0.0126	0.6545	0.0122
MaLA-500 Llama 2 10B v1	0.5307	0.4818	0.0129	0.7353	0.0114	0.6241	0.0125	0.4990	0.0129	0.4765	0.0129	0.4792	0.0129	0.4626	0.0128	0.5493	0.0128	0.4871	0.0129	0.5261	0.0128	0.5169	0.0129
MaLA-500 Llama 2 10B v2	0.5307	0.4818	0.0129	0.7353	0.0114	0.6241	0.0125	0.4990	0.0129	0.4765	0.0129	0.4792	0.0129	0.4626	0.0128	0.5493	0.0128	0.4871	0.0129	0.5261	0.0128	0.5169	0.0129
YaYi Llama 2 7B	0.5842	0.4997	0.0129	0.7909	0.0105	0.6870	0.0119	0.5063	0.0129	0.5427	0.0128	0.6142	0.0125	0.4745	0.0129	0.6479	0.0123	0.5003	0.0129	0.5394	0.0128	0.6234	0.0125
TowerBase Llama 2 7B	0.5778	0.4917	0.0129	0.7723	0.0108	0.6982	0.0118	0.5076	0.0129	0.5288	0.0128	0.5831	0.0127	0.4838	0.0129	0.6704	0.0121	0.5036	0.0129	0.5314	0.0128	0.5850	0.0127
TowerInstruct Llama 2 7B	0.5924	0.4931	0.0129	0.8087	0.0101	0.7161	0.0116	0.5069	0.0129	0.5295	0.0128	0.5956	0.0126	0.4871	0.0129	0.6936	0.0119	0.5149	0.0129	0.5414	0.0128	0.6300	0.0124
Occiglot Mistral 7B v0.1	0.5810	0.5129	0.0129	0.7737	0.0108	0.7340	0.0114	0.5208	0.0129	0.5149	0.0129	0.5864	0.0127	0.4798	0.0129	0.6294	0.0124	0.4983	0.0129	0.5314	0.0128	0.6089	0.0126
Occiglot Mistral 7B v0.1 Instruct	0.5939	0.5268	0.0128	0.7942	0.0104	0.7419	0.0113	0.5301	0.0128	0.5275	0.0128	0.6036	0.0126	0.4825	0.0129	0.6506	0.0123	0.5043	0.0129	0.5381	0.0128	0.6334	0.0124
BLOOM 7B	0.5930	0.5857	0.0127	0.7055	0.0117	0.6618	0.0122	0.5725	0.0127	0.6042	0.0126	0.6453	0.0123	0.4891	0.0129	0.5275	0.0128	0.5394	0.0128	0.5731	0.0127	0.6188	0.0125
BLOOMZ 7B	0.5712	0.5652	0.0128	0.7300	0.0114	0.6459	0.0123	0.5109	0.0129	0.5764	0.0127	0.5533	0.0128	0.4825	0.0129	0.5215	0.0129	0.5215	0.0129	0.5817	0.0127	0.5943	0.0126
mGPT	0.5443	0.4931	0.0129	0.5996	0.0126	0.5546	0.0128	0.5460	0.0128	0.5275	0.0128	0.5314	0.0128	0.5122	0.0129	0.5665	0.0128	0.5500	0.0128	0.5725	0.0127	0.5341	0.0128
mGPT 13B	0.5644	0.5162	0.0129	0.6420	0.0123	0.5864	0.0127	0.5539	0.0128	0.5506	0.0128	0.5811	0.0127	0.5122	0.0129	0.5943	0.0126	0.5460	0.0128	0.5764	0.0127	0.5493	0.0128
XGLM 7.5B	0.6075	0.5612	0.0128	0.6982	0.0118	0.6386	0.0124	0.5771	0.0127	0.5884	0.0127	0.6300	0.0124	0.5705	0.0127	0.6340	0.0124	0.5936	0.0126	0.6023	0.0126	0.5890	0.0127
YaYi 7B	0.6067	0.6181	0.0125	0.7432	0.0112	0.6942	0.0119	0.5606	0.0128	0.6367	0.0124	0.6241	0.0125	0.4924	0.0129	0.5215	0.0129	0.5361	0.0128	0.5791	0.0127	0.6678	0.0121
Llama 3 8B	0.6341	0.5864	0.0127	0.7869	0.0105	0.7062	0.0117	0.5579	0.0128	0.6281	0.0124	0.6592	0.0122	0.5109	0.0129	0.6876	0.0119	0.5639	0.0128	0.6300	0.0124	0.6578	0.0122
Llama 3.1 8B	0.6358	0.5910	0.0127	0.7816	0.0106	0.7081	0.0117	0.5533	0.0128	0.6327	0.0124	0.6803	0.0120	0.5242	0.0129	0.6863	0.0119	0.5592	0.0128	0.6115	0.0125	0.6658	0.0121
Gemma 7B	0.6501	0.6042	0.0126	0.8015	0.0103	0.7062	0.0117	0.5758	0.0127	0.6492	0.0123	0.6764	0.0120	0.5228	0.0129	0.7055	0.0117	0.6214	0.0125	0.6327	0.0124	0.6559	0.0122
Gemma 2 9B	0.6767	0.6525	0.0123	0.8015	0.0103	0.7426	0.0113	0.6016	0.0126	0.6691	0.0121	0.7134	0.0116	0.5513	0.0128	0.7373	0.0113	0.6373	0.0124	0.6479	0.0123	0.6896	0.0119
Qwen 2 7B	0.6146	0.5996	0.0126	0.7895	0.0105	0.6976	0.0118	0.5202	0.0129	0.5797	0.0127	0.6439	0.0123	0.4897	0.0129	0.6956	0.0118	0.5142	0.0129	0.5407	0.0128	0.6903	0.0119
Qwen 1.5 7B	0.5985	0.5539	0.0128	0.7816	0.0106	0.6830	0.0120	0.5189	0.0129	0.5566	0.0128	0.6287	0.0124	0.4924	0.0129	0.6327	0.0124	0.5169	0.0129	0.5394	0.0128	0.6797	0.0120
EMMA-500 Llama 2 7B	0.6638	0.6625	0.0122	0.7644	0.0109	0.7002	0.0118	0.6473	0.0123	0.6492	0.0123	0.6863	0.0119	0.5791	0.0127	0.6850	0.0120	0.6473	0.0123	0.6466	0.0123	0.6340	0.0124

Table 23: 0-shot results (ACC) on XWinograd in all languages

Model	Avg	en-acc	stderr	fr-acc	stderr	jp-acc	stderr	pt-acc	stderr	ru-acc	stderr	zh-acc	stderr
Llama 2 7B	0.7247	0.8791	0.0068	0.6627	0.0522	0.7028	0.0148	0.7224	0.0277	0.6825	0.0263	0.6984	0.0205
Llama 2 7B Chat	0.6945	0.8555	0.0073	0.7108	0.0501	0.6840	0.0150	0.6464	0.0295	0.6571	0.0268	0.6131	0.0217
CodeLlama 2 7B	0.7092	0.8452	0.0075	0.6747	0.0517	0.6361	0.0155	0.7224	0.0277	0.6667	0.0266	0.7103	0.0202
LLaMAX Llama 2 7B	0.6749	0.7789	0.0086	0.6145	0.0537	0.7101	0.0147	0.6540	0.0294	0.5556	0.0280	0.7361	0.0197
LLaMAX Llama 2 7B Alpaca	0.6986	0.8275	0.0078	0.6627	0.0522	0.7237	0.0144	0.6616	0.0292	0.5937	0.0277	0.7222	0.0200
MaLA-500 Llama 2 10B v1	0.6589	0.8366	0.0077	0.6386	0.0531	0.6017	0.0158	0.6920	0.0285	0.6190	0.0274	0.5655	0.0221
MaLA-500 Llama 2 10B v2	0.6589	0.8366	0.0077	0.6386	0.0531	0.6017	0.0158	0.6920	0.0285	0.6190	0.0274	0.5655	0.0221
YaYi Llama 2 7B	0.7450	0.8852	0.0066	0.7108	0.0501	0.7226	0.0145	0.7414	0.0270	0.7175	0.0254	0.6925	0.0206
TowerBase Llama 2 7B	0.7429	0.8714	0.0069	0.7470	0.0480	0.6945	0.0149	0.7567	0.0265	0.6476	0.0270	0.7401	0.0196
TowerInstruct Llama 2 7B	0.7400	0.8628	0.0071	0.7711	0.0464	0.6840	0.0150	0.7719	0.0259	0.6635	0.0267	0.6865	0.0207
Occiglot Mistral 7B v0.1	0.7461	0.8654	0.0071	0.7952	0.0446	0.6601	0.0153	0.7186	0.0278	0.6635	0.0267	0.7738	0.0187
Occiglot Mistral 7B v0.1 Instruct	0.7293	0.8589	0.0072	0.7470	0.0480	0.6455	0.0155	0.7072	0.0281	0.6476	0.0270	0.7698	0.0188
BLOOM 7B	0.7013	0.8224	0.0079	0.7108	0.0501	0.5881	0.0159	0.7681	0.0261	0.5746	0.0279	0.7440	0.0195
BLOOMZ 7B	0.6795	0.8340	0.0077	0.7349	0.0487	0.5756	0.0160	0.6730	0.0290	0.5492	0.0281	0.7103	0.0202
mGPT	0.5969	0.6267	0.0100	0.5904	0.0543	0.5349	0.0161	0.5741	0.0305	0.5810	0.0278	0.6746	0.0209
mGPT 13B	0.6359	0.7062	0.0094	0.6386	0.0531	0.5777	0.0160	0.6122	0.0301	0.6000	0.0276	0.6806	0.0208
XGLM 7.5B	0.6884	0.7940	0.0084	0.6506	0.0527	0.6496	0.0154	0.6768	0.0289	0.6349	0.0272	0.7242	0.0199
YaYi 7B	0.6979	0.8404	0.0076	0.7590	0.0472	0.5808	0.0159	0.7224	0.0277	0.5587	0.0280	0.7262	0.0199
Llama 3 8B	0.7684	0.8680	0.0070	0.7108	0.0501	0.7529	0.0139	0.7985	0.0248	0.7143	0.0255	0.7659	0.0189
Llama 3.1 8B	0.7552	0.8757	0.0068	0.6627	0.0522	0.7529	0.0139	0.8061	0.0244	0.6762	0.0264	0.7579	0.0191
Gemma 7B	0.7741	0.8800	0.0067	0.7349	0.0487	0.7539	0.0139	0.7833	0.0255	0.7206	0.0253	0.7718	0.0187
Gemma 2 9B	0.8007	0.8942	0.0064	0.7590	0.0472	0.7977	0.0130	0.8289	0.0233	0.7429	0.0247	0.7817	0.0184
Qwen 2 7B	0.7644	0.8632	0.0071	0.6988	0.0507	0.6882	0.0150	0.7871	0.0253	0.7175	0.0254	0.8313	0.0167
Qwen 1.5 7B	0.7259	0.8331	0.0077	0.6988	0.0507	0.6590	0.0153	0.7034	0.0282	0.6635	0.0267	0.7976	0.0179
EMMA-500 Llama 2 7B	0.7280	0.8245	0.0079	0.7229	0.0494	0.7164	0.0146	0.6920	0.0285	0.6921	0.0261	0.7202	0.0200

D.2 Code Generation

Table 24: Pass@1 on Multipl-E.

Model	C++	C#	Java	JavaScript	Python	Rust	TypeScript
Llama 2 7B	6.74%	5.65%	8.54%	11.34%	11.98%	6.12%	12.04%
CodeLlama 2 7B	26.70%	20.44%	30.56%	32.89%	28.76%	26.23%	33.45%
LLaMAX 2 7B	0.0%	0.0%	0.02%	0.68%	0.0%	0.0%	1.78%
MaLA-500 Llama 2 10B V2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
TowerBase Llama 2 7B	1.01%	6.85%	7.86%	0.87%	8.68%	0.0%	0.0%
Occiglot Mistral 7B v0.1	23.16%	16.14%	19.83%	26.92%	21.45%	16.34%	24.97%
Bloom 7B	5.25%	2.97%	6.37%	6.76%	7.65%	1.01%	7.34%
BloomZ 2 7B	6.87%	3.59%	6.02%	6.96%	7.33%	2.13%	8.04%
Aya23 8B	16.03%	7.91%	14.29%	6.23%	3.56%	11.43%	4.90%
Mistral 7B v0.3	26.12%	22.87%	25.54%	35.24%	24.91%	19.24%	28.76%
Llama 3 8B	34.12%	21.06%	26.56%	36.12%	30.22%	25.19%	37.34%
LLaMAX 3 8B	0.0%	0.0%	0.0%	9.73%	0.91%	0.21%	10.12%
Gemma 7B	29.66%	21.40%	27.35%	35.29%	30.11%	25.48%	30.57%
CodeGemma 7B	33.91%	21.05%	29.43%	37.78%	32.56%	28.70%	37.14%
Qwen 1.5 7B	22.04%	12.42%	17.68%	27.58%	32.11%	8.32%	27.21%
Qwen 2 7B	43.51%	21.47%	38.95%	46.31%	37.49%	34.61%	48.41%
EMMA-500 Llama 2 7B	11.34%	8.94%	11.93%	11.67%	18.97%	6.86%	9.94%

Table 25: Pass@10 on Multipl-E.

Model	C++	C#	Java	JavaScript	Python	Rust	TypeScript
Llama 2 7B	17.92%	14.22%	20.78%	23.12%	24.86%	13.08%	22.19%
CodeLlama 2 7B	51.39%	37.13%	50.38%	59.08%	56.58%	47.68%	53.60%
LLaMAX 2 7B	0.96%	0.0%	0.53%	3.99%	0.0%	0.12%	5.64%
MaLA-500 Llama 2 10B V2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
TowerBase Llama 2 7B	1.01%	6.85%	7.86%	0.87%	8.68%	0.0%	0.0%
Occiglot Mistral 7B v0.1	36.08%	27.49%	35.87%	47.45%	41.76%	29.80%	43.13%
Bloom 7B	11.35%	6.86%	12.79%	12.44%	14.24%	3.68%	12.08%
BloomZ 2 7B	12.55%	8.21%	12.94%	14.10%	14.84%	3.91%	13.27%
Aya23 8B	28.49%	17.34%	27.12%	23.19%	26.72%	26.14%	15.67%
Mistral 7B v0.3	50.23%	35.18%	46.83%	57.23%	54.29%	42.77%	54.22%
Llama 3 8B	55.13%	36.67%	54.34%	62.65%	59.24%	45.68%	63.02%
LLaMAX 3 8B	0.72%	0.0%	0.97%	22.91%	1.58%	1.38%	23.04%
Gemma 7B	55.21%	39.09%	52.02%	61.88%	60.09%	50.34%	61.23%
CodeGemma 7B	62.29%	45.11%	59.74%	70.96%	69.76%	61.27%	72.76%
Qwen 1.5 7B	40.11%	28.95%	37.56%	48.35%	40.34%	20.19%	44.85%
Qwen 2 7B	64.58%	43.17%	63.28%	73.21%	54.34%	65.48%	74.32%
EMMA-500 Llama 2 7B	22.84%	14.29%	21.28%	18.22%	24.94%	15.68%	15.91%

Table 26: Pass@25 on Multipl-E.

Model	C++	C#	Java	JavaScript	Python	Rust	TypeScript
Llama 2 7B	24.89%	18.77%	26.45%	30.31%	31.09%	18.70%	29.55%
CodeLlama 2 7B	64.12%	45.96%	61.98%	72.76%	71.77%	59.98%	70.89%
LLaMAX 2 7B	1.47%	0.0%	1.04%	6.78%	0.0%	0.34%	9.04%
MaLA-500 Llama 2 10B V2	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
TowerBase Llama 2 7B	1.86%	15.74%	22.45%	2.53%	20.18%	0.0%	0.0%
Occiglot Mistral 7B v0.1	44.17%	33.89%	42.77%	56.63%	49.87%	37.94%	55.76%
Bloom 7B	17.27%	10.07%	17.23%	17.55%	17.93%	5.70%	16.81%
BloomZ 2 7B	17.41%	11.98%	18.66%	18.37%	18.78%	6.83%	18.31%
Aya23 8B	37.56%	21.87%	36.45%	33.57%	36.09%	35.34%	23.75%
Mistral 7B v0.3	60.93%	44.80%	56.12%	66.34%	65.33%	55.31%	64.51%
Llama 3 8B	64.42%	43.34%	62.14%	73.51%	71.82%	57.78%	75.09%
LLaMAX 3 8B	1.71%	0.0%	2.32%	35.61%	2.77%	2.08%	30.19%
Gemma 7B	66.14%	47.08%	63.24%	70.22%	70.47%	63.52%	72.56%
CodeGemma 7B	73.66%	50.79%	69.96%	79.48%	78.68%	74.87%	81.12%
Qwen 1.5 7B	51.87%	37.69%	48.71%	59.71%	49.69%	29.78%	55.98%
Qwen 2 7B	74.33%	51.87%	72.03%	81.67%	65.38%	74.97%	80.59%
EMMA-500 Llama 2 7B	31.93%	18.69%	29.85%	26.45%	32.55%	21.32%	22.34%

D.3 Natural Language Inference

Table 27: 0-shot results (ACC) on XNLI in all languages.

Model	Avg	ar-acc	stderr	bg-acc	stderr	de-acc	stderr	el-acc	stderr	en-acc	stderr	es-acc	stderr	fr-acc	stderr	hi-acc	stderr	ru-acc	stderr	sw-acc	stderr	th-acc	stderr	tr-acc	stderr	ur-acc	stderr	vi-acc	stderr	zh-acc	stderr
Llama 2 7B	0.4019	0.3542	0.0096	0.4265	0.0099	0.4711	0.0100	0.3667	0.0097	0.5530	0.0100	0.4052	0.0098	0.5008	0.0100	0.3771	0.0097	0.4237	0.0099	0.3494	0.0096	0.3635	0.0096	0.3727	0.0097	0.3361	0.0095	0.3663	0.0097	0.3618	0.0096
Llama 2 7B Chat	0.3858	0.3442	0.0095	0.3707	0.0097	0.4309	0.0099	0.3815	0.0097	0.5024	0.0100	0.3944	0.0098	0.4482	0.0100	0.3578	0.0096	0.4209	0.0099	0.3422	0.0095	0.3349	0.0095	0.3695	0.0097	0.3390	0.0095	0.3811	0.0097	0.3695	0.0097
CodeLlama 2 7B	0.4019	0.3341	0.0095	0.3775	0.0097	0.4723	0.0100	0.3763	0.0097	0.5478	0.0100	0.4438	0.0100	0.4920	0.0100	0.3594	0.0096	0.4606	0.0100	0.3329	0.0094	0.3502	0.0096	0.3859	0.0098	0.3325	0.0094	0.4040	0.0098	0.3594	0.0096
LLaMAX Llama 2 7B	0.4427	0.3378	0.0095	0.4683	0.0100	0.4896	0.0100	0.4257	0.0099	0.5490	0.0100	0.4759	0.0100	0.4779	0.0100	0.4550	0.0100	0.4554	0.0100	0.4305	0.0099	0.4185	0.0099	0.4329	0.0099	0.4418	0.0100	0.4386	0.0099	0.3438	0.0095
LLaMAX Llama 2 7B Alpaca	0.4509	0.3442	0.0095	0.4639	0.0100	0.4976	0.0100	0.4341	0.0099	0.5811	0.0099	0.4896	0.0100	0.5197	0.0100	0.4562	0.0100	0.4627	0.0100	0.4357	0.0099	0.4080	0.0099	0.4386	0.0099	0.4430	0.0100	0.4309	0.0099	0.3578	0.0096
MaLA-500 Llama 2 10B v1	0.3811	0.3594	0.0096	0.4120	0.0099	0.4751	0.0100	0.3446	0.0095	0.5618	0.0099	0.3410	0.0095	0.4759	0.0100	0.3365	0.0095	0.3394	0.0095	0.3522	0.0096	0.3369	0.0095	0.3382	0.0095	0.3502	0.0096	0.3602	0.0096	0.3325	0.0094
MaLA-500 Llama 2 10B v2	0.3811	0.3594	0.0096	0.4120	0.0099	0.4751	0.0100	0.3446	0.0095	0.5618	0.0099	0.3410	0.0095	0.4759	0.0100	0.3365	0.0095	0.3394	0.0095	0.3522	0.0096	0.3369	0.0095	0.3382	0.0095	0.3502	0.0096	0.3602	0.0096	0.3325	0.0094
YaYi Llama 2 7B	0.4128	0.3414	0.0095	0.4261	0.0099	0.4884	0.0100	0.3735	0.0097	0.5647	0.0099	0.4578	0.0100	0.5104	0.0100	0.3948	0.0098	0.4627	0.0100	0.3570	0.0096	0.3562	0.0096	0.3936	0.0098	0.3349	0.0095	0.3751	0.0097	0.3554	0.0096
TowerBase Llama 2 7B	0.3984	0.3390	0.0095	0.4137	0.0099	0.4787	0.0100	0.3526	0.0096	0.5635	0.0099	0.4169	0.0099	0.4944	0.0100	0.3454	0.0095	0.4594	0.0100	0.3502	0.0096	0.3478	0.0095	0.3574	0.0096	0.3337	0.0095	0.3719	0.0097	0.3522	0.0096
TowerInstruct Llama 2 7B	0.4036	0.3365	0.0095	0.4293	0.0099	0.4884	0.0100	0.3498	0.0096	0.5695	0.0099	0.4651	0.0100	0.4643	0.0100	0.3474	0.0095	0.4627	0.0100	0.3394	0.0095	0.3390	0.0095	0.3787	0.0097	0.3353	0.0095	0.3735	0.0097	0.3747	0.0097
Occiglot Mistral 7B v0.1	0.4235	0.3386	0.0095	0.4137	0.0099	0.5177	0.0100	0.3771	0.0097	0.5586	0.0100	0.5165	0.0100	0.5193	0.0100	0.3574	0.0096	0.4763	0.0100	0.3470	0.0095	0.3739	0.0097	0.4301	0.0099	0.3349	0.0095	0.3863	0.0098	0.4056	0.0098
Occiglot Mistral 7B v0.1 Instruct	0.4081	0.3438	0.0095	0.3884	0.0098	0.5084	0.0100	0.4000	0.0098	0.5566	0.0100	0.4863	0.0100	0.5169	0.0100	0.3430	0.0095	0.4004	0.0098	0.3317	0.0094	0.3635	0.0096	0.3799	0.0097	0.3406	0.0095	0.3759	0.0097	0.3863	0.0098
BLOOM 7B	0.4160	0.3385	0.0067	0.3992	0.0069	0.3978	0.0069	0.3537	0.0068	0.5397	0.0070	0.4882	0.0071	0.4980	0.0071	0.4651	0.0070	0.4303	0.0070	0.3788	0.0069	0.3505	0.0067	0.3509	0.0067	0.4220	0.0070	0.4739	0.0071	0.3535	0.0068
BLOOMZ 7B	0.3713	0.3269	0.0094	0.3402	0.0095	0.4169	0.0099	0.3582	0.0096	0.4687	0.0100	0.3602	0.0096	0.4305	0.0099	0.4040	0.0098	0.3747	0.0097	0.3361	0.0095	0.3309	0.0094	0.3373	0.0095	0.3683	0.0097	0.3667	0.0097	0.3502	0.0096
mGPT	0.4051	0.3382	0.0095	0.4173	0.0099	0.4478	0.0100	0.3542	0.0096	0.4936	0.0100	0.4281	0.0099	0.4406	0.0100	0.4133	0.0099	0.4321	0.0099	0.4189	0.0099	0.3578	0.0096	0.4012	0.0098	0.3422	0.0095	0.4550	0.0100	0.3361	0.0095
XGLM 7.5B	0.4375	0.3349	0.0095	0.4365	0.0099	0.4755	0.0100	0.4040	0.0098	0.5309	0.0100	0.4707	0.0100	0.4542	0.0100	0.4574	0.0100	0.4598	0.0100	0.4570	0.0100	0.4137	0.0099	0.4679	0.0100	0.4193	0.0099	0.4289	0.0099	0.3522	0.0096
YaYi 7B	0.3987	0.3980	0.0098	0.3578	0.0096	0.4261	0.0099	0.3647	0.0096	0.5052	0.0100	0.4771	0.0100	0.4819	0.0100	0.4004	0.0098	0.3912	0.0098	0.3406	0.0095	0.3434	0.0095	0.3317	0.0094	0.3707	0.0097	0.4418	0.0100	0.3494	0.0096
Llama 3 8B	0.4497	0.3365	0.0095	0.4534	0.0100	0.5048	0.0100	0.3928	0.0098	0.5502	0.0100	0.4952	0.0100	0.5056	0.0100	0.4755	0.0100	0.4916	0.0100	0.3892	0.0098	0.4627	0.0100	0.4823	0.0100	0.3349	0.0095	0.4900	0.0100	0.3815	0.0097
Llama 3.1 8B	0.4562	0.3386	0.0095	0.4558	0.0100	0.5145	0.0100	0.3896	0.0098	0.5522	0.0100	0.5028	0.0100	0.5177	0.0100	0.4940	0.0100	0.4916	0.0100	0.3936	0.0098	0.4807	0.0100	0.4932	0.0100	0.3506	0.0096	0.4711	0.0100	0.3976	0.0098
Gemma 7B	0.4258	0.3349	0.0095	0.4349	0.0099	0.4863	0.0100	0.3811	0.0097	0.5205	0.0100	0.4414	0.0100	0.4976	0.0100	0.4434	0.0100	0.4739	0.0100	0.4064	0.0098	0.3795	0.0097	0.4325	0.0099	0.3542	0.0096	0.4329	0.0099	0.3667	0.0097
Gemma 2 9B	0.4674	0.3418	0.0095	0.4952	0.0100	0.5137	0.0100	0.4325	0.0099	0.5345	0.0100	0.5141	0.0100	0.5229	0.0100	0.4731	0.0100	0.4956	0.0100	0.4558	0.0100	0.4988	0.0100	0.5072	0.0100	0.4402	0.0100	0.4570	0.0100	0.3293	0.0094
Qwen 2 7B	0.4277	0.3369	0.0095	0.4546	0.0100	0.4819	0.0100	0.3683	0.0097	0.5426	0.0100	0.4723	0.0100	0.5141	0.0100	0.4498	0.0100	0.4739	0.0100	0.3731	0.0097	0.3827	0.0097	0.4353	0.0099	0.3402	0.0095	0.4357	0.0099	0.3538	0.0096
Qwen 1.5 7B	0.3947	0.3434	0.0095	0.4092	0.0099	0.4277	0.0099	0.3618	0.0096	0.4908	0.0100	0.3787	0.0097	0.4313	0.0099	0.3815	0.0097	0.3888	0.0098	0.3542	0.0096	0.4422	0.0100	0.3884	0.0098	0.3386	0.0095	0.4438	0.0100	0.3398	0.0095
EMMA-500 Llama 2 7B	0.4514	0.3478	0.0095	0.4627	0.0100	0.4707	0.0100	0.4586	0.0100	0.5378	0.0100	0.4707	0.0100	0.4687	0.0100	0.4759	0.0100	0.4655	0.0100	0.4618	0.0100	0.4197	0.0099	0.4486	0.0100	0.4598	0.0100	0.4703	0.0100	0.3522	0.0096

D.4 Math

Tables 28 and 29 show 3-shot results on MGSM by direct and Chain-of-Thought prompting respectively. All the scores are obtained by flexible matching.

Table 28: 3-shot results (ACC) on MGSM by direct prompting and flexible matching.

Model	Avg	bn	bn-stderr	de	de-stderr	en	en-stderr	es	es-stderr	fr	fr-stderr	ja	ja-stderr	ru	ru-stderr	sw	sw-stderr	te	te-stderr	th	th-stderr	zh	zh-stderr
Llama 2 7B	0.0669	0.0280	0.0105	0.0800	0.0172	0.1760	0.0241	0.1120	0.0200	0.1200	0.0206	0.0240	0.0097	0.0800	0.0172	0.0280	0.0105	0.0120	0.0069	0.0080	0.0056	0.0680	0.0160
Llama 2 7B Chat	0.1022	0.0280	0.0105	0.1680	0.0237	0.2280	0.0266	0.1960	0.0252	0.1920	0.0250	0.0240	0.0097	0.1440	0.0222	0.0080	0.0056	0.0080	0.0056	0.0280	0.0105	0.1000	0.0190
CodeLlama 2 7B	0.0593	0.0160	0.0080	0.0880	0.0180	0.1280	0.0212	0.0880	0.0180	0.1040	0.0193	0.0560	0.0146	0.0600	0.0151	0.0200	0.0089	0.0120	0.0069	0.0520	0.0141	0.0280	0.0105
LLaMAX Llama 2 7B	0.0335	0.0280	0.0105	0.0360	0.0118	0.0600	0.0151	0.0200	0.0089	0.0720	0.0164	0.0320	0.0112	0.0240	0.0097	0.0160	0.0080	0.0080	0.0056	0.0160	0.0080	0.0560	0.0146
LLaMAX Llama 2 7B Alpaca	0.0505	0.0400	0.0124	0.0360	0.0118	0.1080	0.0197	0.0600	0.0151	0.0640	0.0155	0.0320	0.0112	0.0440	0.0130	0.0480	0.0135	0.0160	0.0080	0.0320	0.0112	0.0760	0.0168
MaLA-500 Llama 2 10B v1	0.0091	0.0000	0.0000	0.0000	0.0000	0.0120	0.0069	0.0200	0.0089	0.0240	0.0097	0.0120	0.0069	0.0200	0.0089	0.0040	0.0040	0.0000	0.0000	0.0040	0.0040	0.0040	0.0040
MaLA-500 Llama 2 10B v2	0.0091	0.0000	0.0000	0.0000	0.0000	0.0120	0.0069	0.0200	0.0089	0.0240	0.0097	0.0120	0.0069	0.0200	0.0089	0.0040	0.0040	0.0000	0.0000	0.0040	0.0040	0.0040	0.0040
YaYi Llama 2 7B	0.0709	0.0320	0.0112	0.0840	0.0176	0.1560	0.0230	0.1600	0.0232	0.1040	0.0193	0.0520	0.0141	0.0560	0.0146	0.0080	0.0056	0.0120	0.0069	0.0040	0.0040	0.1120	0.0200
TowerBase Llama 2 7B	0.0615	0.0240	0.0097	0.0840	0.0176	0.1160	0.0203	0.0920	0.0183	0.0880	0.0180	0.0480	0.0135	0.1000	0.0190	0.0120	0.0069	0.0080	0.0056	0.0160	0.0080	0.0880	0.0180
TowerInstruct Llama 2 7B	0.0724	0.0160	0.0080	0.1000	0.0190	0.1520	0.0228	0.1560	0.0230	0.1280	0.0212	0.0160	0.0080	0.1000	0.0190	0.0160	0.0080	0.0200	0.0089	0.0200	0.0089	0.0720	0.0164
Occiglot Mistral 7B v0.1	0.1331	0.0320	0.0112	0.2120	0.0259	0.3000	0.0290	0.2720	0.0282	0.2160	0.0261	0.0640	0.0155	0.1520	0.0228	0.0240	0.0097	0.0160	0.0080	0.0800	0.0172	0.0960	0.0187
Occiglot Mistral 7B v0.1 Instruct	0.2276	0.0480	0.0135	0.3400	0.0300	0.4640	0.0316	0.4000	0.0310	0.3160	0.0295	0.1840	0.0246	0.2360	0.0269	0.0640	0.0155	0.0280	0.0105	0.1120	0.0200	0.3120	0.0294
BLOOM 7B	0.0287	0.0240	0.0097	0.0160	0.0080	0.0400	0.0124	0.0360	0.0118	0.0120	0.0069	0.0200	0.0089	0.0360	0.0118	0.0400	0.0124	0.0360	0.0118	0.0200	0.0089	0.0360	0.0118
BLOOMZ 7B	0.0255	0.0320	0.0112	0.0160	0.0080	0.0360	0.0118	0.0240	0.0097	0.0320	0.0112	0.0280	0.0105	0.0360	0.0118	0.0280	0.0105	0.0120	0.0069	0.0120	0.0069	0.0240	0.0097
mGPT	0.0135	0.0000	0.0000	0.0200	0.0089	0.0320	0.0112	0.0080	0.0056	0.0240	0.0097	0.0160	0.0080	0.0240	0.0097	0.0160	0.0080	0.0000	0.0000	0.0000	0.0000	0.0080	0.0056
mGPT 13B	0.0131	0.0000	0.0000	0.0160	0.0080	0.0160	0.0080	0.0080	0.0056	0.0360	0.0118	0.0160	0.0080	0.0200	0.0089	0.0120	0.0069	0.0000	0.0000	0.0080	0.0056	0.0120	0.0069
XGLM 7.5B	0.0102	0.0000	0.0000	0.0080	0.0056	0.0000	0.0000	0.0120	0.0069	0.0000	0.0000	0.0040	0.0040	0.0200	0.0089	0.0200	0.0089	0.0000	0.0000	0.0080	0.0000	0.0280	0.0105
YaYi 7B	0.0276	0.0240	0.0097	0.0200	0.0089	0.0600	0.0151	0.0240	0.0097	0.0560	0.0146	0.0160	0.0080	0.0120	0.0069	0.0120	0.0069	0.0240	0.0097	0.0080	0.0056	0.0480	0.0135
Llama 3 8B	0.2745	0.1760	0.0241	0.3960	0.0310	0.5080	0.0317	0.4720	0.0316	0.3640	0.0305	0.0360	0.0118	0.3760	0.0307	0.2400	0.0271	0.0560	0.0146	0.3680	0.0306	0.0280	0.0105
Llama 3.1 8B	0.2836	0.2000	0.0253	0.4120	0.0312	0.5520	0.0315	0.4840	0.0317	0.3840	0.0308	0.0360	0.0118	0.4000	0.0310	0.2080	0.0257	0.0440	0.0130	0.3760	0.0307	0.0240	0.0097
Gemma 7B	0.3822	0.3440	0.0301	0.4480	0.0315	0.5880	0.0312	0.4800	0.0317	0.3960	0.0310	0.1680	0.0237	0.4120	0.0312	0.3760	0.0307	0.2720	0.0282	0.4280	0.0314	0.2920	0.0288
Gemma 2 9B	0.3295	0.2960	0.0289	0.4040	0.0311	0.5640	0.0314	0.5080	0.0317	0.3600	0.0304	0.0120	0.0069	0.3520	0.0303	0.3760	0.0307	0.3080	0.0293	0.4000	0.0310	0.0440	0.0130
Qwen 2 7B	0.4895	0.4440	0.0315	0.6560	0.0301	0.8080	0.0250	0.7600	0.0271	0.6960	0.0292	0.0160	0.0080	0.6720	0.0298	0.1600	0.0232	0.1480	0.0225	0.5600	0.0315	0.4640	0.0316
Qwen 1.5 7B	0.3156	0.1240	0.0209	0.4400	0.0315	0.5520	0.0315	0.4840	0.0317	0.4520	0.0315	0.1680	0.0237	0.4320	0.0314	0.0720	0.0164	0.0400	0.0124	0.2840	0.0286	0.4240	0.0313
EMMA-500 Llama 2 7B	0.1702	0.0880	0.0180	0.2320	0.0268	0.3400	0.0300	0.2800	0.0285	0.2560	0.0277	0.0920	0.0183	0.2280	0.0266	0.1680	0.0237	0.0240	0.0097	0.1000	0.0190	0.0640	0.0155

Table 29: 3-shot results (ACC) on MGSM by CoT prompting and flexible matching.

Model	Avg	bn	bn-stderr	de	de-stderr	en	en-stderr	es	es-stderr	fr	fr-stderr	ja	ja-stderr	ru	ru-stderr	sw	sw-stderr	te	te-stderr	th	th-stderr	zh	zh-stderr
Llama 2 7B	0.0636	0.0200	0.0089	0.0760	0.0168	0.1600	0.0232	0.1240	0.0209	0.0920	0.0183	0.0360	0.0118	0.0800	0.0172	0.0200	0.0089	0.0080	0.0056	0.0160	0.0080	0.0800	0.0172
Llama 2 7B Chat	0.1091	0.0240	0.0097	0.1760	0.0241	0.2720	0.0282	0.1920	0.0250	0.1880	0.0248	0.0400	0.0124	0.1520	0.0228	0.0200	0.0089	0.0080	0.0056	0.0280	0.0105	0.1160	0.0203
CodeLlama 2 7B	0.0664	0.0160	0.0080	0.0920	0.0183	0.1320	0.0215	0.1080	0.0197	0.1080	0.0197	0.0440	0.0130	0.0760	0.0168	0.0120	0.0069	0.0160	0.0080	0.0600	0.0151	0.0400	0.0124
LLaMAX Llama 2 7B	0.0362	0.0360	0.0118	0.0360	0.0118	0.0720	0.0164	0.0360	0.0118	0.0520	0.0141	0.0280	0.0105	0.0280	0.0105	0.0080	0.0056	0.0080	0.0056	0.0200	0.0089	0.0480	0.0135
LLaMAX Llama 2 7B Alpaca	0.0635	0.0320	0.0112	0.0480	0.0135	0.1520	0.0228	0.0960	0.0187	0.0520	0.0141	0.0440	0.0130	0.0360	0.0118	0.0480	0.0135	0.0040	0.0040	0.0480	0.0135	0.0680	0.0160
MaLA-500 Llama 2 10B v1	0.0073	0.0000	0.0000	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040	0.0160	0.0080	0.0160	0.0080	0.0240	0.0097	0.0040	0.0040	0.0000	0.0000	0.0000	0.0000	0.0080	0.0056
MaLA-500 Llama 2 10B v2	0.0073	0.0000	0.0000	0.0040	0.0040	0.0040	0.0040	0.0040	0.0040	0.0160	0.0080	0.0160	0.0080	0.0240	0.0097	0.0040	0.0040	0.0000	0.0000	0.0000	0.0000	0.0080	0.0056
YaYi Llama 2 7B	0.0722	0.0280	0.0105	0.0840	0.0176	0.1680	0.0237	0.1240	0.0209	0.1040	0.0193	0.0480	0.0135	0.0720	0.0164	0.0200	0.0089	0.0120	0.0069	0.0320	0.0112	0.1240	0.0209
TowerBase Llama 2 7B	0.0616	0.0360	0.0118	0.0800	0.0172	0.1120	0.0200	0.0840	0.0176	0.0760	0.0168	0.0360	0.0118	0.0840	0.0176	0.0120	0.0069	0.0080	0.0056	0.0280	0.0105	0.0920	0.0183
TowerInstruct Llama 2 7B	0.0824	0.0120	0.0069	0.1280	0.0212	0.1880	0.0248	0.1520	0.0228	0.1200	0.0206	0.0240	0.0097	0.1360	0.0217	0.0160	0.0080	0.0120	0.0069	0.0320	0.0112	0.1080	0.0197
Occiglot Mistral 7B v0.1	0.1407	0.0240	0.0097	0.2160	0.0261	0.3320	0.0298	0.2680	0.0281	0.2240	0.0264	0.0640	0.0155	0.1720	0.0239	0.0320	0.0112	0.0120	0.0069	0.0600	0.0151	0.1120	0.0200
Occiglot Mistral 7B v0.1 Instruct	0.2216	0.0400	0.0124	0.3320	0.0298	0.4320	0.0314	0.4200	0.0313	0.3160	0.0295	0.1680	0.0237	0.2440	0.0272	0.0560	0.0146	0.0160	0.0080	0.0840	0.0176	0.2480	0.0274
BLOOM 7B	0.0229	0.0200	0.0089	0.0160	0.0080	0.0480	0.0135	0.0240	0.0097	0.0240	0.0097	0.0040	0.0040	0.0440	0.0130	0.0160	0.0080	0.0200	0.0089	0.0200	0.0089	0.0240	0.0097
BLOOMZ 7B	0.0215	0.0200	0.0089	0.0200	0.0089	0.0240	0.0097	0.0320	0.0112	0.0320	0.0112	0.0200	0.0089	0.0160	0.0080	0.0240	0.0097	0.0200	0.0089	0.0160	0.0080	0.0120	0.0069
mGPT	0.0142	0.0000	0.0000	0.0200	0.0089	0.0200	0.0089	0.0160	0.0080	0.0240	0.0097	0.0120	0.0069	0.0200	0.0089	0.0280	0.0105	0.0000	0.0000	0.0000	0.0000	0.0160	0.0080
mGPT 13B	0.0153	0.0000	0.0000	0.0200	0.0089	0.0360	0.0118	0.0160	0.0080	0.0240	0.0097	0.0120	0.0069	0.0160	0.0080	0.0080	0.0056	0.0040	0.0040	0.0080	0.0056	0.0200	0.0089
XGLM 7.5B	0.0116	0.0000	0.0000	0.0120	0.0069	0.0000	0.0000	0.0080	0.0056	0.0000	0.0000	0.0040	0.0040	0.0160	0.0080	0.0200	0.0089	0.0240	0.0097	0.0160	0.0080	0.0280	0.0105
YaYi 7B	0.0302	0.0480	0.0135	0.0320	0.0112	0.0480	0.0135	0.0360	0.0118	0.0400	0.0124	0.0200	0.0089	0.0120	0.0069	0.0160	0.0080	0.0280	0.0105	0.0080	0.0056	0.0600	0.0151
Llama 3 8B	0.2813	0.1920	0.0250	0.3840	0.0308	0.5400	0.0316	0.4600	0.0316	0.3600	0.0304	0.0200	0.0089	0.3800	0.0308	0.2520	0.0275	0.0640	0.0155	0.3920	0.0309	0.0400	0.0124
Llama 3.1 8B	0.2731	0.1920	0.0250	0.4200	0.0313	0.5280	0.0316	0.4280	0.0314	0.3480	0.0302	0.0280	0.0105	0.3720	0.0306	0.2520	0.0275	0.0560	0.0146	0.3560	0.0303	0.0160	0.0080
Gemma 7B	0.3578	0.3640	0.0305	0.4080	0.0311	0.6080	0.0309	0.4520	0.0315	0.4200	0.0313	0.0360	0.0118	0.3920	0.0309	0.3880	0.0309	0.2480	0.0274	0.4400	0.0315	0.0480	0.0135
Gemma 2 9B	0.4469	0.4880	0.0317	0.5360	0.0316	0.7200	0.0285	0.6040	0.0310	0.5000	0.0317	0.0200	0.0089	0.5000	0.0317	0.5360	0.0316	0.4440	0.0315	0.5520	0.0315	0.0160	0.0080
Qwen 2 7B	0.5147	0.4520	0.0315	0.6600	0.0300	0.8200	0.0243	0.7520	0.0274	0.7160	0.0286	0.0280	0.0105	0.6840	0.0295	0.1720	0.0239	0.1760	0.0241	0.5840	0.0312	0.6080	0.0309
Qwen 1.5 7B	0.3036	0.1280	0.0212	0.4320	0.0314	0.5840	0.0312	0.5040	0.0317	0.4200	0.0313	0.1120	0.0200	0.3680	0.0306	0.0880	0.0180	0.0400	0.0124	0.2560	0.0277	0.2600	0.0278
EMMA-500 Llama 2 7B	0.1809	0.0840	0.0176	0.2240	0.0264	0.3760	0.0307	0.2560	0.0277	0.2160	0.0261	0.0800	0.0172	0.2280	0.0266	0.2120	0.0259	0.0240	0.0097	0.1160	0.0203	0.1640	0.0235

Chapter 5

A Recipe of Parallel Corpora Exploitation for Multilingual Large Language Models

A Recipe of Parallel Corpora Exploitation for Multilingual Large Language Models

Peiqin Lin[1,2], **André F. T. Martins**[3,4,5], **Hinrich Schütze**[1,2]
[1]Center for Information and Language Processing, LMU Munich
[2]Munich Center for Machine Learning
[3]Instituto Superior Técnico, Universidade de Lisboa (Lisbon ELLIS Unit)
[4]Instituto de Telecomunicações [5]Unbabel
linpq@cis.lmu.de

Abstract

Recent studies have highlighted the potential of exploiting parallel corpora to enhance multilingual large language models in both bilingual tasks, e.g., machine translation, and general-purpose tasks, e.g., text classification. Building upon these findings, our comprehensive study aims to identify the most effective strategies for leveraging parallel corpora. We investigate the impact of parallel corpus quality and quantity, training objectives, and model size on the performance of multilingual large language models enhanced with parallel corpora across diverse languages and tasks. Our analysis reveals several key insights: (i) filtering noisy translations is essential for exploiting parallel corpora, while language identification and short sentence filtering have little effect; (ii) even a corpus with just 10K parallel sentences can yield results comparable to those obtained from larger datasets; (iii) employing only the machine translation objective yields the best results among various training objectives and their combinations; (iv) larger multilingual language models benefit more from parallel corpora than smaller models. Our study offers valuable insights into the optimal utilization of parallel corpora to enhance multilingual large language models, extending the generalizability of previous findings from limited languages and tasks to a broader range of scenarios.

1 Introduction

Recent multilingual large language models (mLLMs), represented by BLOOM (Scao et al., 2022), MaLA500 (Lin et al., 2024b), and Aya (Üstün et al., 2024), have shown impressive capacity on diverse tasks across languages. Parallel corpora have emerged as crucial resources for enhancing mLLMs, both for specific tasks, e.g., machine translation (Xu et al., 2023; Alves et al., 2024), and for general-purpose tasks (Cahyawijaya et al., 2023; Zhu et al., 2023; Li et al., 2023).

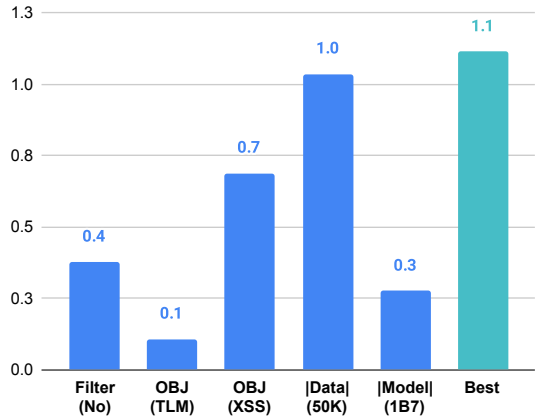


Figure 1: Average performance improvements (y-axis) achieved by mLLMs enhanced with parallel corpora compared to their base models. **Best: Instruction tuning of BLOOM-7B1 with the machine translation objective (MT) using 10K high-quality (i.e., filtered) parallel sentences yields the best results**. Main variations explored include: **Filter (No)** (using the original data); **OBJ (TLM)** (translation language modeling objective); **OBJ (XSS)** (cross-lingual semantic similarity objective); **|Data| (50K)** (a larger 50K-sentence dataset); **|Model| (1B7)** (BLOOM-1B7 model).

However, existing studies often lack in comprehensive exploration of methodologies for harnessing parallel corpora. The quality and quantity of parallel corpora remain inadequately explored, inhibiting the full potential of such resources. Moreover, the influence of different training objectives and mLLM sizes across diverse languages and tasks remains under-investigated. This limitation impedes the generalization of parallel corpus exploitation methods across varied linguistic landscapes and task domains. Therefore, this paper aims to address these gaps by presenting a comprehensive recipe for exploiting parallel corpora for mLLMs. We focus on four key factors, with some main results shown in Figure 1.

Quality: We explore three dimensions of paral-

lel corpus quality: translation accuracy, sentence length, and language identification. Our results show that *translation quality* is vital for exploiting parallel corpora, while sentence length filtering and language identification have minimal impact.

Quantity: Acquiring large amounts of high-quality parallel corpora is challenging, especially for relatively low-resource languages. Our study examines the minimum corpus size necessary to achieve performance improvements across diverse tasks. We find that even a corpus of just *10K sentences* can yield results comparable to those obtained from much larger datasets.

Objective: Previous studies (Cahyawijaya et al., 2023) have investigated the effectiveness of different training objectives and their combinations on classification tasks of Indonesian local languages, using smaller-sized mLLMs up to 1B7 parameters. We extend this investigation by examining the impact of various training objectives and their combinations on larger mLLMs across a range of languages and tasks. Our experiments demonstrate that employing *the machine translation objective* produces the most promising results.

Model Size: The size of mLLMs can greatly impact their ability to comprehend instructions derived from parallel corpora. Our findings indicate that *larger mLLMs* exhibit superior comprehension and cross-task transferability compared to their smaller counterparts. Consequently, they achieve more substantial improvements across a broader spectrum of tasks.

In light of the critical role parallel corpora play in mLLMs, our study provides a comprehensive recipe for effectively exploiting parallel corpora. We have identified four primary factors: quality (§4), quantity (§5), objective (§6), and model size (§7). Our detailed analysis of these factors reveals their great impact on mLLM performance across diverse languages and tasks. By delving into these aspects, we offer actionable insights that can inform the development and optimization of strategies for parallel corpus exploitation, ultimately contributing to the advancement of mLLMs in both bilingual and general-purpose tasks.

2 Related Work

2.1 Parallel Data for Multilingual Language Models

Over the years, multilingual language models have evolved from earlier, smaller models, such as XLM (Conneau and Lample, 2019), XLM-R (Conneau et al., 2020), and Glot500 (Imani et al., 2023), to more recent, larger models, including BLOOM (Scao et al., 2022), MaLA500 (Lin et al., 2024b), and Aya (Üstün et al., 2024). These models consistently demonstrate strong performance across various downstream tasks (Ahuja et al., 2023; Lin et al., 2024c).

Parallel corpora have played a pivotal role in both the analysis (Piqueras and Søgaard, 2022; Lin et al., 2024a) and enhancement (Conneau and Lample, 2019; Ouyang et al., 2020; Yang et al., 2020; Huang et al., 2019; Chi et al., 2021a; Wei et al., 2021; Hu et al., 2021; Chi et al., 2021b; Reid and Artetxe, 2022b; Liu et al., 2023) of small multilingual language models.

In the era of mLLMs, parallel corpora are constructed as instruction data and used to enhance mLLMs through supervised fine-tuning (Cahyawijaya et al., 2023; Zhu et al., 2023; Li et al., 2023). Specifically, Cahyawijaya et al. (2023) propose three methods of incorporating parallel corpora as instruction tuning data: Machine Translation (MT), Translation Language Modeling (TLM), and Cross-Lingual Semantic Similarity (XSS) (see §3.3). However, their evaluation is limited to small models with up to 1.7 billion parameters and focuses solely on classification tasks within Indonesian local languages. Both Zhu et al. (2023) and Li et al. (2023) propose using machine-translation-style instruction data to improve mLLMs but do not explore different training objectives. While these studies yield promising results, their scope is limited. Firstly, they do not explore critical factors such as the quality and quantity of parallel corpora, considering the high cost of collecting high-quality and massive parallel corpora, especially for relatively low-resource languages. Secondly, their investigations do not encompass an in-depth analysis of training objectives and mLLMs with varied model sizes across diverse languages and tasks.

2.2 Key Elements for Language Modeling

Previous research has extensively examined critical factors essential for the pretraining and enhancement of language models.

Quality: Kreutzer et al. (2022) conducted manual audits of prevalent monolingual and parallel corpora, revealing significant portions of low-quality data, particularly in corpora for relatively low-resource languages. Follow-up studies have investigated the impact of data quality on model per-

formance. Artetxe et al. (2022) observed that similar results on downstream tasks can be achieved regardless of the degree of quality of the corpus used for pretraining, while other studies found that the quality of parallel corpora matters for machine translation (Ranathunga et al., 2024) and general-purpose tasks (Reid and Artetxe, 2022a).

Quantity: Recent works (Chen et al., 2023; Zhou et al., 2023; Gupta et al., 2023) have focused on the impact of fine-tuning with small amounts of high-quality instruction data, such as one or a few thousand instances, showing promising performance gains in evaluation tasks. Xu et al. (2023) demonstrate that as few as 10K high-quality parallel sentences can significantly enhance machine translation performance.

Objective: Different training objectives based on parallel corpora for enhancing mLLMs can be viewed as distinct instructions. Wang et al. (2023) explore the impact of various types of instruction tuning data and find that their combination can be optimal in certain scenarios.

Model Size: Recent studies indicate that scaling up language models enhances their capability to excel in diverse and complex reasoning tasks (Wei et al., 2022, 2023; Lu et al., 2023). Follow-up studies (Wei et al., 2023) further illustrate distinct behavioral differences between larger and smaller models.

However, these factors have not yet been comprehensively explored in the context of leveraging parallel corpora to enhance mLLMs across diverse languages and tasks.

3 Setup

3.1 Language

We use three criteria for language selection. Firstly, we select languages well covered by mLLMs as our goal is to assess how parallel data enhances performance after pre-training an mLLM with monolingual data. Secondly, the selected languages should be also covered by several different evaluation benchmarks, allowing for robust evaluation across diverse downstream tasks. Lastly, we select typologically diverse languages, enabling our investigation to generalize to a wide range of relatively low-resource languages. Thus, we select five languages: Arabic (ar), Spanish (es), Hindi (hi), Vietnamese (vi) and Chinese (zh).

3.2 Data

We utilize the OPUS100 dataset (Zhang et al., 2020), an English-centric multilingual corpus, to gather parallel sentences between English (en) and each target language. The quality of OPUS100 is assessed across three dimensions:

Translation Quality Manual quality assessment of the vast amount of parallel corpora is impractical. Instead, we employ COMETKIWI (Rei et al., 2022)[1], a tool for estimating the quality of machine translation outputs across multiple languages. We set a COMETKIWI score threshold τ_c, retaining parallel corpora with scores not lower than τ_c.

Sentence Length Given the variation in character length across languages, we avoid using it as a metric for consistency. Instead, we measure sentence length by the number of tokens, as determined by the tokenizer of our chosen mLLM, BLOOM-7B1. We establish a length threshold τ_l, retaining parallel corpora where both source and target sentences contain no fewer than τ_l tokens.

Language Identification To identify sentences potentially not in the correct language, we employ GlotLID (Kargaran et al., 2023), an open-source language identification model. This language identification filter is applied to both the source and target sentences.

3.3 Training

We select the BLOOM series (Scao et al., 2022) for our investigation due to its offering of different sizes of mLLMs which well cover the five target languages under consideration.[2] The pretraining data size for the five target languages ranges from 23GB (Hindi) to 452GB (English). We explore BLOOM models of various sizes, including 7B1, 3B, and 1B7. Due to limited computational resources, we use LoRA (Hu et al., 2022), which is known for its competitive performance compared to full-parameter training (Alves et al., 2023), for instruction tuning of BLOOM. We configure the learning rate to 1×10^{-4}, weight decay to 0.1, and set the rank of LoRA to 16 based on preliminary experiment in §B. The maximum sequence length for both source and target sentences is set to 128. To maintain consistency across experiments with

[1] https://huggingface.co/Unbabel/wmt23-cometkiwi-da-xxl

[2] Additional experiments with XGLM, as presented in §A, yield conclusions consistent with those of BLOOM.

Objective	Template
MT	`Translate the following text from [SOURCE_LANG] to [TARGET_LANG].\nText: [SOURCE_TEXT]\nTranslation: [TARGET_TEXT]`
TLM	`[INPUT_TEXT]. Denoise the previous [TARGET_LANG] text to its equivalent sentence in [SOURCE_LANG]: [SOURCE_TEXT]\n[TARGET_TEXT]`
XSS	`[SOURCE_LANG] sentence: [SOURCE_TEXT]\n[TARGET_LANG] sentence: [TARGET_TEXT]\nDo the two sentences have the same meaning? [LABEL]`

Table 1: Templates of MT, TLM, and XSS for instruction data construction based on parallel corpora.

different quantities of parallel corpora, we ensure a uniform training budget of 50K parallel sentences. Specifically, we calculate the number of epochs as 50K divided by the number of sentences considered from the OPUS100 dataset. The batch size is 128, and we save the checkpoints every 20 steps.

Following Cahyawijaya et al. (2023), we construct the data for instruction tuning based on the parallel corpora by three distinct patterns: Machine Translation (MT), Translation Language Modeling (TLM), and Cross-Lingual Semantic Similarity (XSS). Table 1 presents the templates for these three objectives. Here, `[SOURCE_LANG]` and `[TARGET_LANG]` represent the language names of the source and target languages, respectively. In our study, we consider both English-to-target-language and target-language-to-English directions, where `[SOURCE_LANG]` represents English or `[TARGET_LANG]` represents English. For MT, `[SOURCE_TEXT]` and `[TARGET_TEXT]` refer to the parallel sentences in the source and target languages, respectively. For TLM, a portion of tokens in `[TARGET_TEXT]` are masked to generate `[INPUT_TEXT]`. For XSS, our objective is to predict whether parallel sentences `[SOURCE_TEXT]` and `[TARGET_TEXT]` are semantically similar, with `[LABEL]` being "Yes" or "No". Specifically, we utilize the parallel corpora as positive examples and introduce perturbations to `[TARGET_TEXT]` to construct negative examples. We consider applying the objectives both individually and in combination. We tune the model with the objective of causal language modeling without loss mask.

3.4 Evaluation

We conduct evaluation across five diverse benchmarks: FLORES (Costa-jussà et al., 2022), MUSE (Lample et al., 2018), MLQA (Lewis et al., 2020), XQUAD (Artetxe et al., 2020), and SIB (Adelani et al., 2023). A comprehensive overview of these benchmarks is available in Table 2. Our evaluation spans both classification tasks (SIB) and generation tasks (FLORES, MUSE, MLQA, and XQuAD), covering a spectrum of cross-language (FLORES, MUSE, and MLQA) and in-language tasks (XQuAD and SIB).

Dataset	Task	|Data|	Metric	I/C	C/G
FLORES	Machine Translation	1012	COMETKIWI	C	G
MUSE	Word Translation	1500	F1	C	G
MLQA	Question Answering	4918 - 5495	F1	C	G
XQuAD	Question Answering	1190	F1	I	G
SIB	Text Classification	204	Acc	I	C

Table 2: Details of evaluation benchmarks. |Data|: Number of samples for evaluation. I/C: In-language/Cross-language. C/G: Classification/Generation.

For translation tasks within FLORES and MUSE, we explore bidirectional translation: from English to other languages (en-xx) and from other languages to English (xx-en). Additionally, for MLQA, we evaluate scenarios where questions are in English and the passages and answers are in other languages (en-xx), as well as situations where questions are in other languages and the passages and answers are in English (xx-en).

To provide a thorough understanding of our evaluation procedures, we offer detailed prompts for each task in §C. In all experiments, we employ a 2-shot in-context learning approach, where the model is given two examples appended to the input query to aid in making predictions.

4 Quality

4.1 Quality of OPUS100

We measure the quality of 500K parallel sentences from OPUS100 for our five language pairs using three key metrics: translation quality, sentence length, and language identification accuracy, as illustrated in Figures 2–4.

A considerable portion of OPUS100 is of low-quality. All quality measures indicate that a large portion of OPUS100 contains low-quality data. Ap-

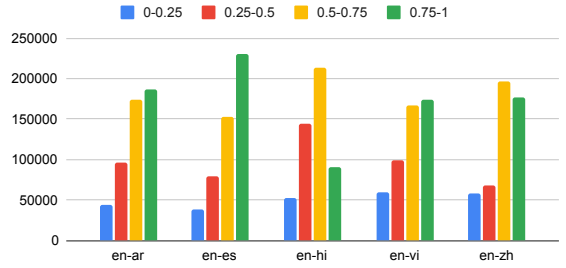


Figure 2: Translation quality measured by COMETWIKI of 500K parallel sentences from OPUS100 for our five language pairs. The COMETWIKI scores are segmented into four ranges: 0-0.25, 0.25-0.5, 0.5-0.75, and 0.75-1. Higher scores represent better translation quality.

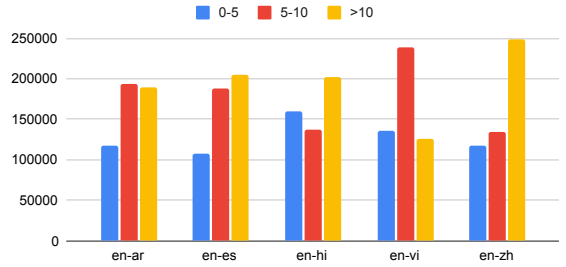


Figure 3: Sentence length of 500K parallel sentences from OPUS100 for our five language pairs. The three categories are 0-5, 5-10, greater than 10 tokens.

proximately 10% of the data has COMETKIWI scores below 0.25, indicating very poor translation quality. Additionally, between 10% to 30% of the data falls within the 0.25-0.5 score range, which is still considered sub-optimal. Regarding sentence length, we find that over 20% of the OPUS100 data consists of very short sentences, with a length of no more than five tokens. For language identification, 13% to 25% of the data is removed due to incorrect language identification results in one of the two parallel sentences.

Relatively low-resource languages suffer more from low-quality issues. For relatively low-resource languages like Hindi, there are fewer high-quality parallel sentences compared to high-resource languages such as Spanish. Analysis of translation quality indicates that the English-Hindi pair has less than 20% of parallel sentences with high COMETWIKI scores (0.75-1), whereas the English-Spanish pair has around 45%. For sentence length, the English-Hindi pair contains 10% more short sentences (0-5 tokens) compared to high-resource language pairs. Moreover, both the English-Arabic and English-Hindi pairs exhibit

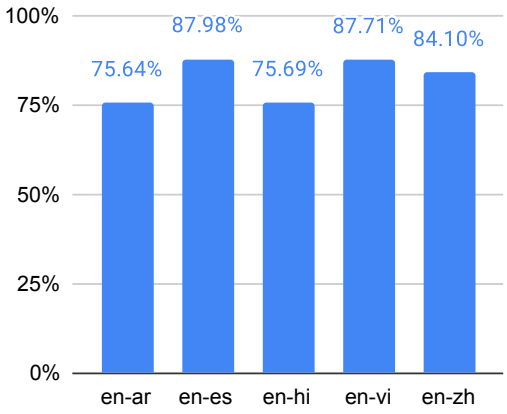


Figure 4: Percentage of sentences retained after language identification filtering of 500K parallel sentences from OPUS100 for our five language pairs.

about 10% more parallel sentences that may be in the wrong languages.

These comprehensive findings underscore the critical importance of data quality when exploiting parallel corpora for mLLM training.

4.2 Effect of Quality

Table 3 presents the performance of BLOOM-7B1 after instruction tuning with the machine translation objective, using 10K parallel corpora with various quality filtering strategies.

Parallel corpora containing noisy translations still improve results. Comparing the results of the experiment with $\tau_c = 0$ (ID 1) to the original model (ID 0), there's an average improvement of 0.4% for all tasks. The most notable improvements are observed in both bilingual tasks (en-xx) and in-language tasks. However, generating English for bilingual tasks yields degraded or marginally improved results. Experiment 1 exhibits 0.7% and 1.1% decrements in FLORES and MUSE respectively, with only a 0.3% improvement in MLQA.

Filtering out noisy translations leads to notable improvements. When $\tau_c = 0.5$, the average performance rises from 53.2% to 53.7%. Further refinement to $\tau_c = 0.75$ achieves an additional 0.3% improvement. These improvements are consistently observed across all evaluated tasks. In the optimal setting (ID 5), there's a 1.2% improvement compared to BLOOM-7B1 (ID 0). The improvements corroborate the reliability of COMETKIWI as a metric for filtering low-quality translations.

Filtering short sentences yields slightly worse results than using unfiltered data. The experi-

ID	Model			FLORES		MUSE		MLQA		XQuAD	SIB	AVG
				EN-XX	XX-EN	EN-XX	XX-EN	EN-XX	XX-EN			
0	BLOOM-7B1			69.1	**72.4**	43.1	53.7	36.4	42.7	47.2	58.1	52.8
	τ_c	τ_l	LID									
1	0	0	✓	69.7	71.7	44.4	52.6	37.8	43.0	47.7	58.8	53.2
2	0.5	0	✓	69.9	72.1	45.0	53.0	**38.1**	43.7	48.1	**59.8**	53.7
3	0.75	5	✓	70.3	72.1	**45.7**	53.6	**38.1**	43.5	47.8	59.2	53.8
4	0.75	0	✗	**70.5**	72.1	44.9	53.7	37.7	**44.0**	**48.3**	59.6	53.9
5	0.75	0	✓	70.3	72.3	45.5	**53.9**	38.0	43.9	**48.3**	59.5	**54.0**

Table 3: Performance (%) of BLOOM-7B1 after instruction tuning with the machine translation objective using 10K parallel corpora with various quality filtering strategies. Parameters include τ_c for COMETWIKI score threshold, τ_l for sentence length threshold, and LID indicating the adoption of language identification filtering.

ment with filtering short sentences (ID 3) achieves comparable or slightly worse results compared to that without filtering short sentences (ID 5). This suggests that short sentences, whether at the word or phrase level, may offer some benefits for sentence-level tasks.

Using data with language identification filtering results in only a 0.1% improvement on average. A comparison of experimental outcomes with and without language identification filtering (ID 4 and 5) reveals that using data with language identification filtering yields merely a 0.1% improvement on average. The most notable performance difference is observed in the MUSE task, where using data with language identification filtering leads to improvements of 0.6% (en-xx) and 0.2% (xx-en). This marginal enhancement may be attributed to the presence of sentences in similar languages within OPUS100, which exhibit minor linguistic variations compared to the true language. These variations could potentially have a slight negative impact on word-level translations while having little impact on sentence-level tasks.

5 Quantity

5.1 Effect of Quantity Across Tasks

Based on Table 4, which shows the performance of BLOOM-7B1 after instruction tuning with the machine translation objective using different amounts of parallel sentences, we can derive the following key findings:

Adding merely 1K parallel sentences helps. Exploiting 1K parallel sentences for instruction tuning improves the overall average score by 1%. This increase is observed across most tasks, with notable improvements in FLORES (en-xx), MUSE (en-xx), and SIB.

Using 10K parallel sentences leads to the optimal performance. The best overall performance is achieved with 10K parallel sentences, resulting in an average score of 54.0%. This setting yields the highest scores in MUSE and SIB. This aligns with the findings in Xu et al. (2023).

More data achieves comparable results. Increasing the number of parallel sentences beyond 10K results in comparable performance. Specifically, using 25K or 50K parallel sentences yields average scores of 53.9%, which are very close to the score obtained with 10K sentences.

The analysis suggests that instruction tuning with a moderate amount of parallel sentences (around 10K) yields the best overall improvement in performance for the BLOOM-7B1 model across various tasks.

5.2 Effect of Quantity Across Languages

We delve deeper into the influence of parallel corpora quantity, as depicted in Table 5.

Using 10K parallel sentences achieves optimal performance across most languages. For the majority of languages, except Vietnamese (vi) and Chinese (zh), the highest performance is obtained with 10K parallel sentences. Even for Vietnamese and Chinese, leveraging 10K parallel sentences can yield comparable results. These observations align with the findings in §5.1.

Different languages exhibit varying appetites for parallel corpora. Across most languages, increasing the number of parallel sentences used for instruction tuning generally leads to incremental improvements in performance. However, for Hindi (hi) and Chinese (zh), transitioning from 1K to 10K parallel sentences does not yield improvement.

|SENT|	FLORES		MUSE		MLQA		XQUAD	SIB	AVG
	EN-XX	XX-EN	EN-XX	XX-EN	EN-XX	XX-EN			
0	69.1	**72.4**	43.1	53.7	36.4	42.7	47.2	58.1	52.8
1K	70.0	72.2	45.3	53.6	**38.2**	43.6	47.9	59.2	53.8
5K	70.3	72.2	45.4	53.5	**38.2**	43.8	48.2	**59.5**	53.9
10K	70.3	72.3	**45.5**	**53.9**	38.0	43.9	48.3	**59.5**	**54.0**
25K	70.3	72.2	45.1	53.8	38.0	**44.0**	**48.4**	**59.5**	53.9
50K	**70.4**	72.2	45.1	53.8	38.1	43.7	48.3	**59.5**	53.9

Table 4: Task performance (%) of BLOOM-7B1 after instruction tuning with the machine translation objective using varying amounts of parallel sentences, obtained with the best filtering strategy (ID 5) as shown in Table 3. |SENT| indicates the number of parallel sentences used for instruction tuning, with |SENT|=0 representing the original BLOOM-7B1 model.

	ar	es	hi	vi	zh
0	49.5	57.7	46.5	63.8	46.7
1K	50.8	58.1	**47.7**	64.3	47.8
5K	51.2	58.2	47.6	64.6	**47.9**
10K	**51.3**	**58.4**	**47.7**	64.6	47.8
25K	51.2	58.2	**47.7**	**64.7**	47.7
50K	51.2	58.2	**47.7**	**64.7**	47.6

Table 5: Language performance (%) of BLOOM-7B1 after instruction tuning with the machine translation objective using varying amounts of parallel sentences, obtained with the best filtering strategy (ID 5) as shown in Table 3. |SENT| indicates the number of parallel sentences used for instruction tuning, with |SENT|=0 representing the original BLOOM-7B1 model.

This phenomenon may be attributed to BLOOM-7B1's limited proficiency in these languages compared to others, as reflected in the results of the original BLOOM-7B1 model (|SENT|=0).

6 Objective

We explore the effectiveness of different objectives and their combinations, with results in Table 6.

BLOOM-7B1 performs well on English generation tasks. The baseline BLOOM-7B1 model exhibits robust performance across a spectrum of evaluation tasks, notably excelling in English generation tasks such as FLORES (xx-en) and MUSE (xx-en). Further exploitation of parallel corpora fails to yield any discernible improvement.

MT emerges as the top performer. The MT objective consistently outperforms the baseline BLOOM-7B1 model, showcasing an average improvement of 1.2%. Moreover, MT achieves the highest performance in 5 out of 8 evaluated tasks.

TLM exhibits limited effectiveness. While TLM shows slight improvements on average (0.2%), primarily driven by enhancements in tasks like MUSE (en-xx), MLQA (xx-en), XQuAD, and SIB, it also leads to degradation in tasks including FLORES and MUSE (xx-en).

XSS achieves strong performance for classification. Using the XSS objective improves BLOOM-7B1 by 0.7%, though it performs 0.5% worse than MT. The major decrease is observed in translation tasks, especially from English to other languages. However, XSS can still slightly improve translation tasks compared to BLOOM-7B1. Notably, XSS achieves 0.3% better performance on SIB, highlighting its effectiveness for classification.

Combining training objectives does not provide large benefits. While combinations of different objectives can improve BLOOM-7B1 by 0.2% to 1.0%, none surpass the performance of using the MT objective alone. The combination of MT and XSS is the best among the combinations, slightly worse than MT by 0.2%, but better than all other objectives. Notably, MT +XSS achieves the best results on SIB, and TLM +XSS yields the best results on MLQA (xx-en). These observations indicate that no single objective excels across all tasks.

7 Model Size

We explore the impact of parallel corpora on various sizes of BLOOM models, detailed in Table 7.

Smaller models exhibit more pronounced improvements in FLORES. Notably, BLOOM-1B7 demonstrates larger improvements compared to its larger counterparts in the FLORES task, where the prompt is the same as the one used during instruction tuning with the MT objective. This is attributed

Model	FLORES		MUSE		MLQA		XQuAD	SIB	AVG
	EN-XX	XX-EN	EN-XX	XX-EN	EN-XX	XX-EN			
BLOOM-7B1	69.1	**72.4**	43.1	53.7	36.4	42.7	47.2	58.1	52.8
MT	**70.3**	72.3	**45.5**	**53.9**	**38.0**	43.9	**48.3**	59.5	**54.0**
TLM	67.2	72.2	44.3	53.0	36.3	44.4	47.6	58.7	53.0
XSS	69.4	72.2	43.7	53.5	37.0	44.2	**48.3**	60.0	53.5
MT +TLM	69.3	72.1	44.1	53.2	36.8	43.8	47.2	59.5	53.2
MT +XSS	70.3	72.1	44.9	53.3	37.4	44.5	47.9	**60.4**	53.8
TLM +XSS	67.7	72.2	43.0	52.5	34.9	**45.6**	48.2	60.0	53.0
MT +TLM +XSS	69.5	72.1	44.2	53.2	36.1	45.1	47.7	59.0	53.4

Table 6: Performance (%) of BLOOM-7B1 after instruction tuning with different objectives and their combinations using 10K parallel corpora, obtained with the best filtering strategy (ID 5) as shown in Table 3.

Model	FLORES		MUSE		MLQA		XQuAD	SIB	AVG
	EN-XX	XX-EN	EN-XX	XX-EN	EN-XX	XX-EN			
BLOOM-7B1	69.1	72.4	43.1	53.7	36.4	42.7	47.2	58.1	52.8
+ Parallel Data	70.3	72.3	45.5	53.9	38.0	43.9	48.3	59.5	54.0
Δ	01.2	-00.1	02.4	00.2	01.6	01.2	01.0	01.4	01.2
BLOOM-3B	64.0	68.9	39.7	50.9	29.4	26.2	32.7	54.5	45.8
+ Parallel Data	65.0	69.1	41.4	51.6	30.9	26.7	34.5	56.9	47.0
Δ	01.0	00.2	01.8	00.7	01.5	00.5	01.8	02.4	01.2
BLOOM-1B7	59.0	65.8	37.2	48.5	20.0	22.2	24.8	53.0	41.3
+ Parallel Data	61.1	65.7	38.9	48.0	20.8	20.9	24.4	53.0	41.6
Δ	02.0	-00.1	01.6	-00.6	00.8	-01.3	-00.3	00.0	00.3

Table 7: Effect of parallel corpora on BLOOM models of different sizes across various tasks. '+ Parallel Data' indicates instruction tuning of the given mLLM with the MT objective, using 10K parallel corpora obtained with the best filtering strategy (ID 5) as shown in Table 3.

to the smaller models' less developed in-context learning capabilities before instruction tuning, allowing for more substantial improvements when supplemented with parallel corpora.

Larger models excel in diverse tasks. Conversely, larger models generally demonstrate greater enhancements in tasks beyond machine translation. Both BLOOM-7B1 and BLOOM-3B exhibit a 1.2% improvement compared to their original mLLMs, while BLOOM-1B7 shows a slight 0.3% improvement. Specifically, BLOOM-7B1 and BLOOM-3B display notable improvements in tasks except for FLORES, while BLOOM-1B7 achieves comparable or even worse results.

These findings demonstrate that when leveraging parallel corpora to enhance mLLMs, larger models not only exhibit improvements in direct tasks, such as machine translation, but also demonstrate a more substantial overall enhancement across a variety of tasks. In contrast, smaller models primarily show benefits in direct tasks. This difference can be attributed to the superior cross-task transferability of larger mLLMs, where insights gained from parallel corpora in one task contribute to improved performance in others.

8 Conclusion

This paper investigates the impact of four critical factors – data quality, data quantity, objectives, and mLLM sizes – on leveraging parallel corpora to enhance mLLMs across diverse languages and tasks. Our findings underscore the crucial importance of filtering out noisy translations to procure high-quality training data for improving mLLMs. Surprisingly, even a relatively modest dataset of 10K samples can yield promising results. Furthermore, our analysis shows that employing the machine translation objective leads to optimal outcomes. Importantly, larger models exhibit a greater capacity to benefit from parallel corpora, achieving more substantial improvements. This study provides a comprehensive recipe for effectively lever-

aging parallel corpora to enhance mLLMs. These insights significantly contribute to advancing the understanding and optimization of mLLMs across different languages and tasks.

Limitations

Due to limited computational resources, we opted not to explore full-parameter training for leveraging parallel corpora. Instead, we focused on LoRA, drawing on insights from previous studies. Additionally, our investigation is restricted to the BLOOM series, and we did not extend our analysis to other mLLMs. Furthermore, we did not also explore mLLMs larger than 7B1.

Acknowledgements

This work was funded by DFG (SCHU 2246/14-1), the European Research Council (DECOLLAGE, ERC-2022-CoG #101088763), EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (Center for Responsible AI), by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research, and by FCT/MECI through national funds and when applicable co-funded EU funds under UID/50008: Instituto de Telecomunicações.

References

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2023. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. *CoRR*, abs/2309.07445.

Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: multilingual evaluation of generative AI. *CoRR*, abs/2303.12528.

Duarte M. Alves, Nuno Miguel Guerreiro, João Alves, José Pombal, Ricardo Rei, José Guilherme Camargo de Souza, Pierre Colombo, and André F. T. Martins. 2023. Steering large language models for machine translation with finetuning and in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 11127–11148. Association for Computational Linguistics.

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro Henrique Martins, João Alves, M. Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *CoRR*, abs/2402.17733.

Mikel Artetxe, Itziar Aldabe, Rodrigo Agerri, Olatz Perez-de-Viñaspre, and Aitor Soroa. 2022. Does corpus quality really matter for low-resource languages? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11*, pages 7383–7390. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4623–4637. Association for Computational Linguistics.

Samuel Cahyawijaya, Holy Lovenia, Tiezheng Yu, Willy Chung, and Pascale Fung. 2023. Instruct-align: Teaching novel languages with to llms through alignment-based cross-lingual instruction. *CoRR*, abs/2305.13627.

Hao Chen, Yiming Zhang, Qi Zhang, Hantao Yang, Xiaomeng Hu, Xuetao Ma, Yifan Yanggong, and Junbo Zhao. 2023. Maybe only 0.5% data is needed: A preliminary exploration of low training data instruction tuning. *CoRR*, abs/2305.09246.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3576–3588. Association for Computational Linguistics.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021b. Improving pretrained cross-lingual language models via self-labeled word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3418–3430. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised

cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

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, Loïc 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. *CoRR*, abs/2207.04672.

Himanshu Gupta, Saurabh Arjun Sawant, Swaroop Mishra, Mutsumi Nakamura, Arindam Mitra, Santosh Mashetty, and Chitta Baral. 2023. Instruction tuned models are quick learners. *CoRR*, abs/2306.05539.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2021. Explicit alignment objectives for multilingual bidirectional encoders. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3633–3643. Association for Computational Linguistics.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2485–2494. Association for Computational Linguistics.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André F. T. Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1082–1117. Association for Computational Linguistics.

Amir Hossein 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, Singapore, December 6-10, 2023*, pages 6155–6218. Association for Computational Linguistics.

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 Javier Ortiz Suárez, 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 Balli, 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. *Trans. Assoc. Comput. Linguistics*, 10:50–72.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7315–7330. Association for Computational Linguistics.

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2023. Align after pre-train: Improving multilingual generative models with cross-lingual alignment. *CoRR*, abs/2311.08089.

Peiqin Lin, Chengzhi Hu, Zheyu Zhang, André F. T. Martins, and Hinrich Schütze. 2024a. mplm-sim: Better cross-lingual similarity and transfer in multilingual pretrained language models. In *Findings of the*

Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024, pages 276–310. Association for Computational Linguistics.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. 2024b. Mala-500: Massive language adaptation of large language models. *CoRR*, abs/2401.13303.

Peiqin Lin, André F. T. Martins, and Hinrich Schütze. 2024c. XAMPLER: learning to retrieve cross-lingual in-context examples. *CoRR*, abs/2405.05116.

Yihong Liu, Peiqin Lin, Mingyang Wang, and Hinrich Schütze. 2023. OFA: A framework of initializing unseen subword embeddings for efficient large-scale multilingual continued pretraining. *CoRR*, abs/2311.08849.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. Are emergent abilities in large language models just in-context learning? *CoRR*, abs/2309.01809.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-M: enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *CoRR*, abs/2012.15674.

Laura Cabello Piqueras and Anders Søgaard. 2022. Are pretrained multilingual models equally fair across languages? In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 3597–3605. International Committee on Computational Linguistics.

Surangika Ranathunga, Nisansa de Silva, Menan Velayuthan, Aloka Fernando, and Charitha Rathnayake. 2024. Quality does matter: A detailed look at the quality and utility of web-mined parallel corpora. *CoRR*, abs/2402.07446.

Ricardo Rei, Marcos V. Treviso, Nuno Miguel Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte M. Alves, Luísa Coheur, Alon Lavie, and André F. T. Martins. 2022. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation, WMT 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 7-8, 2022*, pages 634–645. Association for Computational Linguistics.

Machel Reid and Mikel Artetxe. 2022a. On the role of parallel data in cross-lingual transfer learning. *CoRR*, abs/2212.10173.

Machel Reid and Mikel Artetxe. 2022b. PARADISE: exploiting parallel data for multilingual sequence-to-sequence pretraining. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 800–810. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Ahmet Üstün, Viraat Aryabumi, Zheng Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *CoRR*, abs/2402.07827.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *CoRR*, abs/2306.04751.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jerry W. Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *CoRR*, abs/2303.03846.

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *CoRR*, abs/2309.11674.

Jian Yang, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training.

In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9386–9393. AAAI Press.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1628–1639. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: less is more for alignment. *CoRR*, abs/2305.11206.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating large language models to non-english by aligning languages. *CoRR*, abs/2308.04948.

A Additional Experiments on XGLM

We explore the effect of parallel corpora quality on XGLM-7.5B, and the results are shown in Table 8. As shown, experiments on XGLM-7.5B exhibits consistent findings as those on BLOOM-7B1.

ID	MODEL			Accuracy
0	XGLM-7.5B			52.8
	τ_c	τ_l	LID	
1	0	0	✓	54.6
2	0.5	0	✓	55.0
3	0.75	5	✓	54.4
4	0.75	0	✗	55.0
5	0.75	0	✓	**55.5**

Table 8: Performance (%) of XGLM-7.5B on SIB after instruction tuning with the machine translation objective using 10K parallel corpora with various quality filtering strategies. Parameters include τ_c for COMETWIKI score threshold, τ_l for sentence length threshold, and LID indicating the adoption of language identification filtering.

B Effect of LoRA Rank

We conduct initial experiments on Arabic to explore the effect of setting the rank of LoRA, and the results are shown in Table 9. As shown, setting the rank of LoRA as 16 lead to best performance. Therefore, we use LoRA rank as 16 across all the experiments.

C Prompt

The prompts of FLORES, MUSE, MLQA, XQuAD, and SIB are shown as follows:

FLORES/MUSE

```
Translate the following
text from [SOURCE_LANG]
to [TARGET_LANG].\nText:
[SOURCE_TEXT]\nTranslation:
[TARGET_TEXT]
```

Rank	Accuracy
8	59.0
16	**62.0**
32	60.0
128	61.0
256	60.5
512	59.5

Table 9: Effect of LoRA rank: Accuracy on SIB using English-Arabic parallel data to improve BLOOM-7B1.

MLQA/XQuAD

```
[Passage] \nQ:
[Question]\nA: [Answer]
```

SIB

```
The topic of the news
[Passage] is [Label]
```

Chapter 6

mPLM-Sim: Better Cross-Lingual Similarity and Transfer in Multilingual Pretrained Language Models

mPLM-Sim: Better Cross-Lingual Similarity and Transfer in Multilingual Pretrained Language Models

Peiqin Lin[*1,2], **Chengzhi Hu**[*3], **Zheyu Zhang**[1], **André F. T. Martins**[4,5,6], **Hinrich Schütze**[1,2]
[1]Center for Information and Language Processing, LMU Munich
[2]Munich Center for Machine Learning [3]Institute of Informatics, LMU Munich
[4]Instituto Superior Técnico, Universidade de Lisboa (Lisbon ELLIS Unit)
[5]Instituto de Telecomunicações [6]Unbabel
linpq@cis.lmu.de, {Chengzhi.Hu, Zheyu.Zhang}@campus.lmu.de

Abstract

Recent multilingual pretrained language models (mPLMs) have been shown to encode strong language-specific signals, which are not explicitly provided during pretraining. It remains an open question whether it is feasible to employ mPLMs to measure language similarity, and subsequently use the similarity results to select source languages for boosting cross-lingual transfer. To investigate this, we propose mPLM-Sim, a language similarity measure that induces the similarities across languages from mPLMs using multi-parallel corpora. Our study shows that mPLM-Sim exhibits moderately high correlations with linguistic similarity measures, such as lexicostatistics, genealogical language family, and geographical sprachbund. We also conduct a case study on languages with low correlation and observe that mPLM-Sim yields more accurate similarity results. Additionally, we find that similarity results vary across different mPLMs and different layers within an mPLM. We further investigate whether mPLM-Sim is effective for zero-shot cross-lingual transfer by conducting experiments on both low-level syntactic tasks and high-level semantic tasks. The experimental results demonstrate that mPLM-Sim is capable of selecting better source languages than linguistic measures, resulting in a 1%-2% improvement in zero-shot cross-lingual transfer performance.[1]

1 Introduction

Recent multilingual pretrained language models (mPLMs) trained with massive data, e.g., mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020) and BLOOM (Scao et al., 2022), have become a standard for multilingual representation learning. Follow-up works (Wu and Dredze, 2019; Libovický et al., 2020; Liang et al., 2021; Chang et al., 2022) show that these mPLMs encode strong language-specific signals which are not explicitly provided during pretraining. However, the possibility of using mPLMs to measure language similarity and utilizing the similarity results to pick source languages for enhancing cross-lingual transfer is not yet thoroughly investigated.

To investigate language similarity in mPLMs, we propose mPLM-Sim, a measure that leverages mPLMs and multi-parallel corpora to measure similarity between languages. Using mPLM-Sim, we intend to answer the following research questions.

(Q1) *What is the correlation between mPLM-Sim and linguistic similarity?*

We compute Pearson correlation between similarity results of mPLM-Sim and linguistic similarity measures. The results show that mPLM-Sim has a moderately high correlation with some linguistic measures, such as lexical-based and language-family-based measures. Additional case studies on languages with low correlation demonstrate that mPLMs can acquire the similarity patterns among languages through pretraining on massive data.

(Q2) *Do different layers of an mPLM produce different similarity results?*

Jawahar et al. (2019); Sabet et al. (2020); Choenni and Shutova (2022) have demonstrated that different linguistic information is encoded across different layers of an mPLM. We analyze the performance of mPLM-Sim across layers and show that mPLM-Sim results vary across layers, aligning with previous findings. Specifically, the embedding layer captures lexical information, whereas the middle layers reveal more intricate similarity patterns encompassing general, geographical, and syntactic aspects. However, in the high layers, the ability to distinguish between languages becomes less prominent. Furthermore, we observe that clustering of languages also varies by layer, shedding new light on how the representation of language-specific information changes throughout layers.

[*]Equal contribution.
[1]Our code is open-sourced at https://github.com/cisnlp/mPLM-Sim.

Findings of the Association for Computational Linguistics: EACL 2024, pages 276–310

(Q3) *Do different mPLMs produce different similarity results?*

We make a comprehensive comparison among a diverse set of 11 mPLMs in terms of architecture, modality, model size, and tokenizer. The experimental results show that input modality (text or speech), model size, and data used for pretraining have large effects on mPLM-Sim while tokenizers and training objectives have little effect.

(Q4) *Can mPLM-Sim choose better source languages for zero-shot cross-lingual transfer?*

Previous works (Lin et al., 2019; Pires et al., 2019; Lauscher et al., 2020; Nie et al., 2022; Wang et al., 2023; Imai et al., 2023) have shown that the performance of cross-lingual transfer positively correlates with linguistic similarity. However, we find that there can be a mismatch between mPLM subspaces and linguistic clusters, which may lead to a failure of zero-shot cross-lingual transfer for low-resource languages. Intuitively, mPLM-Sim can select the source languages that boost cross-lingual transfer better than linguistic similarity since it captures the subspaces learned during pretraining (and which are the basis for successful transfer). To examine this, we conduct experiments on four datasets that require reasoning about different levels of syntax and semantics for a diverse set of low-resource languages. The results show that mPLM-Sim achieves 1%-2% improvement over linguistic similarity measures for cross-lingual transfer.

2 Setup

2.1 mPLM-Sim

Generally, a transformer-based mPLM consists of N layers: $N-1$ transformer layers plus the static embedding layer. Given a multi-parallel corpus[2], mPLM-Sim aims to provide the similarity results of N layers for an mPLM across L languages considered. In this context, we define languages using the ISO 639-3 code combined with the script, e.g., "eng_Latn" represents English written in Latin.

For each sentence x in the multi-parallel corpus, the mPLM computes its sentence embedding for the ith layer of the mPLM: $\boldsymbol{h}_i = E(x)$. For mPLMs with bidirectional encoders, including encoder architecture, e.g., XLM-R, and encoder-decoder architecture, e.g., mT5, $E(\cdot)$ is a mean pooling operation over hidden states, which performs better than [CLS] and MAX strategies (Reimers and Gurevych, 2019). For mPLMs with auto-regressive encoders, e.g., mGPT, $E(\cdot)$ is a position-weighted mean pooling method, which gives later tokens a higher weight (Muennighoff, 2022). Finally, sentence embeddings for all sentences of the L languages are obtained.

For ith layer, the similarity of each language pair is computed using the sentence embeddings of all multi-parallel sentences. Specifically, we get the cosine similarity of each parallel sentence of the language pair, and then average all similarity scores across sentences as the final score of the pair. Finally, we have a similarity matrix $\boldsymbol{S}_i \in \mathbb{R}^{L \times L}$ across L languages for the ith layer of the mPLM.

2.2 mPLMs, Corpora and Languages

We consider a varied set of 11 mPLMs for our investigation, differing in model size, number of covered languages, architecture, modality, and data used for pretraining. Full list and detailed information of the selected mPLMs are shown in Tab. 1.

We work with three multi-parallel corpora: the text corpora Flores (Costa-jussà et al., 2022) and Parallel Bible Corpus (PBC, (Mayer and Cysouw, 2014)) and the speech corpus Fleurs (Conneau et al., 2022). Flores covers more than 200 languages. Since both PBC and Fleurs are not fully multi-parallel, we reconstruct them to make them multi-parallel. After recostruction, PBC covers 379 languages, while Fleurs covers 67 languages. PBC consists of religious text, and both Flores and Fleurs are from web articles. The speech of Fleurs is aligned to the text of Flores, enabling us to compare text mPLMs with speech mPLMs. We use 500 multi-parallel sentences from each corpus. Languages covered by mPLMs and corpora are listed in §A.

2.3 Evaluation

Pearson Correlation We compute Pearson correlation scores to measure how much mPLM-Sim correlates with seven linguistic similarity measures: LEX, GEN, GEO, SYN, INV, PHO and FEA. LEX is computed based on the edit distance of the two corpora. The six others are provided by lang2vec. GEN is based on language family. GEO is orthodromic distance, i.e., the shortest distance between two points on the surface of the earth. SYN is derived from the syntactic structures of the languages. Both INV and PHO are phonological features. INV

[2]Monolingual corpora covering multiple languages can be also used to measure language similarity. Our initial experiments (§B.1) show that parallel corpora yield better results while using fewer sentences than monolingual corpora. Therefore, we use parallel corpora for our investigation.

Model	Size	|Lang|	|Layer|	Tokenizer	Arch.	Objective	Modality	Data
mBERT (Devlin et al., 2019)	172M	104	13	Subword	Enc	MLM, NSP	Text	Wikipedia
XLM-R-Base (Conneau et al., 2020)	270M	100	13	Subword	Enc	MLM	Text	CC
XLM-R-Large (Conneau et al., 2020)	559M	100	25	Subword	Enc	MLM	Text	CC
Glot500 (Imani et al., 2023)	395M	515	13	Subword	Enc	MLM	Text	Glot500-c
mGPT (Shliazhko et al., 2022)	1.3B	60	25	Subword	Dec	CLM	Text	Wikipedia+mC4
mT5-Base (Xue et al., 2021)	580M	101	13	Subword	Enc-Dec	MLM	Text	mC4
CANINE-S (Clark et al., 2022)	127M	104	17	Char	Enc	MLM, NSP	Text	Wikipedia
CANINE-C (Clark et al., 2022)	127M	104	17	Char	Enc	MLM, NSP	Text	Wikipedia
XLM-Align (Chi et al., 2021b)	270M	94	13	Subword	Enc	MLM, TLM, DWA	Text	Wikipedia+CC
NLLB-200 (Costa-jussà et al., 2022)	1.3B	204	25	Subword	Enc-Dec	MT	Text	NLLB
XLS-R-300M (Babu et al., 2021)	300M	128	25	-	Enc	MSP	Speech	CommonVoice

Table 1: 11 mPLMs considered in the paper. |Layer| denotes the number of layers used for measuring similarity. Both the static embedding layer and all layers of the transformer are considered. For encoder-decoder architectures, we only consider the encoder. |Lang|: the number of languages covered. Arch.: Architecture. Enc: Encoder. Dec: Decoder. MLM: Masked Language Modeling. CLM: Causal Language Modeling. TLM: Translation Language Modeling. NSP: Next Sentence Prediction. DWA: Denoising Word Alignment. MT: Machine Translation. MSP: Masked Speech Prediction. CC: CommonCrawl.

Task	Corpus	|Train|	|Dev|	|Test|	|Lang|	Metric	Domain
Sequence Labeling	NER (Pan et al., 2017)	5,000	500	100-10,000	108	F1	Wikipedia
	POS (de Marneffe et al., 2021)	5,000	500	100-22,358	60	F1	Misc
Text Classification	MASSIVE (FitzGerald et al., 2022)	11,514	2,033	2,974	44	Acc	Misc
	Taxi1500 (Ma et al., 2023)	860	106	111	130	F1	Bible

Table 2: Evaluation dataset statistics. |Train|/|Dev|: train/dev set size (source language). |Test|: test set size (target language). |Lang|: number of target languages.

is derived from PHOIBLE, while PHO is based on WALS and Ethnologue. FEA is computed by combining GEN, GEO, SYN, INV and PHO.

For each target language, we have the similarity scores between the target language and the other $L-1$ languages based on the similarity matrix $\boldsymbol{S}_i$ for layer i (see §2.1), and also the similarity scores based on the considered linguistic similarity measure j. Then we compute the Pearson correlation r_i^j between these two similarity score lists. We choose the highest correlation score across all layers as the result of each target language since the results for different languages vary across layers. Finally, we report MEAN (M) and MEDIAN (Mdn) of the correlation scores for all languages. Here, we consider 32 languages covered by all models and corpora.

Case Study In addition to the quantitative evaluation, we conduct manual analysis for languages that exhibit low correlation scores. We apply complete linkage hierarchical clustering to get the similar languages of the analyzed language for analysis. Specifically, the languages which have the most common shared path in the hierarchical tree with the target language are considered as similar languages. To analyze as many languages as possible, we consider the setting of Glot500 and PBC.

Cross-Lingual Transfer To compare mPLM-Sim with linguistic measures for zero-shot cross-lingual transfer, we run experiments for low-resource languages on four datasets, including two for sequence labeling, and two for text classification. Details of the four tasks are shown in Tab. 2.

We selected six high-resource and typologically diverse languages, namely Arabic (arb_Arab), Chinese (cmn_Hani), English (eng_Latn), Hindi (hin_Deva), Russian (rus_Cyrl), and Spanish (spa_Latn), as source languages. For a fair comparison, we use the same amount of source language data for fine-tuning and validation as shown in Tab. 2.

The evaluation targets all languages that are covered by both Glot500 and Flores and have at least 100 samples, excluding the six source languages. The language list for evaluation is provided in §A.

We obtain the most similar source language for each target language by applying each of the seven linguistic similarity measures (LEX, GEN, GEO, SYN, INV, PHO, FEA) and our mPLM-Sim. Here, we consider the setting of Glot500 and Flores for mPLM-Sim since extensive experiments (see §B.2) show that Flores provides slightly better similarity results than PBC. For the linguistic similarity mea-

	XLM-R-Base		XLM-R-Large		mT5-Base		mGPT		mBERT		Glot500	
	M	Mdn	M	Mdn	M	Mdn	M	Mdn	M	Mdn	M	Mdn
LEX	0.740	0.859	0.684	0.862	0.628	0.796	0.646	0.848	0.684	0.882	0.741	0.864
GEN	0.489	0.563	0.570	0.609	0.577	0.635	0.415	0.446	0.513	0.593	0.527	0.600
GEO	0.560	0.656	0.587	0.684	0.528	0.586	0.348	0.362	0.458	0.535	0.608	0.674
SYN	0.637	0.662	0.709	0.738	0.594	0.612	0.548	0.591	0.611	0.632	0.577	0.607
INV	0.272	0.315	0.312	0.292	0.295	0.321	0.340	0.394	0.216	0.246	0.248	0.293
PHO	0.112	0.151	0.207	0.258	0.166	0.176	0.184	0.239	0.111	0.125	0.094	0.144
FEA	0.378	0.408	0.443	0.466	0.354	0.371	0.455	0.479	0.346	0.361	0.358	0.372
AVG	0.455	0.516	0.502	0.559	0.449	0.500	0.420	0.480	0.420	0.482	0.451	0.508
	CANINE-S		CANINE-C		NLLB-200		XLM-Align		XLS-R-300M		AVG	
	M	Mdn	M	Mdn	M	Mdn	M	Mdn	M	Mdn	M	Mdn
LEX	0.661	0.821	0.639	0.784	0.722	0.856	0.728	0.869	0.285	0.262	0.651	0.791
GEN	0.548	0.629	0.565	0.633	0.538	0.626	0.516	0.606	0.401	0.353	0.514	0.572
GEO	0.504	0.560	0.533	0.624	0.490	0.499	0.616	0.690	0.531	0.541	0.524	0.583
SYN	0.476	0.521	0.507	0.559	0.375	0.370	0.634	0.669	0.354	0.389	0.548	0.577
INV	0.329	0.390	0.369	0.406	0.337	0.373	0.252	0.315	0.191	0.180	0.287	0.321
PHO	0.112	0.137	0.117	0.173	0.101	0.108	0.105	0.143	0.124	0.115	0.130	0.161
FEA	0.317	0.297	0.367	0.360	0.311	0.326	0.368	0.399	0.203	0.175	0.355	0.365
AVG	0.421	0.479	0.442	0.506	0.411	0.451	0.460	0.527	0.298	0.288	0.430	0.481

Table 3: Comparison across mPLMs: Pearson correlation between mPLM-Sim and seven similarity measures for all mPLMs and Flores/Fleurs on 32 languages. mPLM-Sim strongly correlates with LEX, moderate strongly correlates with GEN, GEO, and SYN, and weakly correlates with INV, PHO, and FEA.

sures, if the most similar source language is not available due to missing values in lang2vec, we use eng_Latn as the source language. We also compare mPLM-Sim with the ENG baseline defined as using eng_Latn as the source language for all target languages.

We use the same hyper-parameter settings as in (Hu et al., 2020; FitzGerald et al., 2022; Ma et al., 2023). Specifically, we set the batch size to 32 and the learning rate to 2e-5 for both NER and POS, and fine-tune Glot500 for 10 epochs. For MASSIVE, we use a batch size of 16, a learning rate of 4.7e-6, and train for 100 epochs. For Taxi1500, we use a batch size of 32, a learning rate of 2e-5, and train for 30 epochs. In all tasks, we select the model for evaluating target languages based on the performance of the source language validation set.

3 Results

3.1 Comparison Between mPLM-Sim and Linguistic Similarity

Tab. 3 shows the Pearson correlation between mPLM-Sim and linguistic similarity measures of 11 mPLMs, and also the average correlations of all 11 mPLMs. We observe that mPLM-Sim strongly correlates with LEX, which is expected since mPLMs learn language relationships from data and LEX similarity is the easiest pattern to learn. Besides, mPLM-Sim has moderately strong correlations with GEN, GEO, and SYN, which shows that mPLMs can learn high-level patterns for language similarity. mPLM-Sim also has a weak correlation with INV, and a very weak correlation with PHO, indicating mPLMs do not capture phonological similarity well. Finally, mPLM-Sim correlates with FEA weakly since FEA is the measure combining both high- and low-correlated linguistics features.

To further compare mPLM-Sim with linguistic similarity measures, we conduct a manual analysis on languages for which mPLM-Sim has weak correlations with LEX, GEN, and GEO. As mentioned in §2, with the setting of Glot500 and PBC, we apply hierarchical clustering and use similar results for analysis.

We find that mPLM-Sim can deal well with languages that are not covered by lang2vec. For example, Norwegian Nynorsk (nno_Latn) is not covered by lang2vec, and mPLM-Sim can correctly find its similar languages, i.e., Norwegian Bokmål

(nob_Latn) and Norwegian (nor_Latn). Furthermore, mPLM-Sim can well capture the similarity between languages which cannot be well measured by either LEX, GEN, or GEO.

For LEX, mPLM-Sim can capture similar languages written in different scripts. A special case is the same languages in different scripts. Specifically, mPLM-Sim matches Uighur in Latin and Arabic (uig_Arab and uig_Latn), also Karakalpak in Latin and Cyrillic (kaa_Latn and kaa_Cyrl). In general, mPLM-Sim does a good job at clustering languages from the same language family but written in different scripts, e.g., Turkic (Latn, Cyrl, Arab) and Slavic (Latn, Cyrl).

For GEN, mPLM-Sim captures correct similar languages for isolates and constructed languages. Papantla Totonac (top_Latn) is a language of the Totonacan language family and spoken in Mexico. It shares areal features with the Nahuan languages (nch_Latn, ncj_Latn, and ngu_Latn) of the Uto-Aztecan family, which are all located in the Mesoamerican language area.[3] Esperanto (epo_Latn) is a constructed language whose vocabulary derives primarily from Romance languages, and mPLM-Sim correctly identifies Romance languages such as French (fra_Latn) and Italian (ita_Latn) as similar. The above two cases show the superiority of mPLM-Sim compared to GEN.

The GEO measure may not be suitable for certain language families, such as Austronesian languages and mixed languages. Austronesian languages have the largest geographical span among language families prior to the spread of Indo-European during the colonial period.[4] Moreover, for mixed languages, such as creole languages, their similar languages are often geographically distant due to colonial history. In contrast to GEO, mPLM-Sim can better cluster these languages.

The above analysis shows that it is non-trivial to use either LEX, GEN, or GEO for measuring language similarity. In contrast, mPLM-Sim directly captures similarity from mPLMs and can therefore produce better similarity results.

However, we observe that obtaining accurate similarity results from mPLMs using mPLM-Sim can be challenging for certain languages. To gain further insights into this issue, we examine the correlation between performances, specifically the correlation between mPLM-Sim and GEN, and the sizes of the pretraining data. Surprisingly, we find a remarkably weak correlation (-0.008), suggesting that differences in pretraining data sizes do not significantly contribute to variations in performances.

Instead, our findings indicate a different key factor: the coverage of multiple languages within the same language family. This observation is substantiated by a strong correlation of 0.617 between the diversity of languages within a language family (measured by the number of languages included) and the performance of languages belonging to that particular language family.

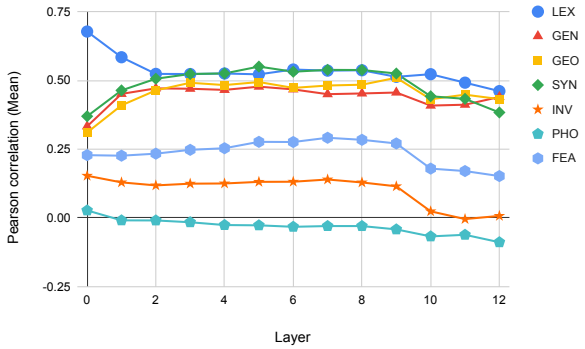


Figure 1: Comparison across layers: Pearson correlation (MEAN) between mPLM-Sim and linguistic similarity measures across layers for Glot500 and Flores on 32 languages. Correlation between mPLM-Sim and LEX peaks in the first layer and decreases, while the correlation with GEN, GEO, and SYN slightly increases in the low layers before reaching its peak.

3.2 Comparison Across Layers for mPLM-Sim

We analyze the correlation between mPLM-Sim and linguistic similarity measures across different layers of an mPLM, specifically for Glot500. The results, presented in Fig. 1, demonstrate the variation in mPLM-Sim results across layers. Notably, in the first layer, mPLM-Sim exhibits a high correlation with LEX, which gradually decreases as we move to higher layers. Conversely, the correlation between mPLM-Sim and GEN, GEO, and SYN shows a slight increase in the lower layers, reaching its peak in layer 1 or 2 of the mPLM. However, for the higher layers (layers 10-12), all correlations slightly decrease. We also performed further visualization and analysis across layers using the setting of Glot500 and Flores for mPLM-Sim (§C). The findings are consistent with our observations from Fig. 1.

[3] https://en.wikipedia.org/wiki/Mesoamerican_language_area

[4] https://en.wikipedia.org/wiki/Austronesian_languages

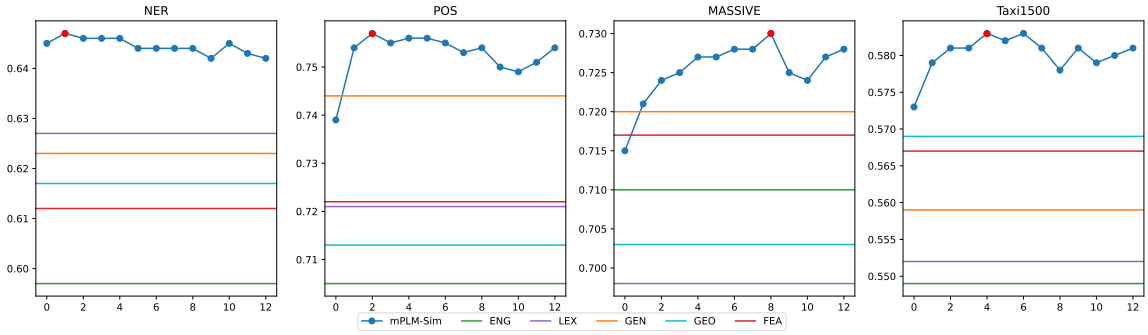


Figure 2: Macro average results (averaged over target languages) on cross-lingual transfer for baselines and for mPLM-Sim in all layers of Glot500. ENG represents using English as the source language. LEX, GEN, GEO, and FEA indicate using the most similar languages based on the corresponding similarity measures as the source language. The red dots of mPLM-Sim highlight the layer with the highest score.

Furthermore, our case study shows that the layers which have highest correlations between mPLM-Sim and LEX, GEN, or GEO vary across languages. For example, Atlantic–Congo languages achieve highest correlation with GEN at the 1st layer, while Mayan languages at the 6th layer. This finding demonstrates that language-specific information changes across layers.

3.3 Comparison Across Models for mPLM-Sim

Tab. 3 presents a broad comparison among 11 different mPLMs, revealing several key findings.

Firstly, the decoder architecture has a negative impact on performance due to the inherent difficulty in obtaining accurate sentence-level representations from the decoder. For example, the decoder-only mPLM mGPT performs worse than encoder-only mPLMs such as XLM-R and mBERT. This observation is reinforced by the comparison between XLM-R-Large and mT5-Base, which have nearly identical model sizes. Remarkably, XLM-R-Large outperforms mT5-Base on AVG by 5% for both Mean (M) and Median (Mdn) scores.

Additionally, tokenizer-free mPLMs achieve comparable performance to subword-tokenizer-based mPLMs. Notably, mPLMs such as mBERT, CANINE-S, and CANINE-C, which share pretraining settings, exhibit similar performances.

The size of mPLMs also influences mPLM-Sim in terms of LEX, GEN, and SYN. Comparing XLM-R-Base with XLM-R-Large, higher-level language similarity patterns are more evident in larger mPLMs. Specifically, XLM-R-Large shows a higher correlation with high-level patterns such as GEN and SYN, while having a lower correlation with low-level patterns like LEX, compared to XLM-R-Base.

The training objectives adopted in mPLMs also impact the performance of mPLM-Sim. Task-specific mPLMs, such as NLLB-200, perform slightly worse than general-purpose mPLMs. Besides, XLM-Align, which leverages parallel objectives to align representations across languages, achieves comparable results to XLM-R-Base. This highlights the importance of advancing methods to effectively leverage parallel corpora.

The choice of pretraining data is another important factor. For example, mBERT uses Wikipedia, while XLM-R-Base uses CommonCrawl, which contains more code-switching. As a result, XLM-R-Base has a higher correlation with GEO and achieves higher AVG compared to mBERT.

The speech mPLM, i.e., XLS-R-300M, exhibits lower correlation than text mPLMs, consistent with findings from Abdullah et al. (2023). XLS-R-300M learns language similarity from speech data, which is biased towards the accents of speakers. Consequently, XLS-R-300M has a higher correlation with GEO, which is more related to accents, than other similarity measures.

Factors such as the number of languages have minimal effects on mPLM-Sim. Glot500, covering over 500 languages, achieves comparable results with XLM-R-Base.

3.4 Effect for Cross-Lingual Transfer

The macro average results of cross-lingual transfer across target languages for both mPLM-Sim and baselines are presented in Fig. 2. Among the evaluated tasks, ENG exhibits the worst performance in three out of four tasks, emphasizing the importance

		Language	GEN		mPLM-Sim		Δ		Language	GEN		mPLM-Sim		Δ
high end	NER	jpn_Jpan	0.177	eng_Latn	0.451	cmn_Hani	0.275	POS	jpn_Jpan	0.165	eng_Latn	0.534	cmn_Hani	0.369
		kir_Cyrl	0.391	eng_Latn	0.564	rus_Cyrl	0.173		mlt_Latn	0.603	arb_Arab	0.798	spa_Latn	0.196
		mya_Mymr	0.455	cmn_Hani	0.607	hin_Deva	0.153		wol_Latn	0.606	eng_Latn	0.679	spa_Latn	0.074
low end		pes_Arab	0.653	hin_Deva	0.606	arb_Arab	-0.047		ekk_Latn	0.815	eng_Latn	0.790	rus_Cyrl	-0.025
		tgl_Latn	0.745	eng_Latn	0.667	spa_Latn	-0.078		bam_Latn	0.451	eng_Latn	0.411	spa_Latn	-0.039
		sun_Latn	0.577	eng_Latn	0.490	spa_Latn	-0.087		gla_Latn	0.588	rus_Cyrl	0.548	spa_Latn	-0.040
high end	MASSIVE	mya_Mymr	0.616	cmn_Hani	0.707	hin_Deva	0.091	Taxi1500	tgk_Cyrl	0.493	hin_Deva	0.724	rus_Cyrl	0.231
		amh_Ethi	0.532	arb_Arab	0.611	hin_Deva	0.079		kin_Latn	0.431	eng_Latn	0.619	spa_Latn	0.188
		jpn_Jpan	0.384	eng_Latn	0.448	cmn_Hani	0.064		kik_Latn	0.384	eng_Latn	0.555	spa_Latn	0.172
low end		cym_Latn	0.495	rus_Cyrl	0.480	spa_Latn	-0.015		ckb_Arab	0.622	hin_Deva	0.539	arb_Arab	-0.083
		tgl_Latn	0.752	eng_Latn	0.723	spa_Latn	-0.028		nld_Latn	0.713	eng_Latn	0.628	spa_Latn	-0.085
		deu_Latn	0.759	eng_Latn	0.726	spa_Latn	-0.033		kac_Latn	0.580	cmn_Hani	0.483	hin_Deva	-0.097

Table 4: Results for three languages each with the largest (high end) and smallest (low end) gains from mPLM-Sim vs. GEN for four tasks. mPLM-Sim's gain over GEN is large at the high end and smaller negative at the low end. We report both the selected source languages and the results on the evaluated target languages. For mPLM-Sim, the results are derived from the layers exhibiting the best performances as shown in Fig. 2. See §E for detailed results for each task and each target language.

of considering language similarity when selecting source languages for cross-lingual transfer. mPLM-Sim surpasses all linguistic similarity measures in every task, including both syntactic and semantic tasks, across all layers except layer 0. This indicates that mPLM-Sim is more effective in selecting source languages that enhance the performance of target languages compared to linguistic similarity measures.

For low-level syntactic tasks, the lower layers (layer 1 or 2) exhibit superior performance compared to all other layers. Conversely, for high-level semantic tasks, it is the middle layer of the mPLM that consistently achieves the highest results across all layers. This can be attributed to its ability to capture intricate similarity patterns.

In Tab. 4, we further explore the benefits of mPLM-Sim in cross-lingual transfer. We present a comprehensive analysis of the top 3 performance improvements and declines across languages. We compare mPLM-Sim and GEN across four cross-lingual transfer tasks. By examining these results, we gain deeper insights into the advantages of mPLM-Sim in facilitating effective cross-lingual transfer.

The results clearly demonstrate that mPLM-Sim has a substantial performance advantage over GEN for certain target languages. On one hand, for languages without any source language in the same language family, such as Japanese (jpn_Jpan), mPLM-Sim successfully identifies its similar language, Chinese (cmn_Hani), whereas GEN fails to do so. Notably, in the case of Japanese, mPLM-Sim outperforms GEN by 27.5% for NER, 36.9% for POS, and 6.4% for MASSIVE.

On the other hand, for languages having source languages within the same language family, mPLM-Sim accurately detects the appropriate source language, leading to improved cross-lingual transfer performance. In the case of Burmese (mya_Mymr), mPLM-Sim accurately identifies Hindi (hin_Deva) as the source language, while GEN mistakenly selects Chinese (cmn_Hani). This distinction results in a significant performance improvement of 15.3% for NER and 9.1% for MASSIVE.

However, we also observe that mPLM-Sim falls short for certain languages when compared to GEN, although the losses are smaller in magnitude compared to the improvements. This finding suggests that achieving better performance in cross-lingual transfer is not solely dependent on language similarity. As mentioned in previous studies such as Lauscher et al. (2020) and Nie et al. (2022), the size of the pretraining data for the source languages also plays a crucial role in cross-lingual transfer.

4 Related Work

4.1 Language Typology and Clustering

Similarity between languages can be due to common ancestry in the genealogical language tree, but also influenced by linguistic influence and borrowing (Aikhenvald and Dixon, 2001; Haspelmath, 2004). Linguists have conducted extensive relevant research by constructing high-quality typological, geographical, and phylogenetic databases, including WALS (Dryer and Haspelmath, 2013), Glottolog (Hammarström et al., 2017), Ethnologue (Saggion et al., 2023), and PHOIBLE (Moran et al.,

2014; Moran and McCloy, 2019). The lang2vec tool (Littell et al., 2017) further integrates these datasets into multiple linguistic distances. Despite its integration of multiple linguistic measures, lang2vec weights each measure equally, and the quantification of these measures for language similarity computation remains a challenge.

In addition to linguistic measures, some non-lingustic measures are also proposed to measure similarity between languages. Specifically, Holman et al. (2011) use Levenshtein (edit) distance to compute the lexical similarity between languages. Lin et al. (2019) propose dataset-dependent features, which are statistical features specific to the corpus used, e.g., lexical overlap. Ye et al. (2023) measure language similarity with basic concepts across languages. However, these methods fail to capture deeper similarities beyond surface-level features.

Language representation is another important category of language similarity measures. Before the era of multilingual pretrained language models (mPLMs), exploiting distributed language representations for measuring language similarity have been studied (Östling and Tiedemann, 2017; Bjerva and Augenstein, 2018). Recent mPLMs trained with massive data have become a new standard for multilingual representation learning. Tan et al. (2019) represent each language by an embedding vector and cluster them in the embedding space. Fan et al. (2021b) find the representation sprachbund of mPLMs, and then train separate mPLMs for each sprachbund. However, these studies do not delve into the research questions mentioned in §1, and it motivates us to carry out a comprehensive investigation of language similarity using mPLMs.

4.2 Multilingual Pretrained Language Models

The advent of mPLMs, e.g., mBERT (Devlin et al., 2019), XLM (Conneau and Lample, 2019), and XLM-R (Conneau et al., 2020), have brought significant performance gains on numerous multilingual natural language understanding benchmarks (Hu et al., 2020).

Given their success, a variety of following mPLMs are proposed. Specifically, different architectures, including decoder-only, e.g., mGPT (Shliazhko et al., 2022) and BLOOM (Scao et al., 2022), and encoder-decoder, e.g., mT5 (Xue et al., 2021), are designed. Tokenizer-free models, including CANINE (Clark et al., 2022), ByT5 (Xue et al., 2022), and Charformer (Tay et al., 2022), are also proposed. Clark et al. (2022) introduce CANINE-S and CANINE-C. CANINE-S adopts a subword-based loss, while CANINE-C uses a character-based one. Glot500 (Imani et al., 2023) extends XLM-R to cover more than 500 languages using vocabulary extension and continued pretraining. Both InfoXLM (Chi et al., 2021a) and XLM-Align (Chi et al., 2021b) exploit parallel objectives to further improve mPLMs. Some mPLMs are specifically proposed for Machine Translation, e.g., M2M-100 (Fan et al., 2021a) and NLLB-200 (Costa-jussà et al., 2022). XLS-R-300M (Babu et al., 2021) is a speech (as opposed to text) model.

Follow-up works show that strong language-specific signals are encoded in mPLMs by means of probing tasks (Wu and Dredze, 2019; Rama et al., 2020; Pires et al., 2019; Müller et al., 2021; Liang et al., 2021; Choenni and Shutova, 2022) and investigating the geometry of mPLMs (Libovický et al., 2020; Chang et al., 2022; Wang et al., 2023). Concurrent with our work, Philippy et al. (2023) have verified that the language representations encoded in mBERT correlate with both linguistic typology and cross-lingual transfer on XNLI for 15 languages. However, these methods lack in-depth analysis and investigate on a limited set of mPLMs and downstream tasks. This inspires us to conduct quantitative and qualitative analysis on linguistic typology and cross-lingual transfer with a broad and diverse set of mPLMs and downstream tasks.

5 Conclusion

In this paper, we introduce mPLM-Sim, a novel approach for measuring language similarities. Extensive experiments substantiate the superior performance of mPLM-Sim compared to linguistic similarity measures. Our study reveals variations in similarity results across different mPLMs and layers within an mPLM. Furthermore, our findings reveal that mPLM-Sim effectively identifies the source language to enhance cross-lingual transfer.

The results obtained from mPLM-Sim have significant implications for multilinguality. On the one hand, it can be further used in linguistic study and downstream applications, such as cross-lingual transfer, as elaborated in the paper. On the other hand, these findings provide valuable insights for improving mPLMs, offering opportunities for their further development and enhancement.

Limitations

(1) The performance of mPLM-Sim may be strongly influenced by the quality and quantity of data used for training mPLMs, as well as the degree to which the target language can be accurately represented. (2) The success of mPLM-Sim depends on the supporting languages of mPLMs. We conduct further experiment and analysis at §D. (3) As for §3.3, we are unable to conduct a strictly fair comparison due to the varying settings in which mPLMs are pretrained, including the use of different corpora and model sizes.

Acknowledgements

This work was funded by the European Research Council (NonSequeToR, grant #740516, and DECOLLAGE, ERC-2022-CoG #101088763), EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020, and by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (Center for Responsible AI). Peiqin Lin acknowledges travel support from ELISE (GA no 951847).

References

Badr M Abdullah, Mohammed Maqsood Shaik, and Dietrich Klakow. 2023. On the nature of discrete speech representations in multilingual self-supervised models. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 159–161.

Alexandra Y. Aikhenvald and R. M. W. Dixon. 2001. *Areal diffusion and genetic inheritance*. Oxford University Press, Oxford.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. XLS-R: self-supervised cross-lingual speech representation learning at scale. *CoRR*, abs/2111.09296.

Johannes Bjerva and Isabelle Augenstein. 2018. From phonology to syntax: Unsupervised linguistic typology at different levels with language embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 907–916. Association for Computational Linguistics.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. 2022. The geometry of multilingual language model representations. *CoRR*, abs/2205.10964.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3576–3588. Association for Computational Linguistics.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021b. Improving pretrained cross-lingual language models via self-labeled word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3418–3430. Association for Computational Linguistics.

Rochelle Choenni and Ekaterina Shutova. 2022. Investigating language relationships in multilingual sentence encoders through the lens of linguistic typology. *Comput. Linguistics*, 48(3):635–672.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Trans. Assoc. Comput. Linguistics*, 10:73–91.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

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. *CoRR*, abs/2205.12446.

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, Loïc 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. *CoRR*, abs/2207.04672.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Comput. Linguistics*, 47(2):255–308.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Matthew S Dryer and Martin Haspelmath. 2013. The world atlas of language structures online.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021a. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22:107:1–107:48.

Yimin Fan, Yaobo Liang, Alexandre Muzio, Hany Hassan, Houqiang Li, Ming Zhou, and Nan Duan. 2021b. Discovering representation sprachbund for multilingual pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 881–894. Association for Computational Linguistics.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gökhan Tür, and Prem Natarajan. 2022. MASSIVE: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *CoRR*, abs/2204.08582.

Harald Hammarström, Robert Forkel, and Martin Haspelmath. 2017. Glottolog 3.0. *Max Planck Institute for the Science of Human History*.

Martin Haspelmath. 2004. How hopeless is genealogical linguistics, and how advanced is areal linguistics? *Studies in Language*, 28(1):209–223.

Eric W Holman, Cecil H Brown, Søren Wichmann, André Müller, Viveka Velupillai, Harald Hammarström, Sebastian Sauppe, Hagen Jung, Dik Bakker, Pamela Brown, et al. 2011. Automated dating of the world's language families based on lexical similarity. *Current Anthropology*, 52(6):841–875.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Sakura Imai, Daisuke Kawahara, Naho Orita, and Hiromune Oda. 2023. Theoretical linguistics rivals embeddings in language clustering for multilingual named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 139–151. Association for Computational Linguistics.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André F. T. Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3651–3657. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulic, and Goran Glavas. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4483–4499. Association for Computational Linguistics.

Sheng Liang, Philipp Dufter, and Hinrich Schütze. 2021. Locating language-specific information in contextualized embeddings. *CoRR*, abs/2109.08040.

Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1663–1674. Association for Computational Linguistics.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019.

Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3125–3135. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori S. Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 8–14. Association for Computational Linguistics.

Chunlan Ma, Ayyoob ImaniGooghari, Haotian Ye, Ehsaneddin Asgari, and Hinrich Schütze. 2023. Taxi1500: A multilingual dataset for text classification in 1500 languages.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel bible corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 3158–3163. European Language Resources Association (ELRA).

Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena.

Steven Moran, Daniel McCloy, and Richard Wright. 2014. Phoible online.

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *CoRR*, abs/2202.08904.

Benjamin Müller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2214–2231. Association for Computational Linguistics.

Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2022. Cross-lingual retrieval augmented prompt for low-resource languages. *CoRR*, abs/2212.09651.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 644–649. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1946–1958. Association for Computational Linguistics.

Fred Philippy, Siwen Guo, and Shohreh Haddadan. 2023. Identifying the correlation between language distance and cross-lingual transfer in a multilingual representation space. *CoRR*, abs/2305.02151.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001. Association for Computational Linguistics.

Taraka Rama, Lisa Beinborn, and Steffen Eger. 2020. Probing multilingual BERT for genetic and typological signals. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1214–1228. International Committee on 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1627–1643. Association for Computational Linguistics.

Horacio Saggion, Sanja Štajner, Daniel Ferrés, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2023. Findings of the tsar-2022 shared task on multilingual lexical simplification. *arXiv preprint arXiv:2302.02888*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron

Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *CoRR*, abs/2204.07580.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 963–973. Association for Computational Linguistics.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Prakash Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast character transformers via gradient-based subword tokenization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2023. NLNDE at semeval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis. *CoRR*, abs/2305.00090.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 833–844. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Trans. Assoc. Comput. Linguistics*, 10:291–306.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.

Haotian Ye, Yihong Liu, and Hinrich Schütze. 2023. A study of conceptual language similarity: comparison and evaluation. *CoRR*, abs/2305.13401.

A Languages

Tab. 5-10 show the language list covered by mPLMs and corpora.

Tab. 11 provides the languages used for evaluating cross-lingual transfer.

	mBERT CANINE-S CANINE-C	XLM-R-Base XLM-R-Large	Glot500	mGPT	mT5-Base	XLM-Align	NLLB-200	XLS-R-300M	Flores	PBC	Fleurs
ace_Arab							✓		✓		
ace_Latn			✓				✓		✓	✓	
ach_Latn			✓							✓	
acm_Arab			✓				✓		✓		
acq_Arab							✓		✓		
acr_Latn			✓							✓	
aeb_Arab							✓		✓		
afr_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
agw_Latn			✓							✓	
ahk_Latn			✓							✓	
ajp_Arab			✓				✓		✓		
aka_Latn			✓				✓		✓	✓	
aln_Latn			✓							✓	
als_Latn			✓				✓		✓	✓	
alt_Cyrl			✓							✓	
alz_Latn			✓							✓	
amh_Ethi		✓	✓		✓	✓	✓	✓	✓	✓	✓
aoj_Latn			✓							✓	
apc_Arab			✓				✓		✓		
arb_Arab	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
arb_Latn							✓		✓		
arn_Latn			✓							✓	
ars_Arab							✓		✓		
ary_Arab			✓				✓		✓	✓	
arz_Arab			✓				✓		✓	✓	
asm_Beng		✓	✓			✓	✓	✓	✓	✓	✓
ast_Latn	✓		✓				✓		✓		✓
awa_Deva							✓		✓		
ayr_Latn			✓				✓		✓	✓	
azb_Arab	✓		✓				✓		✓	✓	
azj_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
bak_Cyrl	✓		✓	✓		✓	✓	✓	✓	✓	
bam_Latn			✓				✓		✓	✓	
ban_Latn			✓				✓		✓	✓	
bar_Latn	✓		✓							✓	
bba_Latn			✓							✓	
bbc_Latn			✓							✓	
bci_Latn			✓							✓	
bcl_Latn			✓							✓	
bel_Cyrl	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bem_Latn			✓				✓		✓	✓	
ben_Beng	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bho_Deva			✓				✓		✓		
bhw_Latn			✓							✓	
bim_Latn			✓							✓	
bis_Latn			✓							✓	
bjn_Arab							✓		✓		
bjn_Latn			✓				✓		✓		
bod_Tibt			✓				✓	✓	✓	✓	
bos_Latn	✓	✓	✓				✓	✓	✓		✓
bqc_Latn			✓							✓	
bre_Latn	✓	✓	✓					✓		✓	
bts_Latn			✓							✓	
btx_Latn			✓							✓	
bug_Latn							✓		✓		
bul_Cyrl	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bum_Latn			✓							✓	
bzj_Latn			✓							✓	
cab_Latn			✓							✓	
cac_Latn			✓							✓	
cak_Latn			✓							✓	
caq_Latn			✓							✓	
cat_Latn	✓	✓	✓		✓	✓	✓	✓	✓	✓	
cbk_Latn			✓							✓	
cce_Latn			✓							✓	
ceb_Latn	✓		✓		✓		✓	✓	✓	✓	✓
ces_Latn	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
cfm_Latn			✓							✓	
che_Cyrl	✓		✓							✓	
chk_Latn			✓							✓	
chv_Cyrl	✓		✓	✓				✓		✓	
cjk_Latn			✓				✓		✓		

Table 5: Languages covered by mPLMs and corpora.

	mBERT CANINE-S CANINE-C	XLM-R-Base XLM-R-Large	Glot500	mGPT	mT5-Base	XLM-Align	NLLB-200	XLS-R-300M	Flores	PBC	Fleurs
ckb_Arab		✓	✓		✓	✓	✓	✓	✓	✓	✓
ckb_Latn			✓							✓	
cmn_Hani	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
cnh_Latn			✓					✓		✓	
crh_Cyrl			✓							✓	
crh_Latn			✓				✓		✓		
crs_Latn			✓							✓	
csy_Latn			✓							✓	
ctd_Latn			✓							✓	
ctu_Latn			✓							✓	
cuk_Latn			✓							✓	
cym_Latn	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
dan_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
deu_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
dik_Latn							✓		✓		
djk_Latn			✓							✓	
dln_Latn			✓							✓	
dtp_Latn			✓							✓	
dyu_Latn			✓				✓		✓	✓	
dzo_Tibt			✓				✓		✓	✓	
efi_Latn			✓							✓	
ekk_Latn	✓	✓	✓		✓	✓	✓	✓	✓		✓
ell_Grek	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
eng_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
enm_Latn			✓							✓	
epo_Latn		✓	✓		✓	✓	✓	✓	✓	✓	
eus_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
ewe_Latn			✓				✓		✓	✓	
fao_Latn			✓				✓	✓	✓	✓	
fij_Latn			✓				✓		✓	✓	
fil_Latn			✓		✓					✓	
fin_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
fon_Latn			✓				✓		✓	✓	
fra_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
fry_Latn	✓	✓	✓		✓			✓		✓	
fur_Latn			✓				✓		✓		
fuv_Latn							✓		✓		
gaa_Latn			✓							✓	
gaz_Latn		✓	✓				✓		✓		
gil_Latn			✓							✓	
giz_Latn			✓							✓	
gkn_Latn			✓							✓	
gkp_Latn			✓							✓	
gla_Latn		✓	✓		✓		✓		✓	✓	
gle_Latn	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
glg_Latn	✓	✓	✓		✓	✓	✓	✓	✓		
glv_Latn			✓					✓		✓	
gom_Latn			✓							✓	
gor_Latn			✓							✓	
grc_Grek			✓							✓	
guc_Latn			✓							✓	
gug_Latn			✓				✓	✓	✓	✓	
guj_Gujr	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
gur_Latn			✓							✓	
guw_Latn			✓							✓	
gya_Latn			✓							✓	
gym_Latn			✓							✓	
hat_Latn	✓		✓		✓		✓	✓	✓	✓	
hau_Latn		✓	✓		✓		✓	✓	✓	✓	✓
haw_Latn			✓		✓			✓		✓	
heb_Hebr	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
hif_Latn			✓							✓	
hil_Latn			✓							✓	
hin_Deva	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
hin_Latn		✓	✓		✓					✓	
hmo_Latn			✓							✓	
hne_Deva			✓				✓		✓	✓	
hnj_Latn			✓		✓					✓	
hra_Latn			✓							✓	
hrv_Latn	✓	✓	✓			✓	✓	✓	✓	✓	✓
hui_Latn			✓							✓	
hun_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6: Languages covered by mPLMs and corpora.

	mBERT CANINE-S CANINE-C	XLM-R-Base XLM-R-Large	Glot500	mGPT	mT5-Base	XLM-Align	NLLB-200	XLS-R-300M	Flores	PBC	Fleurs
hus_Latn			✓							✓	
hye_Armn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
iba_Latn			✓							✓	
ibo_Latn			✓		✓		✓		✓	✓	✓
ifa_Latn			✓							✓	
ifb_Latn			✓							✓	
ikk_Latn			✓							✓	
ilo_Latn			✓				✓		✓	✓	
ind_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
isl_Latn	✓	✓	✓		✓	✓	✓	✓	✓	✓	
ita_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ium_Latn			✓							✓	
ixl_Latn			✓							✓	
izz_Latn			✓							✓	
jam_Latn			✓							✓	
jav_Latn	✓	✓	✓		✓		✓	✓	✓	✓	✓
jpn_Jpan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
kaa_Cyrl			✓							✓	
kaa_Latn			✓							✓	
kab_Latn			✓				✓	✓	✓	✓	
kac_Latn			✓				✓		✓	✓	
kal_Latn			✓							✓	
kam_Latn			✓				✓		✓		✓
kan_Knda	✓	✓	✓		✓	✓	✓	✓	✓	✓	
kas_Arab							✓		✓		
kas_Deva							✓		✓		
kat_Geor	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
kaz_Cyrl	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
kbp_Latn			✓				✓		✓	✓	
kea_Latn			✓				✓		✓		✓
kek_Latn			✓							✓	
khk_Cyrl							✓		✓		
khm_Khmr		✓	✓		✓	✓	✓	✓	✓	✓	
kia_Latn			✓							✓	
kik_Latn			✓				✓		✓	✓	
kin_Latn			✓				✓	✓	✓	✓	
kir_Cyrl	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
kjb_Latn			✓							✓	
kjh_Cyrl			✓							✓	
kmb_Latn			✓				✓		✓		
kmm_Latn			✓							✓	
kmr_Cyrl			✓							✓	
kmr_Latn			✓				✓		✓	✓	
knc_Arab							✓		✓		
knc_Latn							✓		✓		
kng_Latn			✓				✓		✓		
knv_Latn			✓							✓	
kor_Hang	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
kpg_Latn			✓							✓	
krc_Cyrl			✓							✓	
kri_Latn			✓							✓	
ksd_Latn			✓							✓	
kss_Latn			✓							✓	
ksw_Mymr			✓							✓	
kua_Latn			✓							✓	
lam_Latn			✓							✓	
lao_Laoo		✓	✓		✓	✓	✓	✓	✓	✓	
lat_Latn	✓	✓	✓		✓	✓		✓		✓	
lav_Latn	✓	✓	✓	✓	✓	✓		✓		✓	
ldi_Latn			✓							✓	
leh_Latn			✓							✓	
lhu_Latn			✓							✓	
lij_Latn			✓				✓		✓		
lim_Latn			✓				✓		✓		
lin_Latn			✓				✓	✓	✓	✓	✓
lit_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
lmo_Latn	✓		✓				✓		✓		
loz_Latn			✓							✓	
ltg_Latn							✓		✓		
ltz_Latn	✓		✓		✓		✓	✓	✓	✓	✓
lua_Latn			✓				✓		✓		
lug_Latn			✓				✓	✓	✓	✓	

Table 7: Languages covered by mPLMs and corpora.

	mBERT CANINE-S CANINE-C	XLM-R-Base XLM-R-Large	Glot500	mGPT	mT5-Base	XLM-Align	NLLB-200	XLS-R-300M	Flores	PBC	Fleurs
luo_Latn			✓				✓		✓	✓	
lus_Latn			✓				✓		✓	✓	
lvs_Latn			✓				✓		✓		
lzh_Hani			✓							✓	
mad_Latn			✓							✓	
mag_Deva							✓		✓		
mah_Latn			✓							✓	
mai_Deva			✓				✓		✓	✓	
mal_Mlym	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
mam_Latn			✓							✓	
mar_Deva	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
mau_Latn			✓							✓	
mbb_Latn			✓							✓	
mck_Latn			✓							✓	
mcn_Latn			✓							✓	
mco_Latn			✓							✓	
mdy_Ethi			✓							✓	
meu_Latn			✓							✓	
mfe_Latn			✓							✓	
mgh_Latn			✓							✓	
mgr_Latn			✓							✓	
mhr_Cyrl			✓							✓	
min_Arab							✓		✓		
min_Latn	✓		✓				✓		✓	✓	
miq_Latn			✓							✓	
mkd_Cyrl	✓	✓	✓		✓	✓	✓	✓	✓	✓	
mlt_Latn			✓		✓	✓	✓	✓	✓	✓	✓
mni_Beng							✓		✓		
mon_Cyrl		✓	✓	✓	✓	✓		✓			✓
mos_Latn			✓				✓		✓	✓	
mps_Latn			✓							✓	
mri_Latn			✓		✓		✓	✓	✓	✓	✓
mrw_Latn			✓							✓	
mwm_Latn			✓							✓	
mxv_Latn			✓							✓	
mya_Mymr	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
myv_Cyrl			✓							✓	
mzh_Latn			✓							✓	
nan_Latn			✓							✓	
naq_Latn			✓							✓	
nav_Latn			✓							✓	
nbl_Latn			✓							✓	
nch_Latn			✓							✓	
ncj_Latn			✓							✓	
ndc_Latn			✓							✓	
nde_Latn			✓							✓	
ndo_Latn			✓							✓	
nds_Latn	✓		✓							✓	
nep_Deva	✓	✓	✓		✓	✓		✓		✓	✓
ngu_Latn			✓							✓	
nia_Latn			✓							✓	
nld_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
nmf_Latn			✓							✓	
nnb_Latn			✓							✓	
nno_Latn	✓		✓			✓	✓	✓	✓	✓	
nob_Latn	✓		✓				✓		✓	✓	
nor_Latn		✓	✓		✓	✓		✓		✓	
npi_Deva			✓				✓		✓	✓	
nse_Latn			✓							✓	
nso_Latn			✓				✓		✓	✓	
nus_Latn							✓		✓		
nya_Latn			✓		✓		✓		✓	✓	✓
nyn_Latn			✓							✓	
nyy_Latn			✓							✓	
nzi_Latn			✓							✓	
oci_Latn	✓		✓				✓	✓	✓		✓
ory_Orya		✓	✓			✓	✓	✓	✓	✓	
oss_Cyrl			✓	✓						✓	
ote_Latn			✓							✓	
pag_Latn			✓				✓		✓	✓	
pam_Latn			✓							✓	
pan_Guru	✓	✓	✓		✓	✓	✓	✓	✓	✓	

Table 8: Languages covered by mPLMs and corpora.

	mBERT CANINE-S CANINE-C	XLM-R-Base XLM-R-Large	Glot500	mGPT	mT5-Base	XLM-Align	NLLB-200	XLS-R-300M	Flores	PBC	Fleurs
pap_Latn			✓				✓		✓	✓	
pau_Latn			✓							✓	
pbt_Arab							✓		✓		
pcm_Latn			✓							✓	
pdt_Latn			✓							✓	
pes_Arab	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
pis_Latn			✓							✓	
pls_Latn			✓							✓	
plt_Latn	✓	✓	✓		✓		✓	✓	✓	✓	
poh_Latn			✓							✓	
pol_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
pon_Latn			✓							✓	
por_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
prk_Latn			✓							✓	
prs_Arab			✓				✓		✓	✓	
pxm_Latn			✓							✓	
qub_Latn			✓							✓	
quc_Latn			✓							✓	
qug_Latn			✓							✓	
quh_Latn			✓							✓	
quw_Latn			✓							✓	
quy_Latn			✓				✓		✓	✓	
quz_Latn			✓							✓	
qvi_Latn			✓							✓	
rap_Latn			✓							✓	
rar_Latn			✓							✓	
rmy_Latn			✓							✓	
ron_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
rop_Latn			✓							✓	
rug_Latn			✓							✓	
run_Latn			✓				✓		✓	✓	
rus_Cyrl	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
sag_Latn			✓				✓		✓	✓	
sah_Cyrl			✓	✓				✓		✓	
san_Deva		✓	✓			✓	✓	✓	✓	✓	
san_Latn			✓							✓	
sat_Olck			✓				✓		✓		
sba_Latn			✓							✓	
scn_Latn	✓		✓				✓		✓		
seh_Latn			✓							✓	
shn_Mymr							✓		✓		
sin_Sinh		✓	✓		✓	✓	✓	✓	✓	✓	
slk_Latn	✓	✓	✓		✓	✓	✓	✓	✓	✓	
slv_Latn	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
sme_Latn			✓							✓	
smo_Latn			✓		✓		✓		✓	✓	
sna_Latn			✓		✓		✓	✓	✓	✓	✓
snd_Arab		✓	✓		✓	✓	✓	✓	✓	✓	✓
som_Latn		✓	✓		✓		✓	✓	✓	✓	✓
sop_Latn			✓							✓	
sot_Latn			✓		✓		✓		✓	✓	
spa_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
sqi_Latn	✓	✓	✓		✓	✓		✓		✓	
srm_Latn			✓							✓	
srn_Latn			✓							✓	
sro_Latn			✓				✓		✓		
srp_Cyrl	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
srp_Latn			✓							✓	
ssw_Latn			✓				✓		✓	✓	
sun_Latn	✓	✓	✓		✓		✓	✓	✓	✓	
suz_Deva			✓							✓	
swe_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
swh_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
sxn_Latn			✓							✓	
szl_Latn			✓				✓		✓		
tam_Latn		✓								✓	
tam_Taml	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
taq_Latn							✓		✓		
taq_Tfng							✓		✓		
tat_Cyrl	✓		✓	✓		✓	✓	✓	✓	✓	
tbz_Latn			✓							✓	
tca_Latn			✓							✓	

Table 9: Languages covered by mPLMs and corpora.

	mBERT CANINE-S CANINE-C	XLM-R-Base XLM-R-Large	Glot500	mGPT	mT5-Base	XLM-Align	NLLB-200	XLS-R-300M	Flores	PBC	Fleurs
tdt_Latn			✓							✓	
tel_Telu	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
teo_Latn			✓							✓	
tgk_Cyrl	✓		✓	✓	✓	✓	✓	✓	✓	✓	
tgl_Latn	✓	✓	✓	✓		✓	✓	✓	✓	✓	
tha_Thai		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
tih_Latn			✓							✓	
tir_Ethi			✓				✓		✓	✓	
tlh_Latn			✓							✓	
tob_Latn			✓							✓	
toh_Latn			✓							✓	
toi_Latn			✓							✓	
toj_Latn			✓							✓	
ton_Latn			✓							✓	
top_Latn			✓							✓	
tpi_Latn			✓				✓	✓	✓	✓	
tpm_Latn			✓							✓	
tsn_Latn			✓				✓		✓	✓	
tso_Latn			✓				✓		✓	✓	
tsz_Latn			✓							✓	
tuc_Latn			✓							✓	
tui_Latn			✓							✓	
tuk_Cyrl			✓							✓	
tuk_Latn			✓	✓			✓	✓	✓	✓	
tum_Latn			✓				✓		✓	✓	
tur_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
twi_Latn			✓				✓		✓	✓	
tyv_Cyrl			✓	✓						✓	
tzh_Latn			✓							✓	
tzm_Tfng							✓		✓		
tzo_Latn			✓							✓	
udm_Cyrl			✓							✓	
uig_Arab		✓	✓			✓	✓		✓	✓	
uig_Latn			✓							✓	
ukr_Cyrl	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
umb_Latn			✓				✓		✓		
urd_Arab	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
urd_Latn		✓								✓	
uzn_Cyrl			✓							✓	
uzn_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
vec_Latn			✓				✓		✓		
ven_Latn			✓							✓	
vie_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
wal_Latn			✓							✓	
war_Latn	✓		✓				✓	✓	✓	✓	
wol_Latn			✓				✓		✓	✓	
xav_Latn			✓							✓	
xho_Latn		✓	✓		✓		✓		✓	✓	✓
yan_Latn			✓							✓	
yao_Latn			✓							✓	
yap_Latn			✓							✓	
ydd_Hebr		✓	✓		✓	✓	✓	✓	✓		
yom_Latn			✓							✓	
yor_Latn	✓		✓	✓	✓		✓	✓	✓	✓	
yua_Latn			✓							✓	
yue_Hani			✓				✓	✓	✓	✓	
zai_Latn			✓							✓	
zlm_Latn			✓							✓	
zom_Latn			✓							✓	
zsm_Latn	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
zul_Latn			✓		✓		✓	✓	✓	✓	✓

Table 10: Languages covered by mPLMs and corpora.

Task	Language List
NER (108)	ace_Latn, afr_Latn, als_Latn, amh_Ethi, arz_Arab, asm_Beng, ast_Latn, azj_Latn, bak_Cyrl, bel_Cyrl, ben_Beng, bho_Deva, bod_Tibt, bos_Latn, bul_Cyrl, cat_Latn, ceb_Latn, ces_Latn, ckb_Arab, crh_Latn, cym_Latn, dan_Latn, deu_Latn, ekk_Latn, ell_Grek, epo_Latn, eus_Latn, fao_Latn, fin_Latn, fra_Latn, fur_Latn, gla_Latn, gle_Latn, glg_Latn, gug_Latn, guj_Gujr, heb_Hebr, hrv_Latn, hun_Latn, hye_Armn, ibo_Latn, ilo_Latn, ind_Latn, isl_Latn, ita_Latn, jav_Latn, jpn_Jpan, kan_Knda, kat_Geor, kaz_Cyrl, khm_Khmr, kin_Latn, kir_Cyrl, kor_Hang, lij_Latn, lim_Latn, lin_Latn, lit_Latn, lmo_Latn, ltz_Latn, mal_Mlym, mar_Deva, min_Latn, mkd_Cyrl, mlt_Latn, mri_Latn, mya_Mymr, nld_Latn, nno_Latn, oci_Latn, ory_Orya, pan_Guru, pes_Arab, plt_Latn, pol_Latn, por_Latn, ron_Latn, san_Deva, scn_Latn, sin_Sinh, slk_Latn, slv_Latn, snd_Arab, som_Latn, srp_Cyrl, sun_Latn, swe_Latn, swh_Latn, szl_Latn, tam_Taml, tat_Cyrl, tel_Telu, tgk_Cyrl, tgl_Latn, tha_Thai, tuk_Latn, tur_Latn, uig_Arab, ukr_Cyrl, urd_Arab, uzn_Latn, vec_Latn, vie_Latn, war_Latn, ydd_Hebr, yor_Latn, yue_Hani, zsm_Latn
POS (60)	afr_Latn, ajp_Arab, amh_Ethi, bam_Latn, bel_Cyrl, bho_Deva, bul_Cyrl, cat_Latn, ceb_Latn, ces_Latn, cym_Latn, dan_Latn, deu_Latn, ekk_Latn, ell_Grek, eus_Latn, fao_Latn, fin_Latn, fra_Latn, gla_Latn, gle_Latn, glg_Latn, heb_Hebr, hrv_Latn, hun_Latn, hye_Armn, ind_Latn, isl_Latn, ita_Latn, jav_Latn, jpn_Jpan, kaz_Cyrl, kmr_Latn, kor_Hang, lij_Latn, lit_Latn, mlt_Latn, nld_Latn, pes_Arab, pol_Latn, por_Latn, ron_Latn, san_Deva, sin_Sinh, slk_Latn, slv_Latn, swe_Latn, tam_Taml, tat_Cyrl, tel_Telu, tgl_Latn, tha_Thai, tur_Latn, uig_Arab, ukr_Cyrl, urd_Arab, vie_Latn, wol_Latn, yor_Latn, yue_Hani
Massive (44)	afr_Latn, als_Latn, amh_Ethi, azj_Latn, ben_Beng, cat_Latn, cym_Latn, dan_Latn, deu_Latn, ell_Grek, fin_Latn, fra_Latn, heb_Hebr, hun_Latn, hye_Armn, ind_Latn, isl_Latn, ita_Latn, jav_Latn, jpn_Jpan, kan_Knda, kat_Geor, khm_Khmr, kor_Hang, lvs_Latn, mal_Mlym, mya_Mymr, nld_Latn, nob_Latn, pes_Arab, pol_Latn, por_Latn, ron_Latn, slv_Latn, swe_Latn, swh_Latn, tam_Taml, tel_Telu, tgl_Latn, tha_Thai, tur_Latn, urd_Arab, vie_Latn, zsm_Latn
Taxi1500 (130)	ace_Latn, afr_Latn, aka_Latn, als_Latn, ary_Arab, arz_Arab, asm_Beng, ayr_Latn, azb_Arab, bak_Cyrl, bam_Latn, ban_Latn, bel_Cyrl, bem_Latn, ben_Beng, bul_Cyrl, cat_Latn, ceb_Latn, ces_Latn, ckb_Arab, cym_Latn, dan_Latn, deu_Latn, dyu_Latn, dzo_Tibt, ell_Grek, epo_Latn, eus_Latn, ewe_Latn, fao_Latn, fij_Latn, fin_Latn, fon_Latn, fra_Latn, gla_Latn, gle_Latn, gug_Latn, guj_Gujr, hat_Latn, hau_Latn, heb_Hebr, hne_Deva, hrv_Latn, hun_Latn, hye_Armn, ibo_Latn, ilo_Latn, ind_Latn, isl_Latn, ita_Latn, jav_Latn, kab_Latn, kac_Latn, kan_Knda, kat_Geor, kaz_Cyrl, kbp_Latn, khm_Khmr, kik_Latn, kin_Latn, kir_Cyrl, kng_Latn, kor_Hang, lao_Laoo, lin_Latn, lit_Latn, ltz_Latn, lug_Latn, luo_Latn, mai_Deva, mar_Deva, min_Latn, mkd_Cyrl, mlt_Latn, mos_Latn, mri_Latn, mya_Mymr, nld_Latn, nno_Latn, nob_Latn, npi_Deva, nso_Latn, nya_Latn, ory_Orya, pag_Latn, pan_Guru, pap_Latn, pes_Arab, plt_Latn, pol_Latn, por_Latn, prs_Arab, quy_Latn, ron_Latn, run_Latn, sag_Latn, sin_Sinh, slk_Latn, slv_Latn, smo_Latn, sna_Latn, snd_Arab, som_Latn, sot_Latn, ssw_Latn, sun_Latn, swe_Latn, swh_Latn, tam_Taml, tat_Cyrl, tel_Telu, tgk_Cyrl, tgl_Latn, tha_Thai, tir_Ethi, tpi_Latn, tsn_Latn, tuk_Latn, tum_Latn, tur_Latn, twi_Latn, ukr_Cyrl, vie_Latn, war_Latn, wol_Latn, xho_Latn, yor_Latn, yue_Hani, zsm_Latn, zul_Latn

Table 11: Languages for evaluating zero-shot cross-lingual transfer. The number in brackets is the number of the evaluated languages.

	mPLM-Sim	Mono	1	5	10
LEX	0.741	0.704	0.688	0.745	0.743
GEN	0.527	0.504	0.480	0.482	0.510
GEO	0.608	0.597	0.523	0.562	0.597
SYN	0.577	0.583	0.556	0.560	0.573
INV	0.248	0.245	0.226	0.265	0.260
PHO	0.094	0.109	0.114	0.118	0.102
FEA	0.358	0.369	0.347	0.371	0.360
AVG	**0.451**	0.444	0.419	0.444	0.449

Table 12: Comparison of pearson correlation result: Pearson correlation between seven similarity measurs and mPLM-Sim (500 multi-parallel sentences), Mono (Monolingual corpora) and the results of using different amounts (1, 5, 10) of multi-parallel sentences.

	mPLM-Sim	Mono	1	5	10
NER	0.647	0.644	0.644	0.646	0.647
POS	0.751	0.737	0.748	0.753	0.752
Massive	0.730	0.730	0.723	0.728	0.730
Taxi	0.583	0.585	0.580	0.582	0.582
AVG	**0.678**	0.674	0.674	0.677	**0.678**

Table 13: Comparison of cross-lingual transfer result: Cross-lingual transfer result for four tasks from mPLM-Sim (500 multi-parallel sentences), Mono (Monolingual corpora) and the results of using different amounts (1, 5, 10) of multi-parallel sentences.

B Comparison Across Corpora for mPLM-Sim

B.1 Monolingual vs. Parallel

Both monolingual and parallel corpora can be exploited for obtaining sentence embeddings for measuring language similarity. We conduct experiments of exploiting monolingual corpora for measuring similarity across languages, and also provide the results of using different amounts (1, 5, 10, 500) of multi-parallel sentences.

For the experiment of pearson correlation in Sec. 3.1, the results (MEAN) are shown in Tab. 12. For the experiment of cross-lingual transfer in Sec. 3.4, the results are shown in Tab. 13. Based on these two experiments, we have the conclusions below:

- mPLM-Sim using multi-parallel corpora achieves slightly better results than using monolingual corpora.
- mPLM-Sim (500 sentences) requires less data than exploiting monolingual corpora. Besides, using mPLM-Sim (10 sentences) can achieve comparable results with mPLM-Sim (500 sentences). While including a truly low-resource language for similarity measurement, mPLM-Sim requires around 10 sentences parallel to one existing language, while monolingual corpora requires massive sentences.

In a word, exploiting parallel corpora is better for measuring language similarity than monolingual corpora.

B.2 Flores vs. PBC

To investigate the impact of multi-parallel corpora on the performance of mPLM-Sim, we compare

	Flores		PBC	
	M	Mdn	M	Mdn
LEX	0.741	0.864	0.654	0.735
GEN	0.527	0.600	0.519	0.572
GEO	0.608	0.674	0.546	0.603
SYN	0.577	0.607	0.491	0.528
INV	0.248	0.293	0.254	0.276
PHO	0.094	0.144	0.103	0.098
FEA	0.358	0.372	0.333	0.357
AVG	0.451	0.508	0.414	0.453

Table 14: Comparison across corpora: Pearson correlation between mPLM-Sim and linguistic similarity measures for Glot500 and all corpora on 32 languages. Flores achieves higher correlations than PBC.

the results of Glot500 with Flores and PBC on 32 languages that are covered by both corpora.

Tab. 14 shows that Flores outperforms PBC across all similarity measures, except for PHO. To gain further insights, we conduct a case study focusing on languages that exhibit different performances between the two corpora.

In comparison to PBC, Flores consists of text that is closer to web content and spans a wider range of general domains. For example, a significant portion of Arabic script in Flores is written without short vowels, which are commonly used in texts requiring strict adherence to precise pronunciation, such as the Bible.[5] This discrepancy leads to challenges in tokenization and representation for languages written in Arabic, such as Moroccan Arabic (ary_Arab) and Egyptian Arabic (arz_Arab), resulting in poorer performance.

[5] https://en.wikipedia.org/wiki/Arabic_diacritics

C Visualization and Analysis Across Layers

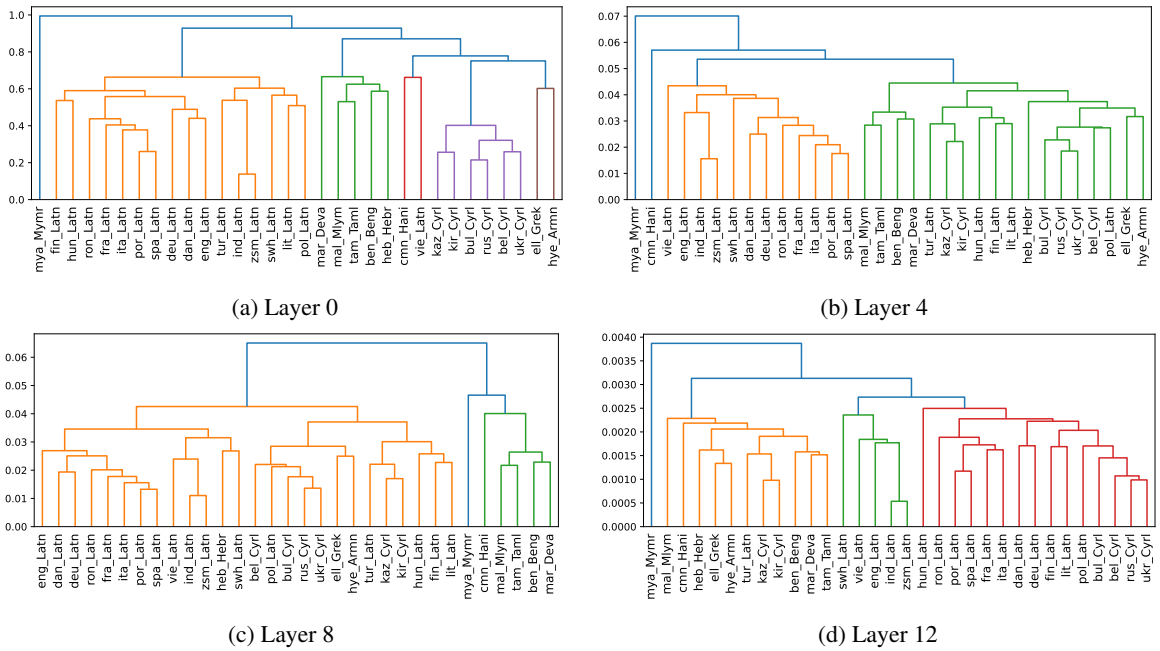
C.1 Hierarchical Clustering Analysis

We conducted hierarchical clustering analysis at different layers (0, 4, 8, and 12) using the setting of Glot500 and Flores for mPLM-Sim. The results, shown in Fig. 3, reveal distinct patterns of language clustering. In layer 0, the clustering primarily emphasizes lexical similarities, with languages sharing the same scripts being grouped together. As we progress to layers 4 and 8, more high-level similarity patterns beyond the surface-level are captured. For instance in these layers, Turkish (tur_Latn) and Polish (pol_Latn) are clustered with their Turkic and Slavic relatives although they use different writing systems. The similarity results of layer 12 are comparatively worse than those of the middle layers. For instance, English (eng_Latn) deviates from its Germanic and Indo-European relatives and instead clusters with Malay languages (ind_Latn, zsm_Latn). This phenomenon can be attributed to the higher layer exhibiting lower inter-cluster distances (comparison between the y-axis range across figures of different layers), which diminishes its ability to effectively discriminate between language clusters.

C.2 Similarity Heatmaps

Fig. 4-7 show the cosine simlarity values in heatmaps at layer 0, 4, 8 and 12, using the Glot500 and Flores settings for mPLM-Sim.

Generally, as the layer number increases, higher cosine similarity values are observed. Layer 0 exhibits a significant contrast in similarity values, whereas layer 12 demonstrates very low contrast. Notably, Burmese (mya_Mymr) consistently receives the lowest values across all layers, indicating the relationship between Burmese and other languages may be not well modeled.



(a) Layer 0 (b) Layer 4

(c) Layer 8 (d) Layer 12

Figure 3: Dendrograms illustrating hierarchical clustering results at layer 0, 4, 8, and 12 for Glot500 and Flores across 32 languages.

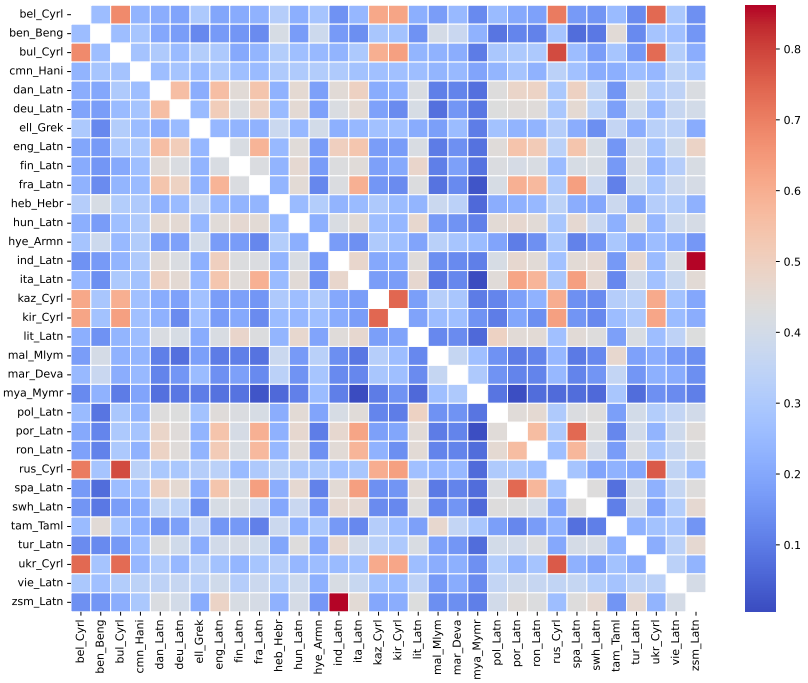


Figure 4: Heatmaps of cosine similarity results at layer 0 for Glot500 and Flores across 32 languages.

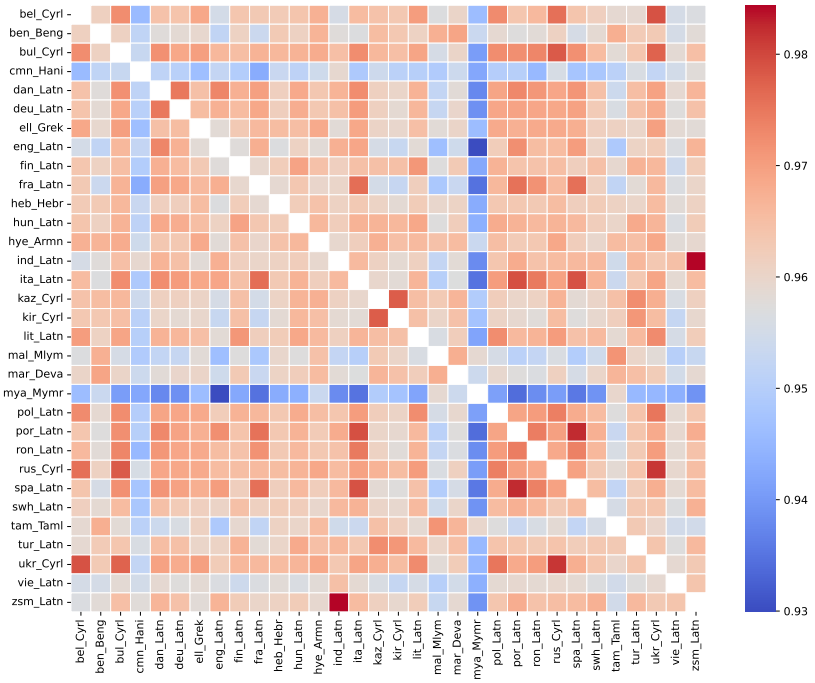


Figure 5: Heatmaps of cosine similarity results at layer 4 for Glot500 and Flores across 32 languages.

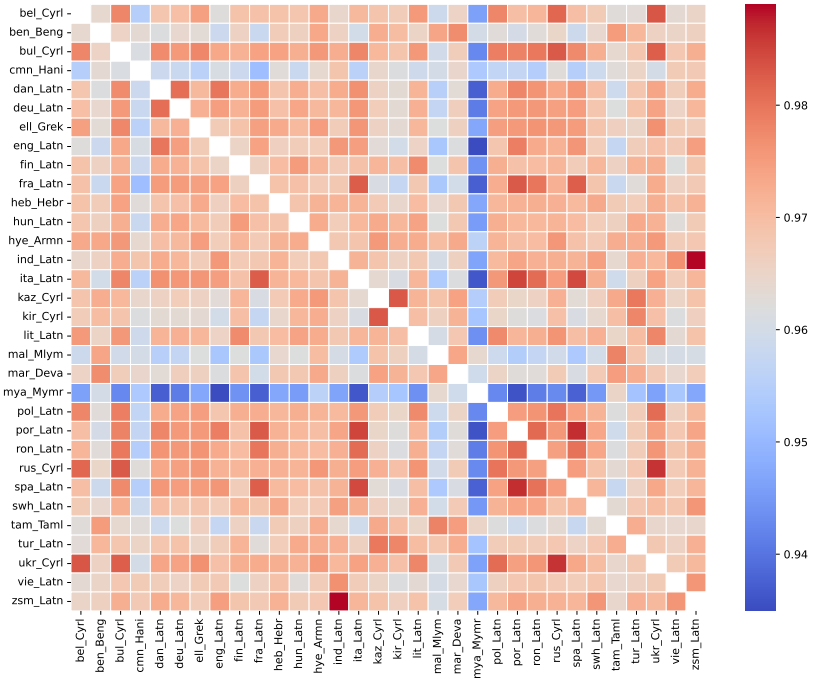


Figure 6: Heatmaps of cosine similarity results at layer 8 for Glot500 and Flores across 32 languages.

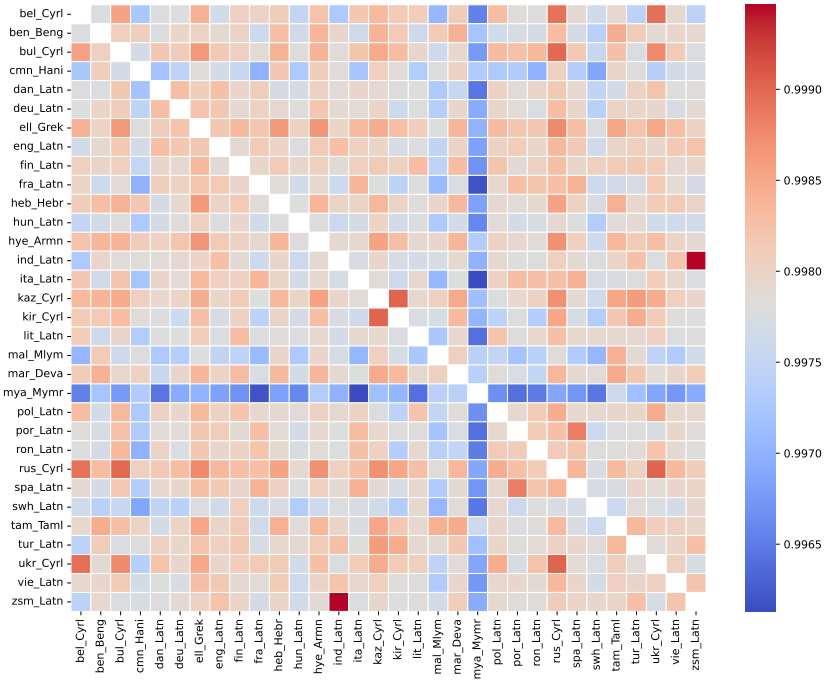


Figure 7: Heatmaps of cosine similarity results at layer 12 for Glot500 and Flores across 32 languages.

D Analysis on Unseen Languages of mPLMs

The success of mPLM-Sim depends on the supporting languages of mPLMs. To get more insights about languages which are this not supported by a specific mPLM, we conduct a new Pearson correlation experiment based on 94 languages unseen by XLM-R. Among 94 languages, there are 24 (25.5%) languages that achieve higher correlation than the average level of seen languages. These 24 languages usually have close languages seen by XLM-R, e.g, the unseen language, Cantonese (yue_Hani) is close to Mandarin (cmn_Hani). It shows that mPLM-Sim can be directly applied to some unseen languages which have close seen languages.

For the unseen languages which mPLM-Sim performs poorly, we can connect it to seen languages using traditional linguistic features, e.g., language family, and then use or weight the similarity results of seen languages as the results of the unseen languages. Since it is shown that mPLM-Sim provides better results than traditional linguistic features in our paper, connecting unseen languages to seen languages would be beneficial for unseen languages.

E Detailed Results of Cross-Lingual Transfer

We report the detailed results for all tasks and languages in Tab. 15-16 (NER), 17 (POS), 18 (MASSIVE), 19-21 (Taxi1500).

	ENG	LEX		GEN		GEO		FEA		mPLM-Sim	
ace_Latn	0.421	0.421	eng_Latn	0.421	eng_Latn	0.427	hin_Deva	0.421	eng_Latn	**0.439**	spa_Latn
afr_Latn	**0.739**	**0.739**	eng_Latn	**0.739**	eng_Latn	0.720	arb_Arab	0.707	rus_Cyrl	**0.739**	eng_Latn
als_Latn	0.767	0.767	eng_Latn	0.737	rus_Cyrl	**0.774**	spa_Latn	0.737	rus_Cyrl	**0.774**	spa_Latn
amh_Ethi	0.450	0.389	cmn_Hani	0.515	arb_Arab	0.515	arb_Arab	**0.554**	hin_Deva	**0.554**	hin_Deva
arz_Arab	0.491	**0.715**	arb_Arab	**0.715**	arb_Arab	**0.715**	arb_Arab	0.491	eng_Latn	**0.715**	arb_Arab
asm_Beng	0.661	0.603	arb_Arab	**0.720**	hin_Deva	**0.720**	hin_Deva	**0.720**	hin_Deva	**0.720**	hin_Deva
ast_Latn	0.813	**0.857**	spa_Latn	**0.857**	spa_Latn	**0.857**	spa_Latn	0.680	hin_Deva	**0.857**	spa_Latn
azj_Latn	0.625	0.625	eng_Latn	0.625	eng_Latn	**0.664**	arb_Arab	0.654	hin_Deva	0.648	spa_Latn
bak_Cyrl	0.558	0.675	rus_Cyrl	0.558	eng_Latn	0.675	rus_Cyrl	**0.681**	hin_Deva	0.675	rus_Cyrl
bel_Cyrl	0.728	**0.748**	rus_Cyrl	**0.748**	rus_Cyrl	0.728	eng_Latn	0.715	arb_Arab	**0.748**	rus_Cyrl
ben_Beng	0.670	0.647	arb_Arab	**0.692**	hin_Deva	**0.692**	hin_Deva	**0.692**	hin_Deva	**0.692**	hin_Deva
bho_Deva	0.544	**0.690**	hin_Deva	**0.690**	hin_Deva	**0.690**	hin_Deva	0.610	arb_Arab	**0.690**	hin_Deva
bod_Tibt	0.417	**0.544**	cmn_Hani	**0.544**	cmn_Hani	0.522	hin_Deva	**0.544**	cmn_Hani	**0.544**	cmn_Hani
bos_Latn	0.697	0.697	eng_Latn	**0.756**	rus_Cyrl	0.715	spa_Latn	0.702	arb_Arab	0.715	spa_Latn
bul_Cyrl	0.748	0.783	rus_Cyrl	0.783	rus_Cyrl	**0.787**	spa_Latn	0.783	rus_Cyrl	0.783	rus_Cyrl
cat_Latn	0.806	**0.808**	spa_Latn	**0.808**	spa_Latn	**0.808**	spa_Latn	0.806	eng_Latn	**0.808**	spa_Latn
ceb_Latn	**0.563**	**0.563**	eng_Latn	**0.563**	eng_Latn	0.211	cmn_Hani	0.530	spa_Latn	0.530	spa_Latn
ces_Latn	**0.760**	**0.760**	eng_Latn	0.741	rus_Cyrl	**0.760**	eng_Latn	0.741	rus_Cyrl	0.741	rus_Cyrl
ckb_Arab	0.707	**0.716**	arb_Arab	0.692	hin_Deva	**0.716**	arb_Arab	0.703	rus_Cyrl	**0.716**	arb_Arab
crh_Latn	0.521	0.521	eng_Latn	0.521	eng_Latn	0.472	arb_Arab	0.402	cmn_Hani	**0.551**	spa_Latn
cym_Latn	0.593	0.593	eng_Latn	0.617	rus_Cyrl	0.593	eng_Latn	0.542	arb_Arab	**0.636**	spa_Latn
dan_Latn	**0.792**	**0.792**	eng_Latn	**0.792**	eng_Latn	**0.792**	eng_Latn	0.747	arb_Arab	**0.792**	eng_Latn
deu_Latn	**0.714**	**0.714**	eng_Latn	**0.714**	eng_Latn	**0.714**	eng_Latn	**0.714**	eng_Latn	0.706	spa_Latn
ekk_Latn	0.713	0.713	eng_Latn	0.713	eng_Latn	0.713	eng_Latn	**0.729**	rus_Cyrl	0.729	spa_Latn
ell_Grek	0.686	0.686	eng_Latn	**0.733**	rus_Cyrl	0.729	spa_Latn	**0.733**	rus_Cyrl	**0.733**	rus_Cyrl
epo_Latn	0.639	0.639	eng_Latn	0.639	eng_Latn	0.639	eng_Latn	0.628	rus_Cyrl	**0.722**	spa_Latn
eus_Latn	0.516	0.516	eng_Latn	0.516	eng_Latn	0.552	spa_Latn	**0.588**	hin_Deva	0.552	spa_Latn
fao_Latn	0.706	0.706	eng_Latn	0.706	eng_Latn	0.706	eng_Latn	0.710	arb_Arab	**0.719**	spa_Latn
fin_Latn	0.728	0.728	eng_Latn	0.728	eng_Latn	0.728	eng_Latn	0.728	rus_Cyrl	**0.760**	spa_Latn
fra_Latn	0.730	0.730	eng_Latn	**0.805**	spa_Latn	0.730	eng_Latn	0.730	eng_Latn	**0.805**	spa_Latn
fur_Latn	0.567	0.567	eng_Latn	0.545	spa_Latn	0.567	eng_Latn	**0.605**	hin_Deva	0.545	spa_Latn
gla_Latn	0.571	0.571	eng_Latn	**0.612**	rus_Cyrl	0.571	eng_Latn	0.576	arb_Arab	0.582	spa_Latn
gle_Latn	0.670	0.670	eng_Latn	0.574	rus_Cyrl	0.670	eng_Latn	**0.688**	spa_Latn	**0.688**	spa_Latn
glg_Latn	0.768	**0.822**	spa_Latn	**0.822**	spa_Latn	**0.822**	spa_Latn	**0.822**	spa_Latn	**0.822**	spa_Latn
gug_Latn	0.552	0.552	eng_Latn	0.552	eng_Latn	**0.566**	spa_Latn	**0.566**	spa_Latn	**0.566**	spa_Latn
guj_Gujr	0.573	0.582	arb_Arab	**0.606**	hin_Deva	**0.606**	hin_Deva	**0.606**	hin_Deva	**0.606**	hin_Deva
heb_Hebr	0.458	0.300	cmn_Hani	**0.542**	arb_Arab	**0.542**	arb_Arab	0.463	rus_Cyrl	**0.542**	arb_Arab
hin_Deva	0.650	**0.697**	arb_Arab	**0.697**	arb_Arab	**0.697**	arb_Arab	**0.697**	arb_Arab	**0.697**	arb_Arab
hrv_Latn	0.738	0.738	eng_Latn	0.746	rus_Cyrl	0.738	eng_Latn	0.746	rus_Cyrl	**0.776**	spa_Latn
hun_Latn	0.727	0.727	eng_Latn	0.727	eng_Latn	0.727	eng_Latn	0.721	rus_Cyrl	**0.762**	spa_Latn
hye_Armn	0.518	**0.533**	arb_Arab	0.518	eng_Latn	**0.533**	arb_Arab	0.512	rus_Cyrl	0.531	hin_Deva
ibo_Latn	**0.574**	**0.574**	eng_Latn	**0.574**	eng_Latn	0.563	spa_Latn	**0.574**	eng_Latn	0.563	spa_Latn
ilo_Latn	0.673	0.673	eng_Latn	0.673	eng_Latn	0.577	cmn_Hani	0.673	eng_Latn	**0.716**	spa_Latn
ind_Latn	**0.594**	**0.594**	eng_Latn	**0.594**	eng_Latn	0.443	hin_Deva	**0.594**	eng_Latn	**0.594**	eng_Latn
isl_Latn	0.707	0.707	eng_Latn	0.707	eng_Latn	0.707	eng_Latn	0.707	eng_Latn	**0.726**	spa_Latn
ita_Latn	**0.764**	0.762	spa_Latn	0.762	spa_Latn	0.762	spa_Latn	0.762	spa_Latn	0.762	spa_Latn
jav_Latn	0.580	0.580	eng_Latn	0.580	eng_Latn	0.215	cmn_Hani	0.529	hin_Deva	**0.614**	spa_Latn
jpn_Jpan	0.177	**0.451**	cmn_Hani	0.177	eng_Latn	**0.451**	cmn_Hani	0.260	hin_Deva	**0.451**	cmn_Hani
kan_Knda	0.531	0.567	arb_Arab	0.531	eng_Latn	**0.638**	hin_Deva	**0.638**	hin_Deva	**0.638**	hin_Deva
kat_Geor	0.644	0.640	arb_Arab	0.644	eng_Latn	0.640	arb_Arab	**0.681**	hin_Deva	**0.681**	hin_Deva
kaz_Cyrl	0.416	**0.525**	rus_Cyrl	0.416	eng_Latn	**0.525**	rus_Cyrl	0.315	cmn_Hani	**0.525**	rus_Cyrl
khm_Khmr	0.404	0.404	eng_Latn	0.404	eng_Latn	0.467	hin_Deva	0.404	eng_Latn	**0.549**	arb_Arab
kin_Latn	0.626	0.626	eng_Latn	0.626	eng_Latn	0.672	arb_Arab	0.626	eng_Latn	**0.726**	spa_Latn
kir_Cyrl	0.391	**0.564**	rus_Cyrl	0.391	eng_Latn	**0.564**	rus_Cyrl	0.455	hin_Deva	**0.564**	rus_Cyrl
kor_Hang	0.470	0.445	cmn_Hani	0.470	eng_Latn	0.445	cmn_Hani	0.445	cmn_Hani	**0.551**	hin_Deva

Table 15: Cross-Lingual Transfer Results of NER (Part 1): The first column is the target language. For each language similarity measure, we report both the source language selected based on similarity and also the evaluation results on target language using the source language. For mPLM-Sim, we report the layer achieving best performance (layer 1).

	ENG	LEX		GEN		GEO		FEA		mPLM-Sim	
lij_Latn	**0.431**	**0.431**	eng_Latn	0.413	spa_Latn	0.413	spa_Latn	0.395	hin_Deva	0.413	spa_Latn
lim_Latn	**0.646**	**0.646**	eng_Latn	**0.646**	eng_Latn	**0.646**	eng_Latn	0.605	hin_Deva	0.621	spa_Latn
lin_Latn	0.486	0.486	eng_Latn	0.486	eng_Latn	**0.555**	arb_Arab	0.486	eng_Latn	0.519	spa_Latn
lit_Latn	**0.707**	**0.707**	eng_Latn	0.699	rus_Cyrl	**0.707**	eng_Latn	0.699	rus_Cyrl	0.699	rus_Cyrl
lmo_Latn	**0.712**	**0.712**	eng_Latn	0.706	spa_Latn	0.706	spa_Latn	0.559	hin_Deva	0.706	spa_Latn
ltz_Latn	0.646	0.646	eng_Latn	0.646	eng_Latn	0.646	eng_Latn	**0.663**	spa_Latn	**0.663**	spa_Latn
mal_Mlym	0.591	0.642	arb_Arab	0.591	eng_Latn	**0.709**	hin_Deva	**0.709**	hin_Deva	**0.709**	hin_Deva
mar_Deva	0.583	**0.725**	hin_Deva	**0.725**	hin_Deva	**0.725**	hin_Deva	**0.725**	hin_Deva	**0.725**	hin_Deva
min_Latn	0.405	0.405	eng_Latn	0.405	eng_Latn	0.363	hin_Deva	0.405	eng_Latn	**0.423**	spa_Latn
mkd_Cyrl	0.696	**0.767**	rus_Cyrl	**0.767**	rus_Cyrl	0.730	spa_Latn	**0.767**	rus_Cyrl	**0.767**	rus_Cyrl
mlt_Latn	0.667	0.667	eng_Latn	0.597	arb_Arab	**0.732**	spa_Latn	0.641	rus_Cyrl	**0.732**	spa_Latn
mri_Latn	0.531	0.531	eng_Latn	0.531	eng_Latn	0.433	cmn_Hani	0.531	eng_Latn	**0.572**	spa_Latn
mya_Mymr	0.493	**0.612**	arb_Arab	0.455	cmn_Hani	0.607	hin_Deva	0.493	eng_Latn	0.607	hin_Deva
nld_Latn	0.779	0.779	eng_Latn	0.779	eng_Latn	0.779	eng_Latn	0.779	eng_Latn	**0.781**	spa_Latn
nno_Latn	**0.762**	**0.762**	eng_Latn	**0.762**	eng_Latn	**0.762**	eng_Latn	0.686	hin_Deva	**0.762**	eng_Latn
oci_Latn	0.678	**0.802**	spa_Latn	**0.802**	spa_Latn	**0.802**	spa_Latn	**0.802**	spa_Latn	**0.802**	spa_Latn
ory_Orya	0.230	0.262	arb_Arab	**0.300**	hin_Deva	0.230	hin_Deva	**0.300**	hin_Deva	**0.300**	hin_Deva
pan_Guru	0.464	**0.470**	hin_Deva	**0.470**	hin_Deva	**0.470**	hin_Deva	**0.470**	hin_Deva	**0.470**	hin_Deva
pes_Arab	0.386	0.606	arb_Arab	**0.653**	hin_Deva	0.606	arb_Arab	**0.653**	hin_Deva	0.606	arb_Arab
plt_Latn	**0.533**	**0.533**	eng_Latn	**0.533**	eng_Latn	0.424	arb_Arab	0.510	rus_Cyrl	0.507	spa_Latn
pol_Latn	**0.754**	**0.754**	eng_Latn	0.719	rus_Cyrl	**0.754**	eng_Latn	0.719	rus_Cyrl	0.719	rus_Cyrl
por_Latn	0.745	**0.803**	spa_Latn	**0.803**	spa_Latn	**0.803**	spa_Latn	0.745	eng_Latn	**0.803**	spa_Latn
ron_Latn	0.632	0.632	eng_Latn	**0.746**	spa_Latn	0.632	eng_Latn	0.614	rus_Cyrl	**0.746**	spa_Latn
san_Deva	0.306	**0.523**	hin_Deva	**0.523**	hin_Deva	**0.523**	hin_Deva	**0.523**	hin_Deva	**0.523**	hin_Deva
scn_Latn	0.676	0.676	eng_Latn	**0.750**	spa_Latn	**0.750**	spa_Latn	0.623	arb_Arab	**0.750**	spa_Latn
sin_Sinh	0.536	0.560	arb_Arab	**0.727**	hin_Deva	**0.727**	hin_Deva	**0.727**	hin_Deva	**0.727**	hin_Deva
slk_Latn	**0.745**	**0.745**	eng_Latn	0.721	rus_Cyrl	**0.745**	eng_Latn	0.659	hin_Deva	0.721	rus_Cyrl
slv_Latn	**0.766**	**0.766**	eng_Latn	0.724	rus_Cyrl	**0.766**	eng_Latn	0.724	rus_Cyrl	0.724	rus_Cyrl
snd_Arab	0.374	0.441	arb_Arab	**0.530**	hin_Deva	**0.530**	hin_Deva	**0.530**	hin_Deva	0.441	arb_Arab
som_Latn	0.598	0.598	eng_Latn	0.562	arb_Arab	0.562	arb_Arab	0.579	hin_Deva	**0.605**	spa_Latn
srp_Cyrl	**0.627**	0.586	rus_Cyrl	0.586	rus_Cyrl	**0.627**	eng_Latn	0.586	rus_Cyrl	0.586	rus_Cyrl
sun_Latn	**0.577**	**0.577**	eng_Latn	**0.577**	eng_Latn	0.492	hin_Deva	**0.577**	eng_Latn	0.490	spa_Latn
swe_Latn	**0.632**	**0.632**	eng_Latn	**0.632**	eng_Latn	**0.632**	eng_Latn	**0.632**	eng_Latn	**0.632**	eng_Latn
swh_Latn	**0.687**	**0.687**	eng_Latn	**0.687**	eng_Latn	0.503	arb_Arab	0.662	spa_Latn	0.662	spa_Latn
szl_Latn	**0.670**	**0.670**	eng_Latn	0.655	rus_Cyrl	**0.670**	eng_Latn	0.631	hin_Deva	0.655	rus_Cyrl
tam_Taml	0.498	0.597	arb_Arab	0.498	eng_Latn	**0.626**	hin_Deva	**0.626**	hin_Deva	**0.626**	hin_Deva
tat_Cyrl	0.630	**0.715**	rus_Cyrl	0.630	eng_Latn	**0.715**	rus_Cyrl	0.672	arb_Arab	**0.715**	rus_Cyrl
tel_Telu	0.420	0.516	arb_Arab	0.420	eng_Latn	**0.539**	hin_Deva	**0.539**	hin_Deva	**0.539**	hin_Deva
tgk_Cyrl	0.588	**0.652**	rus_Cyrl	0.598	hin_Deva	**0.652**	rus_Cyrl	0.629	arb_Arab	**0.652**	rus_Cyrl
tgl_Latn	**0.745**	**0.745**	eng_Latn	**0.745**	eng_Latn	0.466	cmn_Hani	0.667	spa_Latn	0.667	spa_Latn
tha_Thai	0.049	**0.074**	cmn_Hani	0.049	eng_Latn	0.014	hin_Deva	0.049	eng_Latn	**0.074**	cmn_Hani
tuk_Latn	0.577	0.577	eng_Latn	0.577	eng_Latn	0.579	arb_Arab	0.553	cmn_Hani	**0.615**	spa_Latn
tur_Latn	0.712	0.712	eng_Latn	0.712	eng_Latn	0.707	arb_Arab	0.707	rus_Cyrl	**0.758**	spa_Latn
uig_Arab	0.460	**0.547**	arb_Arab	0.460	eng_Latn	0.525	rus_Cyrl	0.485	cmn_Hani	**0.547**	arb_Arab
ukr_Cyrl	0.695	**0.802**	rus_Cyrl	**0.802**	rus_Cyrl	0.695	eng_Latn	**0.802**	rus_Cyrl	**0.802**	rus_Cyrl
urd_Arab	0.596	0.689	arb_Arab	**0.743**	hin_Deva	**0.743**	hin_Deva	**0.743**	hin_Deva	**0.743**	hin_Deva
uzn_Latn	0.713	0.713	eng_Latn	0.713	eng_Latn	0.716	rus_Cyrl	0.479	hin_Deva	**0.792**	spa_Latn
vec_Latn	0.624	0.624	eng_Latn	**0.680**	spa_Latn	**0.680**	spa_Latn	0.549	hin_Deva	**0.680**	spa_Latn
vie_Latn	**0.654**	**0.654**	eng_Latn	**0.654**	eng_Latn	0.406	cmn_Hani	**0.654**	eng_Latn	0.546	rus_Cyrl
war_Latn	0.554	0.554	eng_Latn	0.554	eng_Latn	0.425	cmn_Hani	0.425	cmn_Hani	**0.585**	spa_Latn
ydd_Hebr	0.496	0.496	eng_Latn	0.496	eng_Latn	0.496	eng_Latn	**0.609**	hin_Deva	0.569	arb_Arab
yor_Latn	**0.614**	**0.614**	eng_Latn	**0.614**	eng_Latn	0.612	spa_Latn	0.532	rus_Cyrl	0.612	spa_Latn
yue_Hani	0.261	**0.635**	cmn_Hani	**0.635**	cmn_Hani	**0.635**	cmn_Hani	**0.635**	cmn_Hani	**0.635**	cmn_Hani
zsm_Latn	**0.654**	**0.654**	eng_Latn	**0.654**	eng_Latn	0.522	hin_Deva	**0.654**	eng_Latn	**0.654**	eng_Latn

Table 16: Cross-Lingual Transfer Results of NER (Part 2): The first column is the target language. For each language similarity measure, we report both the source language selected based on similarity and also the evaluation results on target language using the source language. For mPLM-Sim, we report the layer achieving best performance (layer 1).

	ENG	LEX		GEN		GEO		FEA		mPLM-Sim	
afr_Latn	0.850	0.850	eng_Latn	0.850	eng_Latn	0.599	arb_Arab	0.809	rus_Cyrl	**0.854**	spa_Latn
ajp_Arab	**0.671**	0.648	arb_Arab	0.648	arb_Arab	0.648	arb_Arab	0.651	hin_Deva	0.648	arb_Arab
amh_Ethi	0.648	0.645	cmn_Hani	0.670	arb_Arab	0.670	arb_Arab	**0.704**	hin_Deva	**0.704**	hin_Deva
bam_Latn	0.451	0.451	eng_Latn	0.451	eng_Latn	0.411	spa_Latn	**0.484**	hin_Deva	0.411	spa_Latn
bel_Cyrl	0.824	**0.934**	rus_Cyrl	**0.934**	rus_Cyrl	0.824	eng_Latn	0.719	arb_Arab	**0.934**	rus_Cyrl
ben_Beng	0.767	0.583	arb_Arab	**0.803**	hin_Deva	**0.803**	hin_Deva	**0.803**	hin_Deva	**0.803**	hin_Deva
bho_Deva	0.520	**0.682**	hin_Deva	**0.682**	hin_Deva	**0.682**	hin_Deva	0.536	arb_Arab	**0.682**	hin_Deva
bul_Cyrl	0.871	**0.899**	rus_Cyrl	**0.899**	rus_Cyrl	0.882	spa_Latn	**0.899**	rus_Cyrl	**0.899**	rus_Cyrl
cat_Latn	0.860	**0.962**	spa_Latn	**0.962**	spa_Latn	**0.962**	spa_Latn	0.860	eng_Latn	**0.962**	spa_Latn
ceb_Latn	0.605	0.605	eng_Latn	0.605	eng_Latn	0.481	cmn_Hani	**0.634**	spa_Latn	**0.634**	spa_Latn
ces_Latn	0.826	0.826	eng_Latn	**0.874**	rus_Cyrl	0.826	eng_Latn	**0.874**	rus_Cyrl	**0.874**	rus_Cyrl
cym_Latn	**0.621**	**0.621**	eng_Latn	0.612	rus_Cyrl	**0.621**	eng_Latn	0.602	arb_Arab	0.618	spa_Latn
dan_Latn	**0.873**	**0.873**	eng_Latn	**0.873**	eng_Latn	**0.873**	eng_Latn	0.640	arb_Arab	**0.873**	eng_Latn
deu_Latn	**0.850**	**0.850**	eng_Latn	**0.850**	eng_Latn	**0.850**	eng_Latn	**0.850**	eng_Latn	0.784	spa_Latn
ekk_Latn	**0.815**	**0.815**	eng_Latn	**0.815**	eng_Latn	**0.815**	eng_Latn	0.790	rus_Cyrl	0.790	rus_Cyrl
ell_Grek	0.822	0.822	eng_Latn	**0.871**	rus_Cyrl	0.834	spa_Latn	**0.871**	rus_Cyrl	**0.871**	rus_Cyrl
eus_Latn	0.625	0.625	eng_Latn	0.625	eng_Latn	0.681	spa_Latn	**0.702**	hin_Deva	0.681	spa_Latn
fao_Latn	0.869	0.869	eng_Latn	0.869	eng_Latn	0.869	eng_Latn	0.701	arb_Arab	**0.876**	spa_Latn
fin_Latn	0.771	0.771	eng_Latn	0.771	eng_Latn	0.771	eng_Latn	**0.773**	rus_Cyrl	**0.773**	rus_Cyrl
fra_Latn	0.838	0.838	eng_Latn	**0.885**	spa_Latn	0.838	eng_Latn	0.838	eng_Latn	**0.885**	spa_Latn
gla_Latn	0.571	0.571	eng_Latn	**0.588**	rus_Cyrl	0.571	eng_Latn	0.498	arb_Arab	0.548	spa_Latn
gle_Latn	0.578	0.578	eng_Latn	**0.624**	rus_Cyrl	0.578	eng_Latn	0.624	spa_Latn	0.624	spa_Latn
glg_Latn	0.796	**0.864**	spa_Latn	**0.864**	spa_Latn	**0.864**	spa_Latn	**0.864**	spa_Latn	**0.864**	spa_Latn
gug_Latn	0.213	0.213	eng_Latn	0.213	eng_Latn	**0.256**	spa_Latn	**0.256**	spa_Latn	**0.256**	spa_Latn
heb_Hebr	0.636	0.560	cmn_Hani	0.696	arb_Arab	0.696	arb_Arab	**0.704**	rus_Cyrl	0.696	arb_Arab
hin_Deva	**0.665**	0.612	arb_Arab	0.612	arb_Arab	0.612	arb_Arab	0.612	arb_Arab	0.612	arb_Arab
hrv_Latn	0.829	0.829	eng_Latn	**0.899**	rus_Cyrl	0.829	eng_Latn	**0.899**	rus_Cyrl	**0.899**	rus_Cyrl
hun_Latn	0.801	0.801	eng_Latn	0.801	eng_Latn	0.801	eng_Latn	0.740	rus_Cyrl	**0.811**	spa_Latn
hye_Armn	0.817	0.595	arb_Arab	0.817	eng_Latn	0.595	arb_Arab	**0.846**	rus_Cyrl	**0.846**	rus_Cyrl
ind_Latn	**0.814**	**0.814**	eng_Latn	**0.814**	eng_Latn	0.695	hin_Deva	**0.814**	eng_Latn	**0.814**	eng_Latn
isl_Latn	**0.805**	**0.805**	eng_Latn	**0.805**	eng_Latn	**0.805**	eng_Latn	**0.805**	eng_Latn	0.802	spa_Latn
ita_Latn	0.852	**0.906**	spa_Latn	**0.906**	spa_Latn	**0.906**	spa_Latn	**0.906**	spa_Latn	**0.906**	spa_Latn
jav_Latn	**0.742**	**0.742**	eng_Latn	**0.742**	eng_Latn	0.543	cmn_Hani	0.645	hin_Deva	0.731	spa_Latn
jpn_Jpan	0.165	**0.534**	cmn_Hani	0.165	eng_Latn	**0.534**	cmn_Hani	0.402	hin_Deva	**0.534**	cmn_Hani
kaz_Cyrl	0.724	**0.739**	rus_Cyrl	0.724	eng_Latn	**0.739**	rus_Cyrl	0.545	cmn_Hani	**0.739**	rus_Cyrl
kmr_Latn	0.748	0.748	eng_Latn	0.719	hin_Deva	0.646	arb_Arab	0.748	eng_Latn	**0.777**	spa_Latn
kor_Hang	**0.497**	0.447	cmn_Hani	**0.497**	eng_Latn	0.447	cmn_Hani	0.447	cmn_Hani	0.491	hin_Deva
lij_Latn	0.739	0.739	eng_Latn	**0.819**	spa_Latn	**0.819**	spa_Latn	0.685	hin_Deva	**0.819**	spa_Latn
lit_Latn	0.787	0.787	eng_Latn	**0.840**	rus_Cyrl	0.787	eng_Latn	**0.840**	rus_Cyrl	**0.840**	rus_Cyrl
mal_Mlym	**0.847**	0.680	arb_Arab	**0.847**	eng_Latn	0.804	hin_Deva	0.804	hin_Deva	0.804	hin_Deva
mar_Deva	0.813	**0.830**	hin_Deva	**0.830**	hin_Deva	**0.830**	hin_Deva	**0.830**	hin_Deva	**0.830**	hin_Deva
mlt_Latn	0.776	0.776	eng_Latn	0.603	arb_Arab	**0.798**	spa_Latn	0.787	rus_Cyrl	**0.798**	spa_Latn
nld_Latn	**0.874**	**0.874**	eng_Latn	**0.874**	eng_Latn	**0.874**	eng_Latn	**0.874**	eng_Latn	0.855	spa_Latn
pes_Arab	0.675	0.690	arb_Arab	**0.709**	hin_Deva	0.690	arb_Arab	**0.709**	hin_Deva	0.690	arb_Arab
pol_Latn	0.791	0.791	eng_Latn	**0.881**	rus_Cyrl	0.791	eng_Latn	**0.881**	rus_Cyrl	**0.881**	rus_Cyrl
por_Latn	0.857	**0.910**	spa_Latn	**0.910**	spa_Latn	**0.910**	spa_Latn	0.857	eng_Latn	**0.910**	spa_Latn
ron_Latn	0.747	0.747	eng_Latn	**0.816**	spa_Latn	0.747	eng_Latn	0.794	rus_Cyrl	**0.816**	spa_Latn
san_Deva	0.217	**0.319**	hin_Deva	**0.319**	hin_Deva	**0.319**	hin_Deva	**0.319**	hin_Deva	**0.319**	hin_Deva
sin_Sinh	0.546	0.520	arb_Arab	**0.652**	hin_Deva	**0.652**	hin_Deva	**0.652**	hin_Deva	**0.652**	hin_Deva
slk_Latn	0.820	0.820	eng_Latn	**0.865**	rus_Cyrl	0.820	eng_Latn	0.743	hin_Deva	**0.865**	rus_Cyrl
slv_Latn	0.743	0.743	eng_Latn	**0.805**	rus_Cyrl	0.743	eng_Latn	**0.805**	rus_Cyrl	**0.805**	rus_Cyrl
swe_Latn	**0.891**	**0.891**	eng_Latn	**0.891**	eng_Latn	**0.891**	eng_Latn	**0.891**	eng_Latn	**0.891**	eng_Latn
tam_Taml	0.733	0.586	arb_Arab	0.733	eng_Latn	**0.771**	hin_Deva	**0.771**	hin_Deva	**0.771**	hin_Deva
tat_Cyrl	0.675	**0.692**	rus_Cyrl	0.675	eng_Latn	**0.692**	rus_Cyrl	0.587	arb_Arab	**0.692**	rus_Cyrl
tel_Telu	**0.791**	0.653	arb_Arab	**0.791**	eng_Latn	0.781	hin_Deva	0.781	hin_Deva	0.781	hin_Deva
tgl_Latn	0.695	0.695	eng_Latn	0.695	eng_Latn	0.416	cmn_Hani	**0.719**	spa_Latn	**0.719**	spa_Latn
tha_Thai	**0.502**	0.499	cmn_Hani	**0.502**	eng_Latn	0.453	hin_Deva	**0.502**	eng_Latn	0.499	cmn_Hani
tur_Latn	0.671	0.671	eng_Latn	0.671	eng_Latn	0.522	arb_Arab	0.671	rus_Cyrl	**0.697**	spa_Latn
uig_Arab	0.660	0.536	arb_Arab	0.660	eng_Latn	0.670	rus_Cyrl	0.525	cmn_Hani	**0.687**	hin_Deva
ukr_Cyrl	0.821	**0.918**	rus_Cyrl	**0.918**	rus_Cyrl	0.821	eng_Latn	**0.918**	rus_Cyrl	**0.918**	rus_Cyrl
urd_Arab	0.589	0.580	arb_Arab	**0.889**	hin_Deva	**0.889**	hin_Deva	**0.889**	hin_Deva	**0.889**	hin_Deva
vie_Latn	0.648	0.648	eng_Latn	0.648	eng_Latn	0.442	cmn_Hani	0.648	eng_Latn	**0.658**	rus_Cyrl
wol_Latn	0.606	0.606	eng_Latn	0.606	eng_Latn	**0.679**	spa_Latn	0.606	eng_Latn	**0.679**	spa_Latn
yor_Latn	0.644	0.644	eng_Latn	0.644	eng_Latn	0.651	spa_Latn	**0.658**	rus_Cyrl	0.651	spa_Latn
yue_Hani	0.196	**0.787**	cmn_Hani	**0.787**	cmn_Hani	**0.787**	cmn_Hani	**0.787**	cmn_Hani	**0.787**	cmn_Hani

Table 17: Cross-Lingual Transfer Results of POS: The first column is the target language. For each language similarity measure, we report both the source language selected based on similarity and also the evaluation results on target language using the source language. For mPLM-Sim, we report the layer achieving best performance (layer 2).

	ENG	LEX		GEN		GEO		FEA		mPLM-Sim	
afr_Latn	**0.732**	**0.732**	eng_Latn	**0.732**	eng_Latn	0.589	arb_Arab	0.701	rus_Cyrl	**0.732**	eng_Latn
als_Latn	0.708	0.708	eng_Latn	0.721	rus_Cyrl	**0.727**	spa_Latn	**0.727**	spa_Latn	**0.727**	spa_Latn
amh_Ethi	0.557	0.470	cmn_Hani	0.532	arb_Arab	0.532	arb_Arab	**0.611**	hin_Deva	**0.611**	hin_Deva
azj_Latn	0.773	0.773	eng_Latn	0.773	eng_Latn	0.705	arb_Arab	**0.793**	hin_Deva	**0.793**	hin_Deva
ben_Beng	0.676	0.625	arb_Arab	**0.768**	hin_Deva	**0.768**	hin_Deva	**0.768**	hin_Deva	**0.768**	hin_Deva
cat_Latn	0.731	**0.833**	spa_Latn	**0.833**	spa_Latn	**0.833**	spa_Latn	0.731	eng_Latn	**0.833**	spa_Latn
cym_Latn	0.492	0.492	eng_Latn	**0.495**	rus_Cyrl	0.492	eng_Latn	0.433	arb_Arab	0.480	spa_Latn
dan_Latn	**0.838**	**0.838**	eng_Latn	**0.838**	eng_Latn	**0.838**	eng_Latn	0.720	arb_Arab	**0.838**	eng_Latn
deu_Latn	**0.759**	**0.759**	eng_Latn	**0.759**	eng_Latn	**0.759**	eng_Latn	**0.759**	eng_Latn	0.726	spa_Latn
ell_Grek	0.715	0.715	eng_Latn	**0.729**	rus_Cyrl	0.717	spa_Latn	**0.729**	rus_Cyrl	**0.729**	rus_Cyrl
fin_Latn	0.677	0.677	eng_Latn	0.677	eng_Latn	0.677	eng_Latn	**0.701**	rus_Cyrl	**0.701**	rus_Cyrl
fra_Latn	0.812	0.812	eng_Latn	**0.816**	spa_Latn	0.812	eng_Latn	0.812	eng_Latn	**0.816**	spa_Latn
heb_Hebr	0.697	0.576	cmn_Hani	0.691	arb_Arab	0.691	arb_Arab	**0.714**	rus_Cyrl	0.691	arb_Arab
hun_Latn	0.673	0.673	eng_Latn	0.673	eng_Latn	0.673	eng_Latn	**0.698**	rus_Cyrl	**0.698**	rus_Cyrl
hye_Armn	**0.781**	0.729	arb_Arab	**0.781**	eng_Latn	0.729	arb_Arab	0.780	rus_Cyrl	0.780	rus_Cyrl
ind_Latn	**0.819**	**0.819**	eng_Latn	**0.819**	eng_Latn	0.779	hin_Deva	**0.819**	eng_Latn	**0.819**	eng_Latn
isl_Latn	0.658	0.658	eng_Latn	0.658	eng_Latn	0.658	eng_Latn	0.658	eng_Latn	**0.664**	rus_Cyrl
ita_Latn	0.772	**0.817**	spa_Latn	**0.817**	spa_Latn	**0.817**	spa_Latn	**0.817**	spa_Latn	**0.817**	spa_Latn
jav_Latn	**0.507**	**0.507**	eng_Latn	**0.507**	eng_Latn	0.416	cmn_Hani	0.504	hin_Deva	0.495	spa_Latn
jpn_Jpan	0.384	**0.448**	cmn_Hani	0.384	eng_Latn	**0.448**	cmn_Hani	0.363	hin_Deva	**0.448**	cmn_Hani
kan_Knda	0.682	0.628	arb_Arab	0.682	eng_Latn	**0.729**	hin_Deva	**0.729**	hin_Deva	**0.729**	hin_Deva
kat_Geor	0.618	0.605	arb_Arab	0.618	eng_Latn	0.605	arb_Arab	**0.620**	hin_Deva	**0.620**	hin_Deva
khm_Khmr	**0.655**	**0.655**	eng_Latn	**0.655**	eng_Latn	0.636	hin_Deva	**0.655**	eng_Latn	0.611	arb_Arab
kor_Hang	0.758	0.643	cmn_Hani	0.758	eng_Latn	0.643	cmn_Hani	0.643	cmn_Hani	**0.768**	hin_Deva
lvs_Latn	0.661	0.661	eng_Latn	0.661	eng_Latn	0.661	eng_Latn	0.651	hin_Deva	**0.722**	rus_Cyrl
mal_Mlym	0.717	0.678	arb_Arab	0.717	eng_Latn	**0.764**	hin_Deva	**0.764**	hin_Deva	**0.764**	hin_Deva
mya_Mymr	0.688	0.656	arb_Arab	0.616	cmn_Hani	**0.707**	hin_Deva	0.688	eng_Latn	**0.707**	hin_Deva
nld_Latn	**0.813**	**0.813**	eng_Latn	**0.813**	eng_Latn	**0.813**	eng_Latn	**0.813**	eng_Latn	**0.813**	eng_Latn
nob_Latn	**0.847**	**0.847**	eng_Latn	**0.847**	eng_Latn	**0.847**	eng_Latn	**0.847**	eng_Latn	**0.847**	eng_Latn
pes_Arab	**0.831**	0.780	arb_Arab	0.817	hin_Deva	0.780	arb_Arab	0.817	hin_Deva	0.817	hin_Deva
pol_Latn	0.768	0.768	eng_Latn	**0.788**	rus_Cyrl	0.768	eng_Latn	**0.788**	rus_Cyrl	**0.788**	rus_Cyrl
por_Latn	0.793	**0.839**	spa_Latn	**0.839**	spa_Latn	**0.839**	spa_Latn	0.793	eng_Latn	**0.839**	spa_Latn
ron_Latn	0.791	0.791	eng_Latn	**0.814**	spa_Latn	0.791	eng_Latn	0.790	rus_Cyrl	**0.814**	spa_Latn
slv_Latn	0.643	0.643	eng_Latn	**0.720**	rus_Cyrl	0.643	eng_Latn	**0.720**	rus_Cyrl	**0.720**	rus_Cyrl
swe_Latn	**0.834**	**0.834**	eng_Latn	**0.834**	eng_Latn	**0.834**	eng_Latn	**0.834**	eng_Latn	**0.834**	eng_Latn
swh_Latn	0.465	0.465	eng_Latn	0.465	eng_Latn	0.468	arb_Arab	**0.499**	spa_Latn	**0.499**	spa_Latn
tam_Taml	0.698	0.657	arb_Arab	0.698	eng_Latn	**0.737**	hin_Deva	**0.737**	hin_Deva	**0.737**	hin_Deva
tel_Telu	0.695	0.657	arb_Arab	0.695	eng_Latn	**0.756**	hin_Deva	**0.756**	hin_Deva	**0.756**	hin_Deva
tgl_Latn	**0.752**	**0.752**	eng_Latn	**0.752**	eng_Latn	0.648	cmn_Hani	0.723	spa_Latn	0.723	spa_Latn
tha_Thai	**0.791**	0.714	cmn_Hani	**0.791**	eng_Latn	0.752	hin_Deva	**0.791**	eng_Latn	0.714	cmn_Hani
tur_Latn	0.747	0.747	eng_Latn	0.747	eng_Latn	0.650	arb_Arab	0.731	rus_Cyrl	**0.786**	hin_Deva
urd_Arab	0.716	0.686	arb_Arab	**0.806**	hin_Deva	**0.806**	hin_Deva	**0.806**	hin_Deva	**0.806**	hin_Deva
vie_Latn	**0.771**	**0.771**	eng_Latn	**0.771**	eng_Latn	0.680	cmn_Hani	**0.771**	eng_Latn	**0.771**	eng_Latn
zsm_Latn	**0.754**	**0.754**	eng_Latn	**0.754**	eng_Latn	0.731	hin_Deva	**0.754**	eng_Latn	**0.754**	eng_Latn

Table 18: Cross-Lingual Transfer Result of MASSIVE: The first column is the target language. For each language similarity measure, we report both the source language selected based on similarity and also the evaluation results on target language using the source language. For mPLM-Sim, we report the layer achieving best performance (layer 8).

	ENG	LEX		GEN		GEO		FEA		mPLM-Sim	
ace_Latn	0.624	0.624	eng_Latn	0.624	eng_Latn	**0.726**	hin_Deva	0.624	eng_Latn	0.654	spa_Latn
afr_Latn	0.600	0.600	eng_Latn	0.600	eng_Latn	0.455	arb_Arab	0.522	rus_Cyrl	**0.604**	spa_Latn
aka_Latn	**0.518**	**0.518**	eng_Latn	**0.518**	eng_Latn	0.471	spa_Latn	0.469	hin_Deva	0.471	spa_Latn
als_Latn	**0.575**	**0.575**	eng_Latn	0.557	rus_Cyrl	0.536	spa_Latn	0.557	rus_Cyrl	0.536	spa_Latn
ary_Arab	0.421	**0.484**	arb_Arab	**0.484**	arb_Arab	0.465	spa_Latn	0.421	eng_Latn	**0.484**	arb_Arab
arz_Arab	0.325	**0.430**	arb_Arab	**0.430**	arb_Arab	**0.430**	arb_Arab	0.325	eng_Latn	**0.430**	arb_Arab
asm_Beng	0.574	0.548	arb_Arab	**0.600**	hin_Deva	**0.600**	hin_Deva	**0.600**	hin_Deva	**0.600**	hin_Deva
ayr_Latn	**0.694**	**0.694**	eng_Latn	**0.694**	eng_Latn	0.645	spa_Latn	0.564	cmn_Hani	0.685	hin_Deva
azb_Arab	0.527	0.585	arb_Arab	0.527	eng_Latn	0.585	arb_Arab	**0.639**	hin_Deva	**0.639**	hin_Deva
bak_Cyrl	0.632	**0.667**	rus_Cyrl	0.632	eng_Latn	**0.667**	rus_Cyrl	0.635	hin_Deva	**0.667**	rus_Cyrl
bam_Latn	0.487	0.487	eng_Latn	0.487	eng_Latn	**0.617**	spa_Latn	0.531	hin_Deva	**0.617**	spa_Latn
ban_Latn	0.446	0.446	eng_Latn	0.446	eng_Latn	0.483	cmn_Hani	**0.497**	hin_Deva	0.489	spa_Latn
bel_Cyrl	**0.622**	0.571	rus_Cyrl	0.571	rus_Cyrl	**0.622**	eng_Latn	0.530	arb_Arab	0.571	rus_Cyrl
bem_Latn	0.418	0.418	eng_Latn	0.418	eng_Latn	0.477	arb_Arab	**0.517**	spa_Latn	**0.517**	spa_Latn
ben_Beng	**0.667**	0.568	arb_Arab	0.634	hin_Deva	0.634	hin_Deva	0.634	hin_Deva	0.634	hin_Deva
bul_Cyrl	0.612	**0.618**	rus_Cyrl	**0.618**	rus_Cyrl	0.574	spa_Latn	**0.618**	rus_Cyrl	**0.618**	rus_Cyrl
cat_Latn	0.496	**0.614**	spa_Latn	**0.614**	spa_Latn	**0.614**	spa_Latn	0.496	eng_Latn	**0.614**	spa_Latn
ceb_Latn	0.565	0.565	eng_Latn	0.565	eng_Latn	**0.565**	cmn_Hani	0.456	spa_Latn	0.456	spa_Latn
ces_Latn	**0.620**	**0.620**	eng_Latn	0.577	rus_Cyrl	**0.620**	eng_Latn	0.577	rus_Cyrl	0.577	rus_Cyrl
ckb_Arab	0.544	0.539	arb_Arab	**0.622**	hin_Deva	0.539	arb_Arab	0.589	rus_Cyrl	0.539	arb_Arab
cym_Latn	0.488	0.488	eng_Latn	0.435	rus_Cyrl	0.488	eng_Latn	0.469	arb_Arab	**0.501**	spa_Latn
dan_Latn	**0.556**	**0.556**	eng_Latn	**0.556**	eng_Latn	**0.556**	eng_Latn	0.401	arb_Arab	**0.556**	eng_Latn
deu_Latn	0.559	0.559	eng_Latn	0.559	eng_Latn	0.559	eng_Latn	0.559	eng_Latn	**0.561**	spa_Latn
dyu_Latn	0.520	0.520	eng_Latn	0.520	eng_Latn	**0.587**	spa_Latn	0.568	hin_Deva	**0.587**	spa_Latn
dzo_Tibt	0.495	0.612	arb_Arab	**0.682**	cmn_Hani	0.681	hin_Deva	0.681	hin_Deva	0.681	hin_Deva
ell_Grek	0.532	0.532	eng_Latn	**0.547**	rus_Cyrl	0.485	spa_Latn	**0.547**	rus_Cyrl	**0.547**	rus_Cyrl
epo_Latn	**0.548**	**0.548**	eng_Latn	**0.548**	eng_Latn	**0.548**	eng_Latn	0.511	rus_Cyrl	0.530	spa_Latn
eus_Latn	0.196	0.196	eng_Latn	0.196	eng_Latn	**0.299**	spa_Latn	0.268	hin_Deva	**0.299**	spa_Latn
ewe_Latn	0.480	0.480	eng_Latn	0.480	eng_Latn	**0.589**	spa_Latn	0.530	hin_Deva	**0.589**	spa_Latn
fao_Latn	**0.658**	**0.658**	eng_Latn	**0.658**	eng_Latn	**0.658**	eng_Latn	0.591	arb_Arab	0.526	spa_Latn
fij_Latn	0.512	0.512	eng_Latn	0.512	eng_Latn	0.525	cmn_Hani	**0.576**	spa_Latn	**0.576**	spa_Latn
fin_Latn	0.465	0.465	eng_Latn	0.465	eng_Latn	0.465	eng_Latn	**0.518**	rus_Cyrl	**0.518**	rus_Cyrl
fon_Latn	0.462	0.462	eng_Latn	0.462	eng_Latn	**0.562**	spa_Latn	0.462	eng_Latn	**0.562**	spa_Latn
fra_Latn	0.566	0.566	eng_Latn	**0.627**	spa_Latn	0.566	eng_Latn	0.566	eng_Latn	**0.627**	spa_Latn
gla_Latn	0.489	0.489	eng_Latn	0.476	rus_Cyrl	0.489	eng_Latn	0.464	arb_Arab	**0.503**	spa_Latn
gle_Latn	0.375	0.375	eng_Latn	0.387	rus_Cyrl	0.375	eng_Latn	**0.502**	spa_Latn	**0.502**	spa_Latn
gug_Latn	0.396	0.396	eng_Latn	0.396	eng_Latn	**0.561**	spa_Latn	**0.561**	spa_Latn	**0.561**	spa_Latn
guj_Gujr	**0.717**	0.646	arb_Arab	0.680	hin_Deva	0.680	hin_Deva	0.680	hin_Deva	0.680	hin_Deva
hat_Latn	0.571	0.571	eng_Latn	**0.644**	spa_Latn	0.571	eng_Latn	0.584	arb_Arab	**0.644**	spa_Latn
hau_Latn	0.486	0.486	eng_Latn	**0.560**	arb_Arab	0.550	spa_Latn	0.486	eng_Latn	0.550	spa_Latn
heb_Hebr	**0.398**	0.391	cmn_Hani	0.359	arb_Arab	0.359	arb_Arab	0.373	rus_Cyrl	0.359	arb_Arab
hin_Deva	**0.705**	0.618	arb_Arab	0.618	arb_Arab	0.618	arb_Arab	0.618	arb_Arab	0.618	arb_Arab
hne_Deva	0.708	**0.711**	hin_Deva	**0.711**	hin_Deva	**0.711**	hin_Deva	**0.711**	hin_Deva	**0.711**	hin_Deva
hrv_Latn	0.569	0.569	eng_Latn	**0.680**	rus_Cyrl	0.569	eng_Latn	**0.680**	rus_Cyrl	**0.680**	rus_Cyrl
hun_Latn	0.540	0.540	eng_Latn	0.540	eng_Latn	0.540	eng_Latn	**0.609**	rus_Cyrl	**0.609**	rus_Cyrl

Table 19: Cross-Lingual Transfer Results of Taxi1500 (Part 1): The first column is the target language. For each language similarity measure, we report both the source language selected based on similarity and also the evaluation results on target language using the source language. For mPLM-Sim, we report the layer achieving best performance (layer 4).

	ENG	LEX		GEN		GEO		FEA		mPLM-Sim	
hye_Armn	0.650	**0.678**	arb_Arab	0.650	eng_Latn	**0.678**	arb_Arab	0.654	rus_Cyrl	0.654	rus_Cyrl
ibo_Latn	0.544	0.544	eng_Latn	0.544	eng_Latn	**0.566**	spa_Latn	0.544	eng_Latn	**0.566**	spa_Latn
ilo_Latn	0.511	0.511	eng_Latn	0.511	eng_Latn	0.463	cmn_Hani	0.511	eng_Latn	**0.591**	spa_Latn
ind_Latn	0.720	0.720	eng_Latn	0.720	eng_Latn	**0.795**	hin_Deva	0.720	eng_Latn	0.720	eng_Latn
isl_Latn	0.497	0.497	eng_Latn	0.497	eng_Latn	0.497	eng_Latn	0.497	eng_Latn	**0.602**	spa_Latn
ita_Latn	**0.608**	0.593	spa_Latn	0.593	spa_Latn	0.593	spa_Latn	0.593	spa_Latn	0.593	spa_Latn
jav_Latn	0.445	0.445	eng_Latn	0.445	eng_Latn	0.428	cmn_Hani	0.441	hin_Deva	**0.516**	spa_Latn
kab_Latn	0.259	0.259	eng_Latn	0.368	arb_Arab	**0.396**	spa_Latn	0.259	eng_Latn	**0.396**	spa_Latn
kac_Latn	0.451	0.451	eng_Latn	**0.580**	cmn_Hani	0.483	hin_Deva	**0.580**	cmn_Hani	0.483	hin_Deva
kan_Knda	**0.673**	0.637	arb_Arab	**0.673**	eng_Latn	0.640	hin_Deva	0.640	hin_Deva	0.640	hin_Deva
kat_Geor	0.558	0.464	arb_Arab	0.558	eng_Latn	0.464	arb_Arab	**0.672**	hin_Deva	**0.672**	hin_Deva
kaz_Cyrl	0.587	**0.636**	rus_Cyrl	0.587	eng_Latn	**0.636**	rus_Cyrl	0.629	hin_Deva	**0.636**	rus_Cyrl
kbp_Latn	0.357	0.357	eng_Latn	0.357	eng_Latn	0.361	spa_Latn	0.357	eng_Latn	**0.378**	hin_Deva
khm_Khmr	0.653	0.653	eng_Latn	0.653	eng_Latn	**0.679**	hin_Deva	0.653	eng_Latn	**0.679**	hin_Deva
kik_Latn	0.384	0.384	eng_Latn	0.384	eng_Latn	0.456	arb_Arab	**0.555**	spa_Latn	**0.555**	spa_Latn
kin_Latn	0.431	0.431	eng_Latn	0.431	eng_Latn	0.530	arb_Arab	0.431	eng_Latn	**0.619**	spa_Latn
kir_Cyrl	0.623	0.601	rus_Cyrl	0.623	eng_Latn	0.601	rus_Cyrl	**0.750**	hin_Deva	0.601	rus_Cyrl
kng_Latn	0.353	0.353	eng_Latn	0.353	eng_Latn	**0.455**	arb_Arab	**0.455**	arb_Arab	0.381	spa_Latn
kor_Hang	0.614	0.602	cmn_Hani	0.614	eng_Latn	0.602	cmn_Hani	0.602	cmn_Hani	**0.686**	hin_Deva
lao_Laoo	0.689	0.689	eng_Latn	0.689	eng_Latn	**0.711**	cmn_Hani	0.689	eng_Latn	**0.711**	cmn_Hani
lin_Latn	0.504	0.504	eng_Latn	0.504	eng_Latn	**0.541**	arb_Arab	0.504	eng_Latn	0.450	spa_Latn
lit_Latn	0.566	0.566	eng_Latn	**0.594**	rus_Cyrl	0.566	eng_Latn	**0.594**	rus_Cyrl	**0.594**	rus_Cyrl
ltz_Latn	0.546	0.546	eng_Latn	0.546	eng_Latn	0.546	eng_Latn	**0.547**	spa_Latn	**0.547**	spa_Latn
lug_Latn	0.474	0.474	eng_Latn	0.474	eng_Latn	**0.564**	arb_Arab	0.510	spa_Latn	0.510	spa_Latn
luo_Latn	0.394	0.394	eng_Latn	0.394	eng_Latn	**0.435**	arb_Arab	0.394	eng_Latn	0.427	spa_Latn
mai_Deva	0.698	**0.724**	hin_Deva	**0.724**	hin_Deva	**0.724**	hin_Deva	**0.724**	hin_Deva	**0.724**	hin_Deva
mar_Deva	**0.720**	0.665	hin_Deva	0.665	hin_Deva	0.665	hin_Deva	0.665	hin_Deva	0.665	hin_Deva
min_Latn	0.482	0.482	eng_Latn	0.482	eng_Latn	0.464	hin_Deva	0.482	eng_Latn	**0.552**	spa_Latn
mkd_Cyrl	**0.701**	0.648	rus_Cyrl	0.648	rus_Cyrl	0.629	spa_Latn	0.648	rus_Cyrl	0.648	rus_Cyrl
mlt_Latn	0.503	0.503	eng_Latn	0.519	arb_Arab	0.527	spa_Latn	**0.556**	rus_Cyrl	0.527	spa_Latn
mos_Latn	0.360	0.360	eng_Latn	0.360	eng_Latn	**0.506**	spa_Latn	0.360	eng_Latn	**0.506**	spa_Latn
mri_Latn	**0.522**	**0.522**	eng_Latn	**0.522**	eng_Latn	0.391	cmn_Hani	**0.522**	eng_Latn	0.484	spa_Latn
mya_Mymr	0.581	0.574	arb_Arab	0.537	cmn_Hani	**0.674**	hin_Deva	0.581	eng_Latn	**0.674**	hin_Deva
nld_Latn	**0.713**	**0.713**	eng_Latn	**0.713**	eng_Latn	**0.713**	eng_Latn	**0.713**	eng_Latn	0.628	spa_Latn
nno_Latn	**0.704**	**0.704**	eng_Latn	**0.704**	eng_Latn	**0.704**	eng_Latn	0.691	hin_Deva	**0.704**	eng_Latn
nob_Latn	**0.656**	**0.656**	eng_Latn	**0.656**	eng_Latn	**0.656**	eng_Latn	**0.656**	eng_Latn	**0.656**	eng_Latn
npi_Deva	0.694	**0.712**	hin_Deva	**0.712**	hin_Deva	0.694	eng_Latn	**0.712**	hin_Deva	**0.712**	hin_Deva
nso_Latn	0.514	0.514	eng_Latn	0.514	eng_Latn	0.519	arb_Arab	0.519	arb_Arab	**0.564**	spa_Latn
nya_Latn	0.560	0.560	eng_Latn	0.560	eng_Latn	0.584	arb_Arab	0.584	arb_Arab	**0.624**	spa_Latn
ory_Orya	**0.698**	0.635	arb_Arab	0.683	hin_Deva	**0.698**	eng_Latn	0.683	hin_Deva	0.683	hin_Deva
pag_Latn	**0.618**	**0.618**	eng_Latn	**0.618**	eng_Latn	0.572	cmn_Hani	0.610	spa_Latn	0.610	spa_Latn
pan_Guru	**0.709**	0.675	hin_Deva	0.675	hin_Deva	0.675	hin_Deva	0.675	hin_Deva	0.675	hin_Deva
pap_Latn	0.572	0.572	eng_Latn	0.538	spa_Latn	0.538	spa_Latn	**0.607**	arb_Arab	0.538	spa_Latn
pes_Arab	0.624	0.619	arb_Arab	**0.668**	hin_Deva	0.619	arb_Arab	**0.668**	hin_Deva	**0.668**	hin_Deva

Table 20: Cross-Lingual Transfer Results of Taxi1500 (Part 2): The first column is the target language. For each language similarity measure, we report both the source language selected based on similarity and also the evaluation results on target language using the source language. For mPLM-Sim, we report the layer achieving best performance (layer 4).

	ENG	LEX		GEN		GEO		FEA		mPLM-Sim	
plt_Latn	0.503	0.503	eng_Latn	0.503	eng_Latn	0.495	arb_Arab	**0.627**	rus_Cyrl	0.562	spa_Latn
pol_Latn	**0.690**	**0.690**	eng_Latn	0.690	rus_Cyrl	**0.690**	eng_Latn	0.690	rus_Cyrl	0.690	rus_Cyrl
por_Latn	**0.615**	0.605	spa_Latn	0.605	spa_Latn	0.605	spa_Latn	**0.615**	eng_Latn	0.605	spa_Latn
prs_Arab	0.677	0.653	arb_Arab	0.665	hin_Deva	0.665	hin_Deva	**0.691**	cmn_Hani	0.665	hin_Deva
quy_Latn	0.696	0.696	eng_Latn	0.696	eng_Latn	0.693	spa_Latn	**0.718**	hin_Deva	0.693	spa_Latn
ron_Latn	0.582	0.582	eng_Latn	**0.617**	spa_Latn	0.582	eng_Latn	0.589	rus_Cyrl	**0.617**	spa_Latn
run_Latn	0.470	0.470	eng_Latn	0.470	eng_Latn	0.508	arb_Arab	**0.546**	hin_Deva	0.504	spa_Latn
sag_Latn	0.476	0.476	eng_Latn	0.476	eng_Latn	**0.491**	arb_Arab	0.476	eng_Latn	0.442	spa_Latn
sin_Sinh	0.582	0.652	arb_Arab	**0.663**	hin_Deva	**0.663**	hin_Deva	**0.663**	hin_Deva	**0.663**	hin_Deva
slk_Latn	0.568	0.568	eng_Latn	0.592	rus_Cyrl	0.568	eng_Latn	**0.635**	hin_Deva	0.592	rus_Cyrl
slv_Latn	0.635	0.635	eng_Latn	**0.718**	rus_Cyrl	0.635	eng_Latn	**0.718**	rus_Cyrl	**0.718**	rus_Cyrl
smo_Latn	0.600	0.600	eng_Latn	0.600	eng_Latn	**0.630**	cmn_Hani	0.549	arb_Arab	0.625	spa_Latn
sna_Latn	0.443	0.443	eng_Latn	0.443	eng_Latn	0.444	arb_Arab	**0.555**	spa_Latn	**0.555**	spa_Latn
snd_Arab	0.694	0.621	arb_Arab	**0.726**	hin_Deva	**0.726**	hin_Deva	**0.726**	hin_Deva	**0.726**	hin_Deva
som_Latn	0.355	0.355	eng_Latn	0.454	arb_Arab	0.454	arb_Arab	0.424	hin_Deva	**0.485**	spa_Latn
sot_Latn	0.441	0.441	eng_Latn	0.441	eng_Latn	**0.537**	arb_Arab	**0.537**	arb_Arab	0.516	spa_Latn
ssw_Latn	0.437	0.437	eng_Latn	0.437	eng_Latn	0.424	arb_Arab	0.424	arb_Arab	**0.497**	spa_Latn
sun_Latn	0.493	0.493	eng_Latn	0.493	eng_Latn	**0.548**	hin_Deva	0.493	eng_Latn	0.514	spa_Latn
swe_Latn	**0.665**	**0.665**	eng_Latn	**0.665**	eng_Latn	**0.665**	eng_Latn	**0.665**	eng_Latn	**0.665**	eng_Latn
swh_Latn	**0.642**	**0.642**	eng_Latn	**0.642**	eng_Latn	0.558	arb_Arab	0.574	spa_Latn	0.574	spa_Latn
tam_Taml	0.684	0.643	arb_Arab	0.684	eng_Latn	**0.695**	hin_Deva	**0.695**	hin_Deva	**0.695**	hin_Deva
tat_Cyrl	**0.670**	0.664	rus_Cyrl	**0.670**	eng_Latn	0.664	rus_Cyrl	0.648	arb_Arab	0.664	rus_Cyrl
tel_Telu	0.557	0.594	arb_Arab	0.557	eng_Latn	**0.684**	hin_Deva	**0.684**	hin_Deva	**0.684**	hin_Deva
tgk_Cyrl	0.490	**0.724**	rus_Cyrl	0.493	hin_Deva	**0.724**	rus_Cyrl	0.426	arb_Arab	**0.724**	rus_Cyrl
tgl_Latn	**0.628**	**0.628**	eng_Latn	**0.628**	eng_Latn	0.563	cmn_Hani	0.567	spa_Latn	0.567	spa_Latn
tha_Thai	0.600	**0.669**	cmn_Hani	0.600	eng_Latn	0.651	hin_Deva	0.600	eng_Latn	**0.669**	cmn_Hani
tir_Ethi	0.487	0.497	cmn_Hani	0.531	arb_Arab	0.531	arb_Arab	**0.601**	hin_Deva	**0.601**	hin_Deva
tpi_Latn	**0.621**	**0.621**	eng_Latn	**0.621**	eng_Latn	0.579	cmn_Hani	**0.621**	eng_Latn	0.609	spa_Latn
tsn_Latn	0.397	0.397	eng_Latn	0.397	eng_Latn	0.447	arb_Arab	0.413	cmn_Hani	**0.495**	spa_Latn
tuk_Latn	0.537	0.537	eng_Latn	0.537	eng_Latn	**0.649**	arb_Arab	0.592	cmn_Hani	0.604	hin_Deva
tum_Latn	0.559	0.559	eng_Latn	0.559	eng_Latn	0.528	arb_Arab	**0.642**	hin_Deva	0.533	spa_Latn
tur_Latn	0.609	0.609	eng_Latn	0.609	eng_Latn	0.602	arb_Arab	0.615	rus_Cyrl	**0.640**	hin_Deva
twi_Latn	**0.532**	**0.532**	eng_Latn	**0.532**	eng_Latn	0.507	spa_Latn	**0.532**	eng_Latn	0.507	spa_Latn
ukr_Cyrl	0.506	**0.558**	rus_Cyrl	**0.558**	rus_Cyrl	0.506	eng_Latn	**0.558**	rus_Cyrl	**0.558**	rus_Cyrl
vie_Latn	0.642	0.642	eng_Latn	0.642	eng_Latn	**0.656**	cmn_Hani	0.642	eng_Latn	0.614	rus_Cyrl
war_Latn	0.449	0.449	eng_Latn	0.449	eng_Latn	0.472	cmn_Hani	0.472	cmn_Hani	**0.505**	spa_Latn
wol_Latn	0.396	0.396	eng_Latn	0.396	eng_Latn	**0.400**	spa_Latn	0.396	eng_Latn	**0.400**	spa_Latn
xho_Latn	0.486	0.486	eng_Latn	0.486	eng_Latn	**0.507**	arb_Arab	0.486	eng_Latn	0.422	spa_Latn
yor_Latn	0.542	0.542	eng_Latn	0.542	eng_Latn	0.556	spa_Latn	**0.584**	rus_Cyrl	0.556	spa_Latn
yue_Hani	0.577	**0.718**	cmn_Hani	**0.718**	cmn_Hani	**0.718**	cmn_Hani	**0.718**	cmn_Hani	**0.718**	cmn_Hani
zsm_Latn	0.658	0.658	eng_Latn	0.658	eng_Latn	**0.694**	hin_Deva	0.658	eng_Latn	0.658	eng_Latn
zul_Latn	0.504	0.504	eng_Latn	0.504	eng_Latn	0.527	arb_Arab	0.526	rus_Cyrl	**0.529**	spa_Latn

Table 21: Cross-Lingual Transfer Results of Taxi1500 (Part 3). The first column is the target language. For each language similarity measure, we report both the source language selected based on similarity and also the evaluation results on target language using the source language. For mPLM-Sim, we report the layer achieving best performance (layer 4).

Chapter 7

XAMPLER: Learning to Retrieve Cross-Lingual In-Context Examples

XAMPLER: Learning to Retrieve Cross-Lingual In-Context Examples

Peiqin Lin[1,2], **André F. T. Martins**[3,4,5], **Hinrich Schütze**[1,2]
[1]Center for Information and Language Processing, LMU Munich
[2]Munich Center for Machine Learning
[3]Instituto Superior Técnico, Universidade de Lisboa (Lisbon ELLIS Unit)
[4]Instituto de Telecomunicações [5]Unbabel
linpq@cis.lmu.de

Abstract

Recent studies indicate that leveraging off-the-shelf or fine-tuned retrievers, capable of retrieving relevant in-context examples tailored to the input query, enhances few-shot in-context learning for English. However, adapting these methods to other languages, especially low-resource ones, poses challenges due to the scarcity of cross-lingual retrievers and annotated data. Thus, we introduce **XAMPLER: Cross-Lingual Example Retrieval**, a method tailored to tackle the challenge of cross-lingual in-context learning **using only annotated English data**. XAMPLER first trains a retriever based on Glot500, a multilingual small language model, using positive and negative English examples constructed from the predictions of a multilingual large language model, i.e., MaLA500. Leveraging the cross-lingual capacity of the retriever, it can directly retrieve English examples as few-shot examples for in-context learning of target languages. Experiments on two multilingual text classification benchmarks, namely SIB200 with 176 languages and MasakhaNEWS with 16 languages, demonstrate that XAMPLER substantially improves the in-context learning performance across languages. Our code is available at https://github.com/cisnlp/XAMPLER.

1 Introduction

Large language models (LLMs) have shown emergent abilities in in-context learning, where a few input-output examples are provided with the input query. Through in-context learning, LLMs can yield promising results without any parameter updates (Brown et al., 2020). However, the efficacy of in-context learning is highly dependent on the selection of the few-shot examples (Liu et al., 2022).

Recent studies (Luo et al., 2024) have uncovered a more strategic approach to example retrieval. Rather than relying on random selection, these studies advocate for retrieving examples tailored to the input query, resulting in notable performance enhancements in in-context learning. The retrievers employed by these methods can be categorized into two main types: general off-the-shelf retrievers (Liu et al., 2022), e.g., Sentence-BERT (Reimers and Gurevych, 2019), and task-specific fine-tuned retrievers (Rubin et al., 2022), which are trained based on LLM signals (whether an example is helpful) using labeled data.

Utilizing off-the-shelf retrievers has been further validated as an effective approach in multilingual settings (Nie et al., 2022; Winata et al., 2023; Tanwar et al., 2023). However, this method encounters limitations when applied to low-resource languages. Existing multilingual retrievers, e.g.,

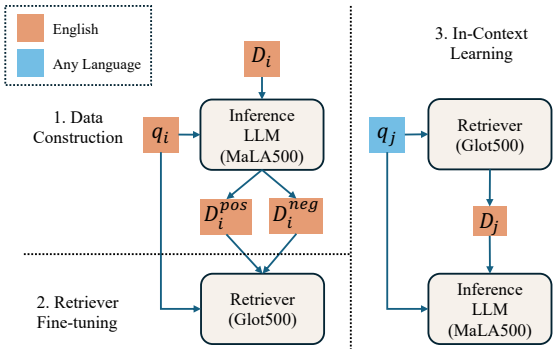


Figure 1: XAMPLER involves three steps: 1. Data Construction: given a query in English q_i, we divide the candidate English examples D_i into positive examples D_i^{pos} and negative examples D_i^{neg} based on the prediction of MaLA500 (Lin et al., 2024b); 2. Retriever Fine-tuning: we fine-tune the retriever based on Glot500 (Imani et al., 2023) using the constructed data; 3. In-Context Learning: given a query in any language q_j, we use the fine-tuned retriever to retrieve relevant English examples D_j as few-shots for in-context learning. **For training, XAMPLER requires English data only. Once trained, the model can be applied to any of the 500 languages covered by MaLA500/Glot500 without any need for (often unavailable) labeled low-resource data.**

SBERT (Reimers and Gurevych, 2020), cover a limited number of languages (i.e., 50+), and language-model-based retrievers (Hu et al., 2020) struggle to effectively align distant languages (Cao et al., 2020; Liu et al., 2023). Additionally, relying on off-the-shelf retrievers might lead to sub-optimal performance. Conversely, adopting task-specific fine-tuned retrievers has been demonstrated as a more effective approach (Rubin et al., 2022). Nonetheless, the availability of data for fine-tuning task-specific retrievers in low-resource languages is limited.

To tackle these challenges, we propose a simple yet effective method that relies solely on annotated English data, termed XAMPLER (Cross-Lingual Example Retrieval). As shown in Fig. 1, given an English query q_i and an English example from the candidate pool D_i, we employ in-context learning with MaLA500 (Lin et al., 2024b), a 10B multilingual LLM covering 534 languages, to predict the label of the query. Based on the correctness of the prediction, we classify the candidate example as either positive or negative, i.e., D_i^{pos} and D_i^{neg}. Then, leveraging the curated dataset, we train a retriever based on Glot500 (Imani et al., 2023), a multilingual small language model covering 534 languages, aiming to minimize the contrastive loss (Rubin et al., 2022; Cheng et al., 2023; Luo et al., 2023). Finally, the trained retriever is directly applied to retrieve valuable few-shot examples in English for the given query in the target language. The retrieved English few-shot examples, along with the input query, are then fed into MaLA500 for in-context learning. Experiments across 176 languages on SIB200 and 16 languages on MasakhaNEWS show that XAMPLER effectively retrieves cross-lingual examples, thereby enhancing in-context learning across languages.

2 Approach

2.1 Problem Definition

Given an input query q_i in any language, our objective is to enhance in-context learning for predicting the label of q_i by retrieving tailored few-shot examples from the pool of candidate examples D. Due to the scarcity of annotated data in low-resource languages, we introduce XAMPLER, namely, Cross-Lingual Example Retrieval. On one hand, we leverage in-domain English examples as the pool of candidate examples D, from which we retrieve cross-lingual examples in English for q_i in any target language. On the other hand, we only consider q_i sourced from English training data to train the task-specific retriever, which is then directly applied for evaluation across languages.

2.2 Data Construction

To train the task-specific retriever aimed at retrieving informative examples for the given query q_i, we consider contrastive learning, which requires both positive and negative examples for each query q_i. We define examples as positive when the LLM accurately predicts the ground truth of q_i while utilizing the example as a one-shot example appended to q_i for in-context learning. Conversely, examples are categorized as negative if the LLM's prediction deviates from the ground truth.

Scoring all pairs of training examples presents a quadratic complexity in $|D|$, making it resource-intensive. Inspired by Rubin et al. (2022), we mitigate this by selecting the top k similar examples as candidates. We utilize Sentence-BERT (SBERT) (Reimers and Gurevych, 2020)[1] for candidate selection. Based on our experiments detailed in Section B, we set $k = 10$. The top k candidates for q_i are denoted as $D_i = \{d_{i,1}, \cdots, d_{i,k}\}$, where each candidate $d_{i,j}$ is represented as $(x_{i,j}, y_{i,j})$, with $x_{i,j}$ being the input and $y_{i,j}$ the corresponding label.

After obtaining the candidate-query pairs $\{(q_i, d_{i,1}), \cdots, (q_i, d_{i,k})\}$, we conduct 1-shot in-context learning with MaLA500 (Lin et al., 2024b) to predict the class of the q_i given the candidate $d_{i,j}$, resulting in a predicted label $\hat{y}_{i,j}$. If MaLA500 correctly predicts the label of q_i (i.e., $\hat{y}_{i,j} = y_i$), we consider the candidate $d_{i,j}$ as a positive example ($d_{i,j}^+$); otherwise a negative example ($d_{i,j}^-$). Finally, we divide D_i into sets of positive and negative examples, denoted as D_i^{pos} and D_i^{neg}, respectively.

2.3 Retriever Fine-tuning

We utilize the contrastive loss (Rubin et al., 2022; Cheng et al., 2023; Luo et al., 2023) to train the task-specific retriever, aiming to maximize the similarity between q_i and $x_{i,j}$ if $x_{i,j}$ is a positive example while minimizing the similarity if $x_{i,j}$ is a negative example. We opt for Glot500 (Imani et al., 2023) with a model size of 395M as the base model for training the retriever, considering the significant cost of fine-tuning an LLM. We train for 50 epochs using the AdamW optimizer with a learning rate of 2e-5 and a batch size of 16. Due to the multilingual nature of Glot500, the fine-tuned retriever

[1]We use version distiluse-base-multilingual-cased-v1.

can be effectively transferred to retrieve in-context examples for other languages.

2.4 In-Context Learning

At test time, when employing in-context learning across languages, where q_i can be in any language, we use the fine-tuned task-specific retriever to retrieve a few cross-lingual examples in English tailored to q_i. The retrieved examples are appended to q_i as input for MaLA500 (Lin et al., 2024b) to predict the label of q_i through in-context learning.

3 Experiment

3.1 Setup

Benchmark We evaluate XAMPLER on two text classification benchmarks: SIB200 (Adelani et al., 2023a) and MasakhaNEWS (Adelani et al., 2023b). The partitioning for these datasets was predefined by the respective benchmarks. SIB200 is a massively multilingual text classification benchmark with seven classes. Our evaluation spans a diverse set of 176 languages, obtained by intersecting the language sets of SIB200 and MaLA500 (see §C). The English training set contains 701 samples, with 204 evaluation samples per language. MasakhaNEWS is a news classification task for 16 African languages, covering six topics. The English training set contains 3.31k samples, with 175 to 948 evaluation samples per language.

Our evaluation framework follows the prompt template used in Lin et al. (2024b): 'The topic of the news [sentence] is [label]', where [sentence] represents the text for classification and [label] is the ground truth. [label] is included when the sample serves as a few-shot example but is omitted when predicting the sample. We opt for English prompt templates over in-language ones due to the labor-intensive nature of crafting templates for non-English languages, especially those with limited resources. MaLA500 takes the concatenation of few-shot examples and q_i as input, then proceeds to estimate the probability distribution across the label set. We measure the performance with accuracy.

Baselines We compare XAMPLER with the following retrieving strategies:

Random Sampling. We randomly select examples from the English candidate pool D.

Multilingual Language Models. We use two massively multilingual language models, Glot500 and MaLA500, as retrievers. Tailored examples are retrieved based on the cosine similarity between the sentence representations of the candidate and the query. For Glot500, we utilize mean pooling over hidden states of the selected layer. For MaLA500, we adopt a position-weighted mean pooling method on the selected layer, assigning higher weights to later tokens (Muennighoff, 2022). We use K-Nearest Neighbors to select the layer that performs best across layers (see §A). The selected layers for Glot500 and MaLA500 are 11 and 21, respectively.

Off-the-shelf Retriever. We also employ three off-the-shelf retrievers trained on parallel data: SBERT (Reimers and Gurevych, 2019), LaBSE (Feng et al., 2022), and Multilingual E5 (Wang et al., 2024).[2]

We set the number of shots as the number of classes and evaluate under two settings: the label-aware setting, where one shot is provided per class, and the label-agnostic setting, where the most similar examples are retrieved regardless of their labels.

	SIB200		MasakhaNEWS	
	Label-Aware	Label-Agnostic	Label-Aware	Label-Agnostic
Random	65.24	61.68	72.32	72.39
Glot500	66.60	68.55	73.35	73.01
MaLA500	66.75	66.25	73.39	71.58
SBERT	67.13	66.59	73.24	72.8
LaBSE	68.51	73.69	72.54	73.29
Multilingual E5	69.09	74.61	73.63	72.61
XAMPLER	**70.18**	**75.91**	**75.02**	**73.85**

Table 1: Average macro-accuracy across the evaluated languages on SIB200 and MasakhaNEWS using XAMPLER compared to the baselines.

3.2 Main Results

The comparison between the baselines and XAMPLER is illustrated in Table 1. Our analysis reveals several insights based on the performance with In-Context Learning (ICL) across different methods. Notably, the random baseline exhibits the worst performance among the baselines using ICL, emphasizing the critical role of example selection for effective in-context learning. Multilingual language models, which retrieve examples based on semantic similarity learned during pretraining, slightly outperform the random baseline. Leveraging an off-the-shelf retriever further improves performance, with Multilingual E5 emerging as the top performer among them. Among all the methods, XAMPLER achieves the highest performance in in-context learning. Specifically, on SIB200, XAMPLER surpasses Multilingual E5 by 1.09% in the label-aware setting and 1.30% in the

[2] https://huggingface.co/intfloat/multilingual-e5-large

label-agnostic setting, and by 1.93% and 1.24% on MasakhaNEWS, respectively.

Our two established practices to reduce resource requirements, i.e., selecting the top-k similar examples as candidates using existing off-the-shelf retrievers and using a smaller base model for retriever training, ensures that XAMPLER operates efficiently. For example, on the SIB200 benchmark (701 samples, 10 candidates per sample) using an NVIDIA GeForce GTX 1080 Ti (11GB), retriever fine-tuning took just 1.5 hours, and retrieving similar examples added only 0.1 seconds per query during inference.

3.3 Ablation Study

To further assess the effectiveness of XAMPLER, we conduct the following ablation studies:

Method	Accuracy (%)
XLT (Glot500)	69.51
XLT (MaLA500)	69.90
MT	74.50
KNN	72.85
XAMPLER	**75.91**

Table 2: Performance comparison of XAMPLER with ablation methods on the SIB200 benchmark.

Cross-lingual Transfer (XLT). Cross-lingual transfer is another approach that utilizes English data. In this method, the multilingual language model is fine-tuned on English data and then evaluated across target languages. Both Glot500 and MaLA500 are included, with their respective cross-lingual transfer methods denoted as XLT (Glot500) and XLT (MaLA500). For Glot500, we perform full-parameter fine-tuning with a batch size of 16 and a learning rate of 1e-5. For MaLA500, which is trained by incorporating LoRA (Hu et al., 2022) into LLaMA 2-7B (Touvron et al., 2023), we update only the LoRA parameters with prompt tuning. The learning rate is set to 1e-3, weight decay is 0.1, the maximum sequence length is 128, and the batch size is 16. The optimizer used is AdamW.

Machine Translation (MT). We translate the examples retrieved by XAMPLER from English to the target language and use these translated examples as few-shot examples for in-context learning. For translation, we use the distilled 600M variant of NLLB-200 (Costa-jussà et al., 2022).

K-Nearest Neighbors (KNN). We consider KNN with the fine-tuned task-specific retriever of XAMPLER for comparison. Specifically, we adopt majority voting based on the labels of the examples retrieved by the given retriever.

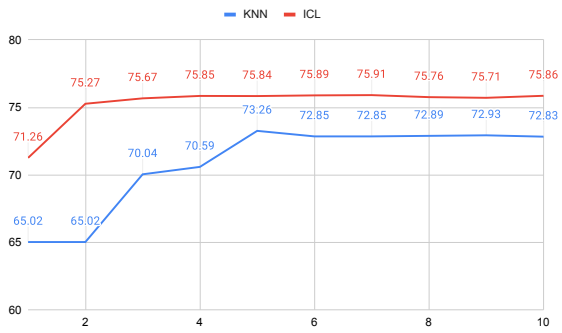


Figure 2: KNN (K-Nearest Neighbors) vs. ICL (In-Context Learning) with different number of shots. X-axis: number of shots. Y-axis: Macro-average accuracy.

The results of the ablation studies are shown in Table 2. XAMPLER outperforms XLT (Glot500) and XLT (MaLA500) by 6.40% and 6.01%, respectively, demonstrating that in-context learning, when combined with informative cross-lingual few-shot examples, is superior to traditional cross-lingual transfer methods via fine-tuning. Translating English examples into target languages results in slightly worse results than XAMPLER by 1.41%. It demonstrates that XAMPLER benefits more from English in-context examples than in-language examples. A comparison between XAMPLER and KNN shows that XAMPLER performs better by 3.06%. We also compared XAMPLER's performance with KNN and ICL using varying numbers of retrieved examples, as illustrated in Figure 2. Interestingly, XAMPLER with ICL exhibits inconsistent superiority over KNN, with performance variances ranging from 3% to 10%. Specifically, XAMPLER with KNN achieves its peak performance with 5 examples, whereas ICL achieves impressive results with only 2 examples. Notably, in comparison to KNN's optimal performance, recorded at 73.26% with 5 shots, XAMPLER with ICL demonstrates a notable improvement of 2.58%. These findings underscore the efficacy of applying in-context learning in effectively leveraging the retrieved examples.

4 Related Work

Early studies (Gao et al., 2021; Liu et al., 2022; Rubin et al., 2022) on retrieving informative examples for few-shot in-context learning often rely on off-the-shelf retrievers to gather semantically similar examples to the query.

While off-the-shelf retrievers have shown promise, the examples they retrieve may not always represent optimal solutions for the given task, potentially resulting in sub-optimal performance. Hence, Rubin et al. (2022) delve into learning-based approaches: if an LLM finds an example useful, the retriever should be encouraged to retrieve it. This approach enables direct training of the retriever using signals derived from query and example pairs in the task of interest.

Several works (Winata et al., 2021; Shi et al., 2022; Winata et al., 2022; Nie et al., 2022; Winata et al., 2023; Tanwar et al., 2023; Cahyawijaya et al., 2024) extend these methods to non-English languages. A study closely related to ours is Shi et al. (2022), which trains a cross-lingual example retriever via distilling the LLM's scoring function and evaluates it on four languages for the Text-to-SQL Semantic Parsing task. However, our contribution lies in addressing the more challenging low-resource scenario, thereby extending the applicability and robustness of the approach proposed by Shi et al. (2022).

5 Conclusion

In this paper, we introduce XAMPLER, a novel approach designed for cross-lingual example retrieval to facilitate in-context learning in any language. Relying solely on English data, XAMPLER trains a task-specific retriever capable of retrieving cross-lingual English examples tailored to any language query, thereby facilitating few-shot in-context learning for any language. Experiments on SIB200 and MasakhaNEWS show that XAMPLER outperforms previous methods by a notable margin.

Limitations

We did not consider other models and benchmarks due to the absence / unavailability of massively multilingual ones. Additionally, while it is acknowledged that English may not universally serve as the optimal source language for cross-lingual transfer across all target languages (Lin et al., 2019; Wang et al., 2023; Lin et al., 2024a), our study does not explore the selection of different source languages due to the predominant availability of training data in English for many tasks.

Acknowledgements

This work was funded by DFG (SCHU 2246/14-1), the European Research Council (DECOLLAGE, ERC-2022-CoG #101088763), EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (Center for Responsible AI), by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research, and by FCT/MECI through national funds and when applicable co-funded EU funds under UID/50008: Instituto de Telecomunicações.

References

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2023a. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. *CoRR*, abs/2309.07445.

David Ifeoluwa Adelani, Marek Masiak, Israel Abebe Azime, Jesujoba O. Alabi, Atnafu Lambebo Tonja, Christine Mwase, Odunayo Ogundepo, Bonaventure F. P. Dossou, Akintunde Oladipo, Doreen Nixdorf, Chris Chinenye Emezue, Sana Sabah Al-Azzawi, Blessing K. Sibanda, Davis David, Lolwethu Ndolela, Jonathan Mukiibi, Tunde Ajayi, Tatiana Moteu Ngoli, Brian Odhiambo, Abraham Toluwase Owodunni, Nnaemeka C. Obiefuna, Muhidin Mohamed, Shamsuddeen Hassan Muhammad, Teshome Mulugeta Ababu, Saheed Abdullahi Salahudeen, Mesay Gemeda Yigezu, Tajuddeen Gwadabe, Idris Abdulmumin, Mahlet Taye Bame, Oluwabusayo Olufunke Awoyomi, Iyanuoluwa Shode, Tolulope Anu Adelani, Habiba Abdulganiyu, Abdul-Hakeem Omotayo, Adetola Adeeko, Afolabi Abeeb, Aremu Anuoluwapo, Samuel Olanrewaju, Clemencia Siro, Wangari Kimotho, Onyekachi Raphael Ogbu, Chinedu E. Mbonu, Chiamaka Chukwuneke, Samuel Fanijo, Jessica Ojo, Oyinkansola Awosan, Tadesse Kebede Guge, Sakayo Toadoum Sari, Pamela Nyatsine, Freedmore Sidume, Oreen Yousuf, Mardiyyah Oduwole, Kanda Patrick Tshinu, Ussen Kimanuka, Thina Diko, Siyanda Nxakama, Sinodos G. Nugussie, Abdulmejid Tuni Johar, Shafie Abdi Mohamed, Fuad Mire Hassan, Moges Ahmed Mehamed, Evrard Ngabire, Jules Jules, Ivan Ssenkungu, and Pontus Stenetorp. 2023b. Masakhanews: News topic classification for african languages. 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, IJCNLP 2023 -Volume 1: Long Papers, Nusa Dua, Bali, November 1 - 4, 2023*, pages 144–159. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda

Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Samuel Cahyawijaya, Holy Lovenia, and Pascale Fung. 2024. Llms are few-shot in-context low-resource language learners. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 405–433. Association for Computational Linguistics.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. UPRISE: universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12318–12337. Association for Computational Linguistics.

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, Loïc 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. *CoRR*, abs/2207.04672.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3816–3830. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André F. T. Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1082–1117. Association for Computational Linguistics.

Peiqin Lin, Chengzhi Hu, Zheyu Zhang, André F. T. Martins, and Hinrich Schütze. 2024a. mplm-sim: Better cross-lingual similarity and transfer in multilingual pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, pages 276–310. Association for Computational Linguistics.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. 2024b. Mala-500: Massive language adaptation of large language models. *CoRR*, abs/2401.13303.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3125–3135. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for*

Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022, pages 100–114. Association for Computational Linguistics.

Yihong Liu, Peiqin Lin, Mingyang Wang, and Hinrich Schütze. 2023. OFA: A framework of initializing unseen subword embeddings for efficient large-scale multilingual continued pretraining. *CoRR*, abs/2311.08849.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Seyed Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y. Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning. *CoRR*, abs/2305.14128.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024. In-context learning with retrieved demonstrations for language models: A survey. *CoRR*, abs/2401.11624.

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *CoRR*, abs/2202.08904.

Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2022. Cross-lingual retrieval augmented prompt for low-resource languages. *CoRR*, abs/2212.09651.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4512–4525. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2655–2671. Association for Computational Linguistics.

Peng Shi, Rui Zhang, He Bai, and Jimmy Lin. 2022. XRICL: cross-lingual retrieval-augmented in-context learning for cross-lingual text-to-sql semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5248–5259. Association for Computational Linguistics.

Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual llms are better cross-lingual in-context learners with alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6292–6307. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 text embeddings: A technical report. *CoRR*, abs/2402.05672.

Mingyang Wang, Heike Adel, Lukas Lange, Jannik Strötgen, and Hinrich Schütze. 2023. NLNDE at semeval-2023 task 12: Adaptive pretraining and source language selection for low-resource multilingual sentiment analysis. *CoRR*, abs/2305.00090.

Genta Indra Winata, Liang-Kang Huang, Soumya Vadlamannati, and Yash Chandarana. 2023. Multilingual few-shot learning via language model retrieval. *CoRR*, abs/2306.10964.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. *CoRR*, abs/2109.07684.

Genta Indra Winata, Shijie Wu, Mayank Kulkarni, Thamar Solorio, and Daniel Preotiuc-Pietro. 2022. Cross-lingual few-shot learning on unseen languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2022 - Volume 1: Long Papers, Online Only, November 20-23, 2022*, pages 777–791. Association for Computational Linguistics.

A KNN Performance Across Layers

We show the 10-shot KNN results across layers with Glot500 and MaLA500 as retrievers in Figure 3 and 4. As shown, layer 21 of MaLA500 and layer 11 of Glot500 achieve the best performance across layers. Therefore, the retrieved results based on these two layers are used in the baselines.

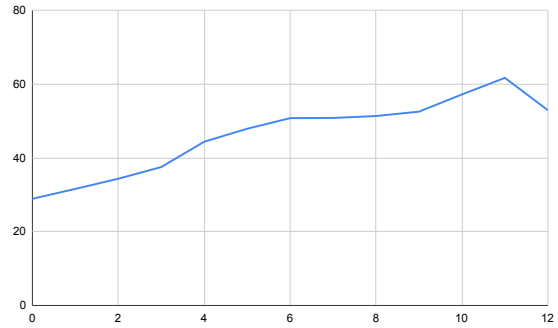


Figure 3: Results of 10-shot KNN (K-Nearest Neighbors) with Glot500 as retriever across layers.

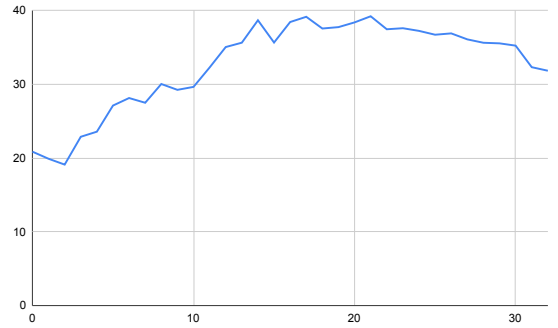


Figure 4: Results of 10-shot KNN (K-Nearest Neighbors) with MaLA500 as retriever across layers.

B Effect of k

We conduct additional experiments to analyze the impact of the parameter k, with the results presented in Figure 5. Our findings indicate that XAMPLER performs optimally when $k = 10$. However, as k exceeds 10, there is a slight decrease in performance. This trend may be attributed to the possibility that increasing k leads to fewer hard negatives for training the retriever.

C Detailed Results

The language list of SIB200 and the results of XAMPLER and the compared baselines are shown in Table 3 and Table 4. The language list of

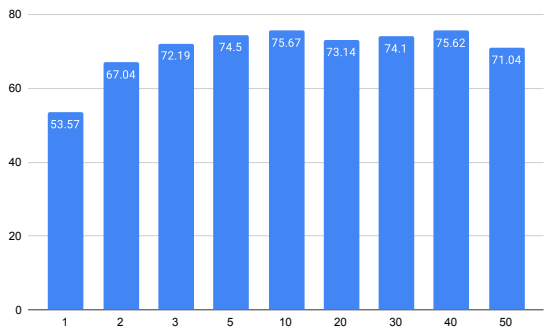


Figure 5: In-context learning with XAMPLER with different k.

MasakhaNEWS and the results of XAMPLER and the compared baselines are shown in Table 5.

	Label-Aware							Label-Agnostic						
	Random	Glot500	MaLA500	SBERT	LaBSE	Multilingual E5	XAMPLER	Random	Glot500	MaLA500	SBERT	LaBSE	Multilingual E5	XAMPLER
ace_Latn	72.55	75.00	72.06	74.51	72.55	74.51	72.06	63.73	71.57	70.10	70.10	76.47	80.39	76.96
acm_Arab	62.25	64.71	65.20	70.10	66.18	68.63	75.00	61.76	71.08	65.69	78.92	75.98	75.98	79.90
afr_Latn	77.45	76.47	77.94	79.90	80.39	81.86	79.41	70.59	79.41	78.43	83.82	88.24	81.86	87.25
ajp_Arab	64.71	67.65	67.16	68.14	69.61	68.14	74.51	64.71	72.06	64.71	77.94	76.47	77.94	84.31
als_Latn	74.51	74.51	77.94	78.92	77.45	79.41	75.00	68.63	75.98	75.00	80.88	83.82	82.84	84.31
amh_Ethi	56.37	59.31	60.29	60.29	61.76	60.78	68.63	57.35	61.27	60.78	60.29	67.16	67.16	71.08
apc_Arab	63.73	67.65	65.20	68.63	68.14	71.08	75.00	61.76	72.06	62.25	78.43	79.90	76.47	85.78
arb_Arab	65.69	69.61	68.63	69.61	72.55	72.06	77.45	65.20	74.02	70.10	76.96	81.37	75.98	83.82
ary_Arab	58.82	65.20	63.24	65.69	64.71	66.18	74.51	57.84	71.57	60.78	71.57	71.57	73.53	80.39
arz_Arab	63.24	66.18	64.71	66.18	67.65	65.20	73.53	62.25	69.61	61.76	76.47	78.92	76.96	82.84
asm_Beng	70.59	72.06	73.04	72.06	75.98	75.49	73.53	69.12	76.47	72.55	64.22	78.92	76.96	83.82
ast_Latn	79.90	76.47	81.37	79.90	79.41	80.39	82.84	74.02	78.43	81.37	85.78	84.31	82.84	89.71
ayr_Latn	39.71	45.59	43.14	46.08	44.61	45.59	42.16	44.12	47.55	47.06	46.08	46.57	53.43	52.45
azb_Arab	47.55	50.49	50.98	50.98	52.45	54.41	64.71	48.04	61.27	49.51	46.57	60.29	61.76	70.59
azj_Latn	77.45	76.47	77.45	77.94	78.92	82.35	77.94	71.57	79.41	76.47	78.92	82.84	83.82	84.80
bak_Cyrl	69.12	75.49	71.57	72.55	72.06	74.51	75.00	66.18	73.53	69.61	77.94	75.00	78.92	80.88
bam_Latn	44.12	46.08	48.04	47.06	49.51	50.49	52.94	43.14	47.55	44.61	46.08	50.00	57.35	49.02
ban_Latn	75.00	74.51	76.47	77.94	78.92	78.43	79.90	69.61	73.53	78.43	75.98	83.33	84.31	82.35
bel_Cyrl	76.96	76.96	75.00	78.43	77.94	78.92	76.47	71.08	77.94	74.02	76.47	84.80	81.86	85.78
bem_Latn	50.49	52.94	52.45	49.51	52.45	54.41	53.43	47.55	58.82	50.98	50.49	54.41	65.69	62.25
ben_Beng	72.06	72.06	67.65	69.61	73.53	72.55	79.41	65.20	75.49	68.14	66.18	77.94	77.94	81.86
bjn_Latn	71.57	73.04	72.55	72.06	73.53	73.53	76.47	71.08	73.04	74.51	70.10	80.39	79.90	83.33
bod_Tibt	53.92	50.49	49.51	50.98	56.86	50.98	50.98	48.53	51.96	51.96	36.76	56.37	50.98	59.31
bos_Latn	78.43	78.43	75.98	78.92	81.37	81.86	79.90	72.06	82.35	77.94	80.39	82.84	82.84	89.22
bul_Cyrl	77.94	78.92	77.45	79.41	78.92	77.94	78.92	72.06	78.92	78.43	81.86	85.78	80.88	85.78
cat_Latn	78.43	77.45	80.88	83.82	80.39	83.33	81.86	74.02	81.86	79.41	84.31	86.76	84.31	88.24
ceb_Latn	75.98	75.98	76.96	76.47	78.92	76.47	79.90	71.08	75.98	76.96	74.51	84.80	81.37	86.27
ces_Latn	75.49	76.96	77.94	76.96	78.43	79.41	77.94	69.61	77.45	76.96	78.92	85.78	82.35	88.73
cjk_Latn	44.12	44.12	44.12	44.12	47.06	46.57	43.14	40.69	46.08	49.02	44.61	50.00	55.88	47.06
ckb_Arab	69.12	71.08	66.67	69.61	69.12	72.06	75.00	63.24	73.53	67.16	66.18	63.24	75.98	81.86
cmn_Hani	76.96	77.45	81.86	79.90	79.41	79.90	86.27	69.12	81.37	77.94	82.84	79.90	80.88	89.71
crh_Latn	71.08	69.12	67.65	72.06	74.02	72.55	72.55	62.25	74.02	69.61	73.04	75.98	75.49	75.00
cym_Latn	77.45	76.96	75.98	75.98	80.39	78.43	77.94	71.57	78.43	79.90	70.59	85.78	84.31	78.92
dan_Latn	81.37	82.84	80.88	82.35	79.90	83.82	84.80	75.49	82.35	82.84	82.35	86.76	82.84	89.71
deu_Latn	80.39	80.88	83.82	83.82	82.35	83.33	83.33	74.51	78.43	84.31	85.78	86.27	85.29	86.76
dyu_Latn	51.96	51.96	50.98	50.98	51.47	56.37	49.02	47.55	49.02	49.51	48.04	54.41	57.84	50.49
dzo_Tibt	31.86	39.22	35.78	33.33	40.69	37.75	50.00	34.31	43.14	37.25	22.06	48.53	34.80	54.90
ell_Grek	74.02	75.00	78.92	77.45	77.45	79.41	76.47	72.06	76.47	74.02	75.98	78.92	80.39	83.82
eng_Latn	82.84	84.31	85.29	85.78	84.31	84.80	86.27	73.04	84.80	85.78	86.27	85.29	87.75	91.67
epo_Latn	73.53	76.96	75.00	74.51	76.96	77.45	79.90	71.08	78.43	77.94	76.96	86.27	84.80	82.84
est_Latn	68.63	73.04	71.08	71.57	74.02	75.49	75.00	67.16	75.98	71.57	72.55	79.90	80.39	79.90
eus_Latn	59.80	70.10	69.12	69.61	69.61	73.04	75.49	63.73	75.49	68.63	69.61	79.90	86.27	83.82
ewe_Latn	42.65	45.59	44.61	47.55	48.04	47.06	48.04	41.67	42.16	43.14	47.55	49.51	54.90	53.43
fao_Latn	62.75	66.18	68.14	64.71	69.61	69.61	73.53	59.31	70.10	63.24	65.69	77.94	77.45	83.82
fij_Latn	43.14	42.65	46.57	45.10	45.59	48.04	50.49	44.12	49.51	47.55	47.55	54.41	60.29	53.43
fin_Latn	75.49	75.00	75.49	74.51	75.98	77.94	77.94	67.16	78.92	74.51	74.51	82.84	80.39	83.33
fon_Latn	43.14	46.57	46.57	46.57	50.98	52.45	42.16	41.18	48.53	42.65	46.08	50.49	57.35	48.04
fra_Latn	78.92	78.43	78.43	80.88	81.37	81.86	83.82	72.06	79.41	81.86	80.39	86.76	82.84	89.22
ful_Latn	45.59	47.06	47.55	50.49	52.94	50.98	43.63	44.61	46.08	50.00	51.96	57.35	55.88	52.94
fur_Latn	69.61	68.63	71.08	69.61	70.59	73.04	75.49	63.73	68.14	70.59	75.98	74.51	80.88	79.41
gla_Latn	63.24	57.84	64.71	65.20	68.63	68.14	62.75	60.29	63.24	68.63	59.80	75.49	74.51	67.16
gle_Latn	65.20	68.14	71.08	71.08	71.08	72.55	66.18	66.18	72.55	67.65	64.71	80.88	81.37	74.02
glg_Latn	79.41	80.88	79.41	79.90	77.94	81.37	84.80	71.57	83.33	81.86	86.76	84.80	87.75	86.76
grn_Latn	63.73	64.22	64.22	67.16	65.69	69.12	70.10	61.76	67.16	66.18	69.61	71.08	75.00	77.94
guj_Gujr	73.53	73.04	70.10	69.61	77.45	73.04	79.90	69.61	73.04	67.65	62.25	77.45	78.92	85.29
hat_Latn	70.10	72.55	75.00	73.53	75.00	76.96	76.47	65.20	77.45	73.53	73.04	82.84	81.37	82.35
hau_Latn	68.14	65.69	67.65	63.73	67.16	68.14	69.12	58.82	68.63	66.67	61.27	76.47	74.02	69.12
heb_Hebr	47.55	50.00	45.10	49.02	52.45	54.41	62.75	47.55	61.76	45.59	50.00	65.69	68.63	77.45
hin_Deva	68.63	71.08	69.61	68.63	73.53	76.47	78.92	69.12	78.43	70.59	63.24	80.39	79.90	85.29
hne_Deva	67.16	74.02	69.12	68.63	75.00	77.45	74.51	63.24	75.00	70.59	62.75	81.37	80.39	80.88
hrv_Latn	79.41	80.88	78.92	77.94	78.92	82.35	80.39	73.04	80.88	76.47	80.39	84.80	82.35	86.76
hun_Latn	76.47	75.00	77.45	77.45	77.45	76.96	80.88	71.08	76.96	76.96	77.94	86.27	78.43	87.25
hye_Armn	74.02	75.49	72.06	74.02	74.51	75.98	76.47	66.67	74.51	71.57	70.59	79.90	80.39	84.80
ibo_Latn	71.08	73.53	72.06	73.53	73.53	74.02	75.00	66.18	71.57	69.61	71.57	79.90	83.33	80.88
ilo_Latn	66.67	69.12	71.08	69.61	73.53	75.49	74.51	62.75	71.57	71.57	74.51	76.47	83.33	78.92
ind_Latn	79.41	81.86	80.88	80.88	81.37	82.35	84.80	74.51	80.88	83.33	83.33	84.80	85.78	90.69
isl_Latn	69.12	69.12	69.61	70.59	71.57	73.04	74.02	65.20	68.63	67.65	69.61	78.92	72.55	81.37
ita_Latn	80.88	79.90	82.35	82.84	83.33	83.33	84.80	73.53	82.35	83.33	86.27	87.25	86.27	90.69
jav_Latn	72.06	74.02	72.55	74.51	75.00	75.49	77.45	69.61	75.98	74.02	76.47	77.45	78.43	85.78
jpn_Jpan	78.43	78.43	80.88	80.88	83.82	81.86	83.33	74.51	77.94	81.86	79.90	86.76	80.88	86.27
kab_Latn	34.31	36.27	37.75	37.75	38.73	39.71	34.31	33.33	38.24	33.33	33.82	40.69	40.20	35.78
kac_Latn	40.20	41.67	41.18	39.71	45.10	44.12	42.65	39.71	40.69	42.65	47.06	47.06	54.90	46.08
kam_Latn	41.18	43.63	40.69	44.12	47.55	47.06	44.12	40.69	50.98	44.61	44.61	51.47	55.39	49.51
kan_Knda	69.61	72.06	66.18	69.61	71.08	73.53	74.02	65.20	74.51	68.63	64.71	76.96	77.45	77.45
kat_Geor	76.47	75.49	77.45	74.51	77.45	77.45	78.92	74.02	79.41	76.96	64.71	83.82	78.92	83.82
kaz_Cyrl	72.55	73.04	73.04	73.53	75.00	75.49	77.45	64.22	75.49	70.59	73.53	80.88	79.41	82.35
kbp_Latn	43.14	44.12	48.04	48.04	47.55	48.53	49.51	43.63	42.16	46.57	47.55	44.61	50.98	47.55
kea_Latn	74.51	75.98	75.00	77.94	75.00	76.96	79.41	68.14	77.45	75.49	82.84	81.37	82.35	78.92
khm_Khmr	76.96	76.47	75.49	74.51	76.47	77.94	80.88	71.57	77.45	74.51	75.00	78.92	79.41	85.29
kik_Latn	50.00	53.92	55.39	52.94	54.90	55.39	58.82	48.04	54.90	52.94	53.43	57.35	64.71	58.82
kin_Latn	49.51	51.96	51.96	50.49	51.47	51.96	56.37	49.02	55.39	50.49	48.04	64.71	72.06	62.75
kir_Cyrl	68.14	69.61	69.61	70.59	72.06	74.51	72.06	64.22	73.04	64.71	69.61	75.98	77.94	78.43
kmb_Latn	40.20	41.67	42.16	43.14	44.61	46.08	43.63	43.63	47.06	44.12	43.14	48.53	53.43	50.98
kmr_Latn	58.33	61.76	61.76	61.27	65.20	66.18	67.16	55.88	69.61	61.27	57.84	74.02	76.96	69.61
kon_Latn	58.33	64.71	61.27	63.24	65.69	67.16	63.24	56.37	61.76	59.80	61.76	59.31	74.51	64.22
kor_Hang	76.96	78.92	80.88	83.82	81.37	81.86	81.37	75.98	80.39	81.86	87.25	86.27	85.29	88.73
lao_Laoo	72.55	70.59	74.51	72.06	73.04	70.59	75.98	66.67	73.53	74.02	70.59	76.47	78.43	84.31
lij_Latn	73.04	73.04	72.06	74.02	74.51	75.49	75.98	67.16	72.55	73.53	78.43	76.96	78.92	80.39
lim_Latn	77.45	76.47	75.98	79.41	77.94	77.45	75.98	68.14	78.92	76.96	81.86	83.82	82.35	78.43
lin_Latn	59.31	57.84	60.78	63.24	60.29	59.31	62.75	52.45	63.24	60.78	65.20	61.27	67.65	66.67

Table 3: 7-shot accuracy with on SIB200.

Bibliography

Ife Adebara, AbdelRahim A. Elmadany, and Muhammad Abdul-Mageed. 2024. Cheetah: Natural language generation for 517 african languages. *CoRR*, abs/2401.01053.

Ife Adebara, AbdelRahim A. Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2022. SERENGETI: massively multilingual language models for africa. *CoRR*, abs/2212.10785.

David Ifeoluwa Adelani, Jesujoba O. Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Hassan Muhammad, Guyo Dub Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Wambui Gitau, Jade Z. Abbott, Mohamed Ahmed, Millicent Ochieng, Aremu Anuoluwapo, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for african news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3053–3070. Association for Computational Linguistics.

Kabir Ahuja, Rishav Hada, Millicent Ochieng, Prachi Jain, Harshita Diddee, Samuel Maina, Tanuja Ganu, Sameer Segal, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: multilingual evaluation of generative AI. *CoRR*, abs/2303.12528.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022a. Adapting pre-trained language models to african languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 4336–4349. International Committee on Computational Linguistics.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022b. Multilingual language model adaptive fine-tuning: A study on african languages. *CoRR*, abs/2204.06487.

Duarte M. Alves, Nuno Miguel Guerreiro, João Alves, José Pombal, Ricardo Rei, José Guilherme Camargo de Souza, Pierre Colombo, and André F. T. Martins. 2023. Steering large language models for machine translation with finetuning and in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 11127–11148. Association for Computational Linguistics.

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro Henrique Martins, João Alves, M. Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *CoRR*, abs/2402.17733.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. 2023a. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. 2023b. Palm 2 technical report. *CoRR*, abs/2305.10403.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Un-

supervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *CoRR*, abs/2309.16609.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 610–623. ACM.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Samuel Cahyawijaya, Holy Lovenia, Tiezheng Yu, Willy Chung, and Pascale Fung. 2023. Instruct-align: Teaching novel languages with to llms through alignment-based cross-lingual instruction. *CoRR*, abs/2305.13627.

Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Leylia Khodra, Ayu Purwarianti, and Pascale Fung. 2021. Indonlg: Benchmark and resources for evaluating indonesian natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8875–8898. Association for Computational Linguistics.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2023. When is multilinguality a curse? language modeling for 250 high- and low-resource languages. *CoRR*, abs/2311.09205.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. 2022. The geometry of multilingual language model representations. *CoRR*, abs/2205.10964.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual bert, a small treebank, and a small corpus. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1324–1334. Association for Computational Linguistics.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Ifeoluwa Adelani, Pontus Stenetorp, Sebastian Riedel, and Mikel Artetxe. 2023. Improving language plasticity via pretraining with active forgetting. *CoRR*, abs/2307.01163.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 5564–5577. Association for Computational Linguistics.

Rochelle Choenni and Ekaterina Shutova. 2022. Investigating language relationships in multilingual sentence encoders through the lens of linguistic typology. *Comput. Linguistics*, 48(3):635–672.

Rochelle Choenni, Ekaterina Shutova, and Dan Garrette. 2023. Examining modularity in multilingual lms via language-specialized subnetworks. *CoRR*, abs/2311.08273.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *CoRR*, abs/2304.08177.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

C. M. Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. 2023. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. *CoRR*, abs/2309.04679.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4555–4567. Association for Computational Linguistics.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2023. Do multilingual language models think better in english? *CoRR*, abs/2308.01223.

Fahim Faisal and Antonios Anastasopoulos. 2022. Phylogeny-inspired adaptation of multilingual models to new languages. *CoRR*, abs/2205.09634.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Daniela Gerz, Ivan Vulic, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. 2018. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 316–327. Association for Computational Linguistics.

Kevin Heffernan, Onur Çelebi, and Holger Schwenk. 2022. Bitext mining using distilled sentence representations for low-resource languages. *CoRR*, abs/2205.12654.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne 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. *CoRR*, abs/2305.07004.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André F. T. Martins,

François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1082–1117. Association for Computational Linguistics.

Shaoxiong Ji, Zihao Li, Indraneil Paul, Jaakko Paavola, Peiqin Lin, Pinzhen Chen, Dayyán O'Brien, Hengyu Luo, Hinrich Schütze, Jörg Tiedemann, et al. 2024. Emma-500: Enhancing massively multilingual adaptation of large language models. *arXiv preprint arXiv:2409.17892*.

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

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, ACL 2020, Online, July 5-10, 2020*, pages 6282–6293. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N. C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4948–4961. 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. *CoRR*, abs/2001.08361.

Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. 2021. Alignment of language agents. *CoRR*, abs/2103.14659.

Fajri Koto, Afshin Rahimi, Jey Han Lau, and Timothy Baldwin. 2020. Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian NLP. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December*

8-13, 2020, pages 757–770. International Committee on Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2023. Align after pre-train: Improving multilingual generative models with cross-lingual alignment. *CoRR*, abs/2311.08089.

Sheng Liang, Philipp Dufter, and Hinrich Schütze. 2021. Locating language-specific information in contextualized embeddings. *CoRR*, abs/2109.08040.

Jindrich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1663–1674. Association for Computational Linguistics.

Peiqin Lin, Chengzhi Hu, Zheyu Zhang, André F. T. Martins, and Hinrich Schütze. 2024a. mplm-sim: Better cross-lingual similarity and transfer in multilingual pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, March 17-22, 2024*, pages 276–310. Association for Computational Linguistics.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. 2024b. Mala-500: Massive language adaptation of large language models. *CoRR*, abs/2401.13303.

Peiqin Lin, André F. T. Martins, and Hinrich Schütze. 2025a. A recipe of parallel corpora exploitation for multilingual large language models. *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Peiqin Lin, André F. T. Martins, and Hinrich Schütze. 2025b. XAMPLER: learning to retrieve cross-lingual in-context examples. *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, pages 100–114. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

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. *CoRR*, abs/1907.11692.

Hongyuan Lu, Haoyang Huang, Dongdong Zhang, Haoran Yang, Wai Lam, and Furu Wei. 2023a. Chain-of-dictionary prompting elicits translation in large language models. *CoRR*, abs/2305.06575.

Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023b. Are emergent abilities in large language models just in-context learning? *CoRR*, abs/2309.01809.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno Miguel Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, M. Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2024. Eurollm: Multilingual language models for europe. *CoRR*, abs/2409.16235.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Benjamin Müller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021a. When being unseen from mbert is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 448–462. Association for Computational Linguistics.

Benjamin Müller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021b. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2214–2231. Association for Computational Linguistics.

Benjamin Müller, Benoît Sagot, and Djamé Seddah. 2020. Can multilingual language models transfer to an unseen dialect? A case study on north african arabizi. *CoRR*, abs/2005.00318.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021, Online, April 19-23, 2021*, pages 80–90. Association for Computational Linguistics.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. Kinyabert: a morphology-aware kinyarwanda language model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5347–5363. Association for Computational Linguistics.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual*

Representation Learning, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav Chaudhary, and Xia Song. 2022. Beyond english-centric bitexts for better multilingual language representation learning. *CoRR*, abs/2210.14867.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: an adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7654–7673. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2021. Unks everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10186–10203. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. 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 M. 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 J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/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. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sara Rajaee and Mohammad Taher Pilehvar. 2022. An isotropy analysis in the multilingual BERT embedding space. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1309–1316. Association for Computational Linguistics.

Taraka Rama, Lisa Beinborn, and Steffen Eger. 2020. Probing multilingual BERT for genetic and typological signals. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 1214–1228. International Committee on Computational Linguistics.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev,

Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2655–2671. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulic, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *J. Artif. Intell. Res.*, 65:569–631.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *CoRR*, abs/2210.03057.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *CoRR*, abs/2204.07580.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zheng, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *CoRR*, abs/2201.11990.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *CoRR*, abs/2104.09864.

Yutao Sun, Li Dong, Barun Patra, Shuming Ma, Shaohan Huang, Alon Benhaim, Vishrav Chaudhary, Xia Song, and Furu Wei. 2023. A length-extrapolatable transformer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 14590–14604. Association for Computational Linguistics.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2024. A benchmark for learning to translate a new language from one grammar book. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin

Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ahmet Üstün, Viraat Aryabumi, Zheng Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *CoRR*, abs/2402.07827.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. Udapter: Language adaptation for truly universal dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2302–2315. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022a. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 863–877. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Gary Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Maitreya Patel, Kuntal Kumar Pal, Mehrad

Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Shailaja Keyur Sampat, Savan Doshi, Siddhartha Mishra, Sujan Reddy, Sumanta Patro, Tanay Dixit, Xudong Shen, Chitta Baral, Yejin Choi, Hannaneh Hajishirzi, Noah A. Smith, and Daniel Khashabi. 2022b. Benchmarking generalization via in-context instructions on 1, 600+ language tasks. *CoRR*, abs/2204.07705.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020a. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2649–2656. Association for Computational Linguistics.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020b. On negative interference in multilingual models: Findings and A meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4438–4450. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 833–844. Association for Computational Linguistics.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1006–1011. The Association for Computational Linguistics.

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, NAACL-HLT 2021, Online, June 6-11, 2021, pages 483–498. Association for Computational Linguistics.

Ziqing Yang, Zihang Xu, Yiming Cui, Baoxin Wang, Min Lin, Dayong Wu, and Zhigang Chen. 2022. CINO: A chinese minority pre-trained language model. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 3937–3949. International Committee on Computational Linguistics.

Zheng Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M. Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Indra Winata, Stella Biderman, Dragomir Radev, and Vassilina Nikoulina. 2022. BLOOM+1: adding language support to BLOOM for zero-shot prompting. *CoRR*, abs/2212.09535.

Chen Zhang, Xiao Liu, Jiuheng Lin, and Yansong Feng. 2024a. Teaching large language models an unseen language on the fly. *CoRR*, abs/2402.19167.

Kexun Zhang, Yee Man Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024b. Hire a linguist!: Learning endangered languages with in-context linguistic descriptions. *CoRR*, abs/2402.18025.

Zhihan Zhang, Dong-Ho Lee, Yuwei Fang, Wenhao Yu, Mengzhao Jia, Meng Jiang, and Francesco Barbieri. 2023. PLUG: leveraging pivot language in cross-lingual instruction tuning. *CoRR*, abs/2311.08711.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer. *CoRR*, abs/2401.01055.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating large language models to non-english by aligning languages. *CoRR*, abs/2308.04948.