COLING 2025

SUMEval-2
The 2nd Workshop on Scaling Up Multilingual & Multi-Cultural Evaluation

Proceedings of the Workshop

January 20, 2025

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-952148-25-5

Preface

Massively Multilingual Language Models (MMLMs) like mBERT, XLMR and XY-LENT support around 100 languages of the world. Additionally, generative models like GPT-4 and BLOOM have shown impressive performance in English and a few high-resource languages. However, most existing multilingual NLP benchmarks reflect a handful of cultures and languages. The languages present in evaluation benchmarks are usually high-resource and largely belong to the Indo-European language family. By extension, the cultures represented in evaluation benchmarks are also largely reflective of Western society. This makes current evaluation unreliable and does not provide a full picture of the performance of MMLMs across the linguistic and cultural landscape. Although efforts are being made to create benchmarks that cover a larger variety of tasks, cultures, languages, and language families, it is clear that scaling-up multilingual and multi-cultural evaluation that can eventually lead to better models for all languages and cultures remains a formidable research challenge. This workshop is the second workshop in the SUMEval series, following a successful first workshop SUMEval 2022 co-located with AACL 2022. This year's workshop SUMEval-2, co-located with COLING 2025, has a wider scope focusing on multicultural evaluation in addition to multilingual evaluation.

Organizing Committee

Hellina Hailu Nigatu, UC Berkeley

Monojit Choudhury, MBZUAI

Oana Ignat, Santa Clara University

Sebastian Ruder, Meta

Sunayana Sitaram, Microsoft

Vishrav Chaudhary, Meta

Table of Contents

Tentative Conference Program

0900 - 0915 Inaugural Remarks

0915 - 1015 Invited Talk by Minsu Park: Cross-Cultural Mood Dynamics and Detection Algorithms

1015 - 1100 Coffee break

1100 - 1200 Panel Discussion 1: Challenges of Collecting Culturally Grounded Multilingual Data for Training and Evaluation of NLP Systems

1200 - 1230 2-min Pitch for Papers (both archival and non-archival papers)

1230 - 1400 Lunch

1400 - 1530 Poster Session (both archival and non-archival)

1530 - 1600 Coffee break

1600 - 1715 Panel Discussion 2: Diverse Cultures. Diverse Problems. Diverse Solutions. Understanding the Nuanced Challenges and Opportunities in Working with Diverse Cultures

1715 - 1730 Closing and Awards

The First Multilingual Model For The Detection of Suicide Texts

Rodolfo Zevallos
Barcelona Supercomputing Center
rodolfo.zevallos@bsc.es

Annika Schoene
Northeastern University
a.schoene@northeastern.edu

John E. Ortega
Northeastern University
j.ortega@northeastern.edu

Abstract

Suicidal ideation is a serious health problem affecting millions of people worldwide. Social networks provide information about these mental health problems through users' emotional expressions. We propose a multilingual model leveraging transformer architectures like mBERT, XML-R, and mT5 to detect suicidal text across posts in six languages - *Spanish*, *English*, *German*, *Catalan*, *Portuguese* and *Italian*. A Spanish suicide ideation tweet dataset was translated into five other languages using SeamlessM4T. Each model was fine-tuned on this multilingual data and evaluated across classification metrics. Results showed mT5 achieving the best performance overall with F1 scores above 85%, highlighting capabilities for cross-lingual transfer learning. The English and Spanish translations also displayed high quality based on perplexity. Our exploration underscores the importance of considering linguistic diversity in developing automated multilingual tools to identify suicidal risk. Limitations exist around semantic fidelity in translations and ethical implications which provide guidance for future human-in-the-loop evaluations.

1 Introduction

According to data published by the World Health Organization (WHO), over 700,000 people die by suicide each year (Organization et al., 2021), with an additional 10 to 20 million attempting to take their own lives. Suicidal behavior typically begins with thoughts and ideations of death, eventually leading to suicide attempts - conscious acts with the purpose of ending one's existence (Liu and Miller, 2014). In this context, social networks have become spaces where individuals often disclose emotions and information that they don't feel comfortable sharing with healthcare providers (Ji et al., 2020; Desmet and Hoste, 2013; Sueki, 2015).

Early identification of signs of suicidal ideation in these online environments poses a major challenge. This is where Natural Language Processing (NLP) and Deep Learning (DL) can play a crucial role in the automatic detection of suicidal thoughts in computational settings. Furthermore, these computational approaches may contribute to the development of tools for harm reduction and prevention. However, the majority of research on suicidal ideation detection has been conducted on *English* language data, resulting in a scarcity of linguistic resources (e.g.: datasets, lexicons, and Language Models) for most other languages.

Previous computational approaches to identifying suicidal ideation have relied heavily on hand-engineered features and domain expertise. For example, some studies have used structural and emotional features to train statistical prediction models on suicide text (Jones and Bennell, 2007; Pestian et al., 2012). Additionally, conventional machine learning algorithms like Logistic Regression (LR) (Ramírez-Cifuentes et al., 2020; Jain et al., 2019; Schoene and Dethlefs, 2016; O'dea et al., 2015), Decision Tree (DT) (Jain et al., 2019; Huang et al., 2015), Naive Bayes (NB) (Shah et al., 2020; Rabani et al., 2020; Chiroma et al., 2018; Schoene and Dethlefs, 2016), Support Vector Machine (SVM) (Renjith et al., 2022; Shah et al., 2020; Ramírez-Cifuentes et al., 2020), K-nearest neighbor algorithm (KNN) (Shah et al., 2020; Vioules et al., 2018) and Extreme Gradient Boost (XGBoost) (Rajesh Kumar et al., 2020; Jain et al., 2019; Ji et al., 2018) have been applied. While achieving some success, these methods depend on costly feature engineering and professional knowledge.

Recently, deep learning has emerged as a promising approach that can automatically learn representations from data (Goldberg, 2022). Moreover, deep learning techniques like CNNs (Yao et al., 2020; Renjith et al., 2022; Tadesse et al., 2019), LSTMs (Haque et al., 2022; Tadesse et al., 2019; Renjith et al., 2022; Ji et al., 2018; Ma et al., 2018), BiLSTM (Haque et al., 2022; Zhang et al., 2022;

Proceedings of the 2nd SUMEval Workshop: Scaling Up Multilingual & Multi-Cultural Evaluation, pages 1–11
January 20, 2025.

He and Lin, 2016) and DLSTMAttention (Zhang et al., 2022; Renjith et al., 2022) have been applied in detecting suicidal ideation, with competitive performance.

On the other hand, with the increasing use of pre-trained language models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and mT5 (Xue et al., 2020), the landscape of suicide ideation detection has evolved significantly. These pre-trained models, developed through massive unsupervised learning on diverse linguistic tasks, offer a powerful foundation for understanding intricate nuances of language. Researchers are now exploring the adaptation of those models for the detection of suicidal ideation (Bhaumik et al., 2023; Devika et al., 2023; Haque et al., 2020). Leveraging the contextual understanding encoded in these pre-trained models, studies have reported promising results in discerning subtle and complex expressions related to suicidal thoughts, contributing to the advancement of automated detection systems (Bhaumik et al., 2023). This shift towards pre-trained language models signifies a paradigmatic enhancement in the field, as it allows for a more nuanced comprehension of linguistic patterns associated with suicidal ideation, thereby enhancing the overall accuracy and sensitivity of detection algorithms.

In our approach, we emphasize the importance of addressing the detection of suicidal texts in the context of using multilingual language models. Translating a corpus from *Spanish* into five different languages and fine-tuning a multilingual language model allows us to classify suicidal texts in various languages, thus expanding the applicability of our approach.

In our research, we aim to explore the effectiveness of multilingual pretrained language models such as mBERT (Devlin et al., 2019), XML-R (Liu et al., 2019) and mT5 (Xue et al., 2020) for detecting suicidal text on social media. The main focus of our study is to leverage these multilingual pretrained models to translate and recognize suicidal text in six languages: *Spanish*, *English*, *German*, *Catalan*, *Portuguese* and *Italian*.

Our primary contribution lies in the implementation of a multilingual language model with the capability to detect suicidal text in these six distinct languages. Additionally, to address the lack of labeled suicidal text in other languages, we utilize a labeled corpus and translate it into five different languages using an automatic translation model (SeamlessM4T(Barrault et al., 2023)). Therefore, this approach enables us to effectively tackle the linguistic diversity present on social media and provides a valuable tool for the early identification of suicidal content in various cultural and linguistic contexts.

The main objectives of our study are:

1. **Prediction of Suicidal Text in Six Languages:** The model focuses on predicting posts with suicidal content by analyzing the words or phrases written by users, utilizing multilingual pretrained language models such as mBERT, XML-R and mT5.

2. **Improvement of Prediction Accuracy in Various Languages:** We aim to enhance the accuracy of predicting suicidal text by incorporating attention mechanisms from multilingual pretrained language models. These attention mechanisms highlight crucial aspects within the obtained information, providing effective detection in six different languages.

The significant contributions of our work include:

1. **Detection of Suicidal Texts Using Multilingual Pretrained Language Models (mBERT, XML-R, mT5):** We propose a model that integrates multilingual pretrained language models, including mBERT, XML-R amd mT5, for effective detection of suicidal texts in social media posts in six different languages.

2. **Prediction of User-Specific Suicidal Tendencies in Various Languages:** The model examines the posts of specific users to determine if they exhibit suicidal tendencies, leveraging the capabilities of the mentioned multilingual pretrained language models.

2 Related Work

The initial approaches to automatic suicide risk detection were based on identifying specific language features present in psychiatric literature. For instance, in Lumontod III (2020); Tadesse et al. (2019), the LIWC dictionary was used to extract emotional and cognitive markers, while Masuda et al. (2013) designed a set of emotional features such as feelings of loneliness, helplessness, and hopelessness. Additionally, Pestian et al. (2010)

employed suicide notes to identify common language themes and styles.

However, the limitations of manual feature engineering in terms of scalability and adaptability have led to the exploration of more recent approaches based on deep learning, specifically pre-trained language models. In a comparative study (Tavchioski et al., 2023; Sawhney et al., 2018), BERT, RoBERTa, BERTweet, and mentalBERT were evaluated on a Reddit dataset, revealing that pre-trained models consistently outperformed traditional classifiers (Valeriano et al., 2020; Maalouf, 2011; Aladağ et al., 2018).

In summary, pre-trained language models have shown promising results, often outperforming traditional methods in automatic suicide risk detection. However, most studies have been limited to relatively small datasets. Regarding linguistic diversity, studies have predominantly focused on English data from platforms such as Twitter (Kabir et al., 2023; Coppersmith et al., 2015), Reddit (Tavchioski et al., 2023; Losada and Crestani, 2016; Losada et al., 2017), and Facebook. Nevertheless, no research has been found exploring multilingual language models for suicide risk detection. Models like mBERT (Devlin et al., 2019), XLM-RoBERTa (Liu et al., 2019), and mT5 (Xue et al., 2020), trained on multilingual data, could transfer linguistic knowledge across related languages, improving performance in low-resource situations for languages with less training data.

As far as is known, there are also no studies utilizing automatically translated datasets to leverage data from other languages. The quality of automatic translations of datasets from a source language to a target language could be crucial in increasing dataset size and improving the performance of trained models.

Both research directions, i.e., multilingual models and automatic translation of data, represent promising yet unexplored areas for automatic suicide risk detection, opening opportunities for significant contributions in this field.

3 Experiments

In this section, we delineate the setup of diverse experiments aimed at exploring the feasibility of a multilingual model capable of classifying suicidal texts across six different languages.

3.1 Dataset

The dataset we utilized in our experiments is the set of 2,068 Spanish tweets introduced in Valeriano et al. (2020). This dataset was compiled by the authors through targeted keyword searches on expressions of suicidal ideation. The tweets were then manually annotated by humans, labeling each as either containing suicidal intent or not – a binary classification scheme. After annotation, the dataset contains 498 tweets (24%) expressing suicidal ideas, with example phrases like "I want to disappear" or "I can't stand life anymore." The remaining 1,570 Spanish tweets do not express suicide risk.

We split the full dataset into training, validation. 80% of the data, encompassing 1,654 Spanish tweets, was used for model training to learn signals of suicidal intent. The validation set makes up 20% of the data, with 414 tweets, which was leveraged during model development for hyperparameter tuning and performance checks. Moreover, we used as test set the Lexicography Saves Lives (LSL) Schoene et al. (2025).

We leveraged this dataset by machine translating the entire corpus of 2,068 Spanish tweets into five other languages: Catalan, English, German, Italian, and Portuguese. The translations were produced using Facebook's SeamlessM4T model (Barrault et al., 2023), allowing us to obtain versions of the suicide texts dataset across multiple languages stemming from the original Spanish source data (Valeriano et al., 2020).

3.2 Pre-trained language models

The recent advances in neural network-based language models have demonstrated substantial improvements across a wide range of natural language processing tasks (Goldberg, 2022). In particular, the introduction of Transformer architectures (Vaswani et al., 2017) led to unprecedented progress in semantic and syntactic modeling capabilities. Unlike previous recurrent models such as LSTMs (Hochreiter and Schmidhuber, 1997), Transformer networks apply a purely attention-based mechanism to learn intricate context representations. By utilizing multiple attention heads in parallel, these architectures can capture both local and global dependencies in a sequence of tokens.

The original authors of the Transformer introduced a specific implementation called BERT (Devlin et al., 2019), which laid the groundwork for a

new generation of contextualized language models. Through pre-training objectives such as predicting subsequent sentences and token masking, BERT achieves a deep syntactic and semantic understanding of language. However, the initial version of BERT was limited to the English language. Subsequent research focused on extending these models to a multilingual context to enable cross-lingual learning.

Adaptations such as mBERT[1] (Devlin et al., 2019) emerged, incorporating shared vocabularies and subword segmentation to represent a wide range of languages. Then, XML-R[2] (Liu et al., 2019) enhanced the multifaceted approach by adding byte-level tokenization and techniques like Whole-Word Masking. Finally, mT5[3] (Xue et al., 2020) adopted an encoder-decoder architecture instead of the exclusively encoder format. Considering the rapid progress in multilingual language models, this work aimed to evaluate three transformative alternatives for the automatic detection of suicidal ideation: mBERT, XML-R, and mT5. Through thorough experimentation, the goal is to determine their capabilities in both language and semantics.

Each chosen model presents unique characteristics, as described earlier, that could positively impact their performance for the given task. Additionally, all of them were pretrained in various languages, incorporating millions of trainable parameters and state-of-the-art techniques to enhance cross-linguistic transfer. In combination, this diversity allows addressing the problem from multiple perspectives, enabling a comprehensive evaluation of the relative advantages of different cutting-edge approaches for such a sensitive scenario as the expression of suicidal intentions.

For this study, three pre-trained language models were utilized and we oultine below further details about the architecture, hyperparameters, and training datasets for each.

3.3 Suicide phrase recognition

In the pursuit of robust multilingual performance, our experiments enlisted the capabilities of four cutting-edge language models: mBERT (Devlin et al., 2019), XML-R (Liu et al., 2019) and mT5 (Xue et al., 2020). To fortify their adaptability, each model underwent a meticulous fine-tuning process. Leveraging the Spanish dataset, as previously detailed, and its translations into six languages—Catalan, English, German, Italian, and Portuguese—we aimed to comprehensively capture the nuances of suicidal text across linguistic variations.

Parameter	**mBERT**	**XML-R**	**mT5**
Starting learning rate	2e-5	3e-5	3e-5
Batch size	16	16	32
Epochs	10	10	10
Dropout	0.3	0.5	0.5
Weight decay	0.01	0.01	0.01
Optimizer	AdamW	AdamW	AdamW

Table 1: Hyperparameters for models fine-tuning

The initial configurations for fine-tuning were aligned with the recommended settings provided by each language model. Subsequently, recognizing the intricate interplay of hyperparameters in influencing model performance, we conducted an exhaustive search to identify the most effective and contextually relevant hyperparameter sets for each individual model. This process was undertaken with a dual purpose: ensuring optimal performance across languages and tailoring the models to the specific intricacies of suicidal text classification.

We fine-tune mBERT, XML-R and mT5 on 1 NVIDIA 4070 GPUs with FP32. Model hyperparameters are tuned on the validation set, where learning rate {2e-5, 3e-5, 3e-5}, batch size {16, 16, 32}, a dropout rate of {0.3, 0.5, 0.5}, a weight decay of 0.01, a warmup proportion of 0.01. For clarity and replicability, the detailed configurations for all models, including the identified hyperparameter sets, are meticulously documented in Table 1.

4 Results

In this section, we present an analysis of the results obtained from our fine-tuned language models—mBERT, XML-R and mT5 deployed in the task of suicidal text classification across six languages. Our objective is to scrutinize the models' performance intricacies, assess their multilingual adaptability, and glean insights into the efficacy of

[1] https://github.com/google-research/bert/blob/master/multilingual.md

[2] https://huggingface.co/xlm-roberta-base

[3] https://github.com/google-research/multilingual-t5

Lang.	mBERT			XML-R			mT5		
	Acc.	F1.	AUC	Acc.	F1.	AUC	Acc.	F1.	AUC
Spanish	82.4	82.1	82.2	84.6	84.3	84.3	87.9	87.7	87.8
English	83.6	83.3	83.1	85.6	85.5	85.4	88.5	88.1	88.1
Italian	78.7	78.5	78.4	80.6	80.6	80.4	83.3	83.2	83.1
German	81.1	80.9	80.7	82.9	82.9	82.8	86.2	86.1	86.0
Catalan	81.3	81.1	81.0	82.8	82.7	82.6	86.2	86.1	86.0
Portuguese	79.9	79.9	79.8	81.7	81.7	81.5	84.9	84.8	84.8

Table 2: Experimental results with mBERT, XML-R, and mT5 across different languages. Notation: Acc. = accuracy and F1. = F1-score.

our approach.

4.1 Classifiers Performance Analysis

We delve into the nuanced evaluation of our language models' performance across six languages: Spanish, Catalan, English, German, Italian, and Portuguese. Table 2 shows that, mT5 displays superior performance over the other two models across all metrics and for all languages. The precision, recall, F1, and AUC scores are consistently high, surpassing 85% in most cases.

This indicates that mT5 is exceptionally good at both positively detecting relevant cases (high recall) as well as minimizing false positives (high precision). It also maintains an adequate balance between both goals, as shown by its high F1-score. There is clearly substantial superiority of mT5 at this task compared to more generic BERT models.

On the other hand, we see that mBERT obtains the lowest scores, although still decent (around 80-83% for key metrics). XML-R improves upon mBERT's results across all languages, suggesting that language-specific pretraining can be beneficial.

Regarding languages, English and Spanish consistently achieve the top scores across all models, followed by German and Catalan. Italian and Portuguese appear to be the most difficult. This could be due to several factors: data availability, similarity to English, etc.

An interesting finding is that the relative gaps between models remain remarkably stable across languages. This implies that the inherent strengths of each model transcend linguistic particularities. While some languages are more complex, all benefit from mT5's architectural improvements over BERT models.

In summary, mT5 is better suited to suicide text detection, especially excelling for English and Spanish. mBERT may perform adequately as a baseline, but there is clear room for improvement with more advanced models such as XML-R and especially mT5.

4.2 Model Validation

To delve deeper into understanding the learning mechanism, we implemented k-fold cross-validation to determine the mean accuracy in our three models: mBERT, XML-R and mT5. Cross-validation is a widely used data resampling strategy to assess the generalization capabilities of predictive models and estimate the true estimation error. In k-fold cross-validation, the learning set is divided into k subgroups of approximately equal length, and the number of subgroups produced is referred to as 'fold'. This partition is achieved by randomly selecting examples from the learning set without replacement. Our language models, including mBERT, XML-R and mT5, were fine-tuned using k = 10 subsets representing the entire training set. Each model was then applied to the remaining subset, known as the validation set, and its performance was evaluated. This process was repeated until all k subsets had served as validation sets.

Subsequently, we proceeded to conduct additional tests in our six languages since our models are multilingual. This variant involves applying our models in scenarios with various languages, adding an additional level of complexity and versatility to the evaluation of their performance in detecting suicidal text. Figure 1 illustrates the F1-score of each of our three language models for each fold in the cross-validation, highlighting their adaptability to diverse subsets of data and linguistic scenarios. This meticulous approach ensures robust training and optimal performance for each of our models in the detection of suicidal text, considering both linguistic diversity and the specific characteristics of mBERT, XML-R and mT5 in this particular context.

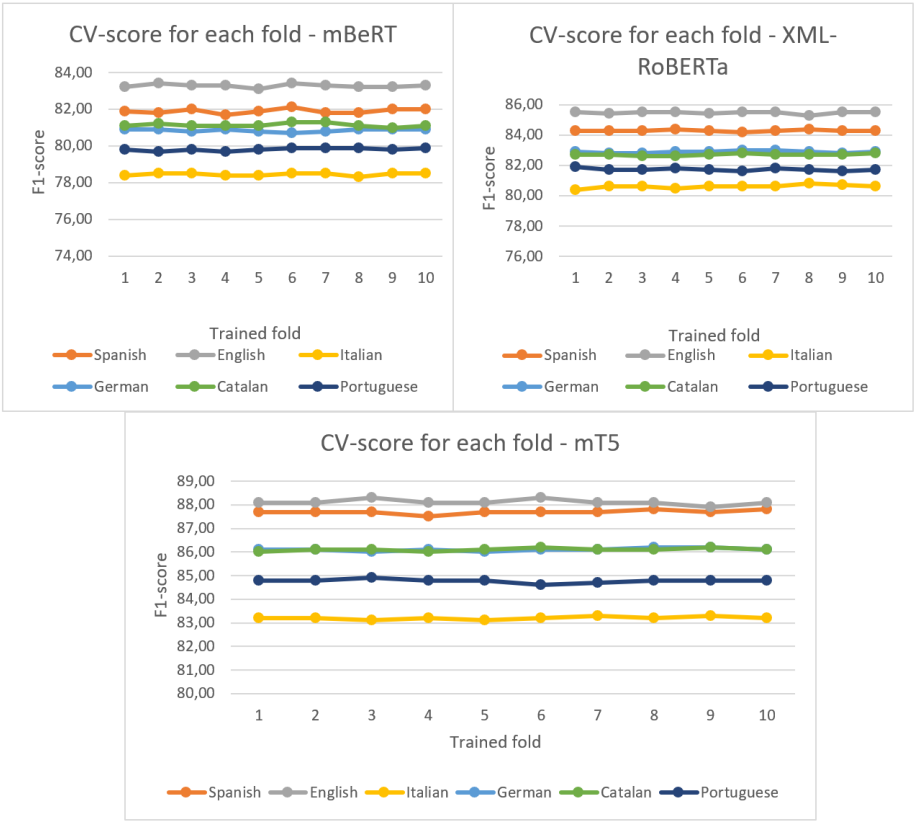


Figure 1: 10-fold Cross-Validation for each language model

5 Translation analysis

For the translation of the Spanish dataset into the other 5 target languages (English, Catalan, German, Italian and Portuguese), this study employed the SeamlessM4T[4] model developed by Facebook (Barrault et al., 2023).

SeamlessM4T is based on the Transformer architecture, demonstrating the effectiveness of cross-lingual model pretraining and transfer learning. In particular, it leverages a sequence-to-sequence model with encoder-decoder structure trained on large-scale data across multiple languages (100 languages).

The key advantages of this specific architecture include:

- Attention-based interactions model both global and local dependencies in input and output sequences. This provides greater context and reduces reliance on recurrence.
- Multi-head self-attention combines representations from different positional offsets, learning synergistic features.
- Masked language modeling and denoising objectives during pretraining further enhance context modeling.

It was pretrained on a variety of language pairs, including Spanish, English, Catalan, Italian and Portuguese. It demonstrated excellent BLEU metrics on translations between these languages, corroborating its suitability for the present cross-lingual research task.

5.1 Evaluation Metrics

Perplexity serves as an indicator to quantify the quality of each translation. We employed monolingual language models specific to each target language, assessing their ability to predict word sequences in the translated texts.

5.2 Results

The perplexity scores for each translation are presented in the Table 3:

[4] https://github.com/facebookresearch/seamless_communication

Translation to Language	Perplexity Score
English[1]	3.43
Catalan[2]	5.65
German[2]	8.18
Italian[2]	7.75
Portuguese[2]	4.61

Table 3: Perplexity scores for each translation.

With a low perplexity of 3.43, the English translation demonstrates notable coherence and fluency, suggesting successful adaptation from the Spanish source. This indicates that the model competently encoded the linguistic intricacies in mapping between the closely-related languages.

The Catalan translation perplexity of 5.65 signifies adequate synchronization from Spanish, though with slightly heightened linguistic complexity. This points to competent cross-lingual transfer learning while highlighting some incremental challenges for the more distant language pair.

However, the German translation perplexity of 8.18 underlines particular difficulties in adaptation, as substantiated by amplified linguistic complexity. As Spanish and German topologically diverge, this outcome spotlights the obstacles for conversion across more disparate languages.

Italian translation returned a perplexity of 7.75, underscoring reasonably effective adaptation despite lower fluency compared to other counterparts. This demonstrates capable inter-lingual transfer learning for the language pair, albeit with some decline in conversion quality.

Finally, the Portuguese translation perplexity of 4.61 reflects adept transformation from Spanish, mirroring the performance benchmark set by English. The proximity between Spanish and Portuguese facilitates smooth cross-lingual mapping, resulting in harmonized coherence.

6 Discussion

This study explores the use of pre-trained multilingual language models, including mBERT, XML-R, and mT5, for the automatic detection of suicidal texts in social media posts across six languages: Spanish, English, German, Catalan, Portuguese, and Italian. The results show mT5 achieving the best performance overall, with F1 scores above 85%, highlighting capabilities for cross-lingual transfer learning.

An interesting finding is that the relative gaps between models remain remarkably stable across languages. This implies that the inherent strengths of each model transcend linguistic particularities. While some languages are more complex, all benefit from mT5's architectural improvements over BERT models.

Regarding limitations, direct extrapolation of the results to other languages must be approached cautiously, given the wide linguistic diversity and potential impact of cultural nuances on interpreting suicidal texts. Furthermore, the quality of translations and, consequently, the predictive model, is inherently tied to the effectiveness of pre-trained models, indicating a constant need for improvements in this area.

While this study presents a promising model for multilingual detection of suicidal texts, there are several directions to extend and strengthen this line of research. Some of these include: expanding linguistic scope by incorporating a broader spectrum of languages; enriching training data with more instances and diversity of sources; using specialized metrics to quantify the usefulness of the early detection model; and implementation of a user-friendly interface enabling integration into healthcare settings.

In summary, the focus on multilingual translation emerges as a crucial step in constructing an effective predictive model for suicidal texts across six languages. The identified conclusions and limitations provide guidance for future developments, emphasizing the need for linguistic and cultural considerations.

7 Conclusion

In the pursuit of a predictive model for suicidal texts in six languages, our exploration into multilingual translation yields critical insights. We observe that translations into *English* and *Portuguese* excel, showcasing the ability to preserve intent and coherence in sensitive contexts such as suicidal content.

Sensitivity to linguistic diversity emerges as a pivotal element in this process. While synchronization in translations into Catalan was acceptable, adaptations into German and Italian posed challenges, underscoring the importance of considering linguistic nuances in constructing a robust predictive model. The versatility of multilingual models,

[1] https://huggingface.co/roberta-large

[2] https://huggingface.co/facebook/xlm-roberta-xl

especially mT5, proves to be a valuable resource in this scenario. These models demonstrate a remarkable ability to maintain the integrity of suicidal content across diverse languages, providing a solid foundation for building a multilingual predictive model. Automated evaluation, though guided by objective metrics such as perplexity, does not replace human assessment for sensitivity and semantic fidelity in suicidal content. The implementation of human evaluations in subsequent phases is essential to ensure the appropriateness and ethical considerations of the model.

In summary, our focus on multilingual translation emerges as a crucial step in constructing a predictive model for suicidal texts in six languages. The identified conclusions and limitations provide guidance for future developments, emphasizing the need for linguistic and cultural considerations, as well as continuous improvements in pre-trained models and human evaluations to achieve an effective and ethical model.

8 Ethical Considerations

There are a number of aspects to consider when using pretrained language models to automatically translate suicide related language, especially given the sensitive nature of the data. Firstly, we have to consider user privacy and be aware of the impact online surveillance, collection of sensitive data and people's health. Furthermore, there are concerns around linguistic, cultural and contextual accuracy when automatically translating suicide-related tweets, where there can be issues around accurate translations and misrepresentation of cultural or conceptual concepts. Finally,

9 Limitations and Future Work

Direct extrapolation of our results to other languages must be approached cautiously, given the wide linguistic diversity and the potential impact of cultural nuances on the interpretation of suicidal texts. Furthermore, the quality of translations and, consequently, the predictive model, is inherently linked to the effectiveness of pre-trained models, indicating a constant need for improvements in this area.

While this study presents a promising model for multilingual detection of suicidal ideation, there are several directions to extend and strengthen this line of research:

- **Expansion of Linguistic Scope** Incorporating a broader spectrum of languages would be key to achieving a globally impactful tool. Languages with limited use of digital technologies like *Hindi*, *Arabic* or *Chinese* pose challenges due to scarce representation in training data. Techniques such as small-scale automatic translation of annotated data and adaptation of models to new languages through transfer learning could help bridge this gap.

- **Enrichment of Training Data** Having more instances and diversity of sources in the initial Spanish dataset would enhance derived models. Collecting content from platforms like Reddit (Zirikly et al., 2019) and Facebook (Ophir et al., 2020) with a higher prevalence of mental health themes could be beneficial. Expanding labels to capture emotional nuances, linguistic subtleties and a more granular view of suicide-realted content (e.g.: moving beyond binary classification) could also contribute.

- **Specialized Metrics** To more precisely quantify the utility of the early detection model, metrics like average latency to high-risk posts or rate of early false negatives should be incorporated. Establishing how these indicators vary across dialectal and sociocultural differences is essential.

- **Implementation for Healthcare Institutions** Developing a user-friendly interface for models that enables integration in healthcare settings would ease the transition of this technology into real-world applications. Achieving integration with existing clinical record systems and care workflows could further its adoption.

Addressing these extensions would provide a comprehensive system with superior accuracy, broad multilingual reach and significant impact on the timely detection and prevention of suicidal behaviors through computing.

References

Ahmet Emre Aladağ, Serra Muderrisoglu, Naz Berfu Akbas, Oguzhan Zahmacioglu, and Haluk O Bingol. 2018. Detecting suicidal ideation on forums: proof-of-concept study. *Journal of medical Internet research*, 20(6):e9840.

Loïc Barrault, Yu-An Chung, Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, et al. 2023. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*.

Runa Bhaumik, Vineet Srivastava, Arash Jalali, Shanta Ghosh, and Ranganathan Chandrasekaran. 2023. Mindwatch: A smart cloud-based ai solution for suicide ideation detection leveraging large language models. *medRxiv*, pages 2023–09.

Fatima Chiroma, Han Liu, and Mihaela Cocea. 2018. Suiciderelated text classification with prism algorithm. In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 2, pages 575–580. IEEE.

Glen Coppersmith, Mark Dredze, Craig Harman, Kristy Hollingshead, and Margaret Mitchell. 2015. Clpsych 2015 shared task: Depression and ptsd on twitter. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality*, pages 31–39.

Bart Desmet and VéRonique Hoste. 2013. Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351–6358.

SP Devika, MR Pooja, MS Arpitha, and Ravi Vinayakumar. 2023. Bert-based approach for suicide and depression identification. In *Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022*, pages 435–444. Springer.

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.

Yoav Goldberg. 2022. *Neural network methods for natural language processing*. Springer Nature.

Farsheed Haque, Ragib Un Nur, Shaeekh Al Jahan, Zarar Mahmud, and Faisal Muhammad Shah. 2020. A transformer based approach to detect suicidal ideation using pre-trained language models. In *2020 23rd international conference on computer and information technology (ICCIT)*, pages 1–5. IEEE.

Rezaul Haque, Naimul Islam, Maidul Islam, and Md Manjurul Ahsan. 2022. A comparative analysis on suicidal ideation detection using nlp, machine, and deep learning. *Technologies*, 10(3):57.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pages 937–948.

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

Xiaolei Huang, Xin Li, Lei Zhang, Tianli Liu, David Chiu, and Tingshao Zhu. 2015. Topic model for identifying suicidal ideation in chinese microblog. In *Proceedings of the 29th pacific asia conference on language, information and computation*, pages 553–562. Waseda University.

Swati Jain, Suraj Prakash Narayan, Rupesh Kumar Dewang, Utkarsh Bhartiya, Nalini Meena, and Varun Kumar. 2019. A machine learning based depression analysis and suicidal ideation detection system using questionnaires and twitter. In *2019 IEEE students conference on engineering and systems (SCES)*, pages 1–6. IEEE.

Shaoxiong Ji, Shirui Pan, Xue Li, Erik Cambria, Guodong Long, and Zi Huang. 2020. Suicidal ideation detection: A review of machine learning methods and applications. *IEEE Transactions on Computational Social Systems*, 8(1):214–226.

Shaoxiong Ji, Celina Ping Yu, Sai-fu Fung, Shirui Pan, and Guodong Long. 2018. Supervised learning for suicidal ideation detection in online user content. *Complexity*, 2018.

Natalie J Jones and Craig Bennell. 2007. The development and validation of statistical prediction rules for discriminating between genuine and simulated suicide notes. *Archives of Suicide Research*, 11(2):219–233.

Mohsinul Kabir, Tasnim Ahmed, Md Bakhtiar Hasan, Md Tahmid Rahman Laskar, Tarun Kumar Joarder, Hasan Mahmud, and Kamrul Hasan. 2023. Deptweet: A typology for social media texts to detect depression severities. *Computers in Human Behavior*, 139:107503.

Richard T Liu and Ivan Miller. 2014. Life events and suicidal ideation and behavior: A systematic review. *Clinical psychology review*, 34(3):181–192.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

David E Losada and Fabio Crestani. 2016. A test collection for research on depression and language use. In *International conference of the cross-language evaluation forum for European languages*, pages 28–39. Springer.

David E Losada, Fabio Crestani, and Javier Parapar. 2017. Clef 2017 erisk overview: Early risk prediction on the internet: Experimental foundations. *CLEF (Working Notes)*, 850.

Robinson Z Lumontod III. 2020. Seeing the invisible: Extracting signs of depression and suicidal ideation from college students' writing using liwc a computerized text analysis. *Int. J. Res. Stud. Educ*, 9:31–44.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Maher Maalouf. 2011. Logistic regression in data analysis: an overview. *International Journal of Data Analysis Techniques and Strategies*, 3(3):281–299.

Naoki Masuda, Issei Kurahashi, and Hiroko Onari. 2013. Suicide ideation of individuals in online social networks. *PloS one*, 8(4):e62262.

Bridianne O'dea, Stephen Wan, Philip J Batterham, Alison L Calear, Cecile Paris, and Helen Christensen. 2015. Detecting suicidality on twitter. *Internet Interventions*, 2(2):183–188.

Yaakov Ophir, Refael Tikochinski, Christa SC Asterhan, Itay Sisso, and Roi Reichart. 2020. Deep neural networks detect suicide risk from textual facebook posts. *Scientific reports*, 10(1):16685.

World Health Organization et al. 2021. Suicide worldwide in 2019: global health estimates.

John Pestian, Henry Nasrallah, Pawel Matykiewicz, Aurora Bennett, and Antoon Leenaars. 2010. Suicide note classification using natural language processing: A content analysis. *Biomedical informatics insights*, 3:BII–S4706.

John P Pestian, Pawel Matykiewicz, and Michelle Linn-Gust. 2012. What's in a note: construction of a suicide note corpus. *Biomedical informatics insights*, 5:BII–S10213.

Syed Tanzeel Rabani, Qamar Rayees Khan, and Akib Mohi Ud Din Khanday. 2020. Detection of suicidal ideation on twitter using machine learning & ensemble approaches. *Baghdad science journal*, 17(4):1328–1328.

E Rajesh Kumar, KVSN Rama Rao, Soumya Ranjan Nayak, and Ramesh Chandra. 2020. Suicidal ideation prediction in twitter data using machine learning techniques. *Journal of Interdisciplinary Mathematics*, 23(1):117–125.

Diana Ramírez-Cifuentes, Ana Freire, Ricardo Baeza-Yates, Joaquim Puntí, Pilar Medina-Bravo, Diego Alejandro Velazquez, Josep Maria Gonfaus, and Jordi Gonzàlez. 2020. Detection of suicidal ideation on social media: multimodal, relational, and behavioral analysis. *Journal of medical internet research*, 22(7):e17758.

Shini Renjith, Annie Abraham, Surya B Jyothi, Lekshmi Chandran, and Jincy Thomson. 2022. An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms. *Journal of King Saud University-Computer and Information Sciences*, 34(10):9564–9575.

Ramit Sawhney, Prachi Manchanda, Puneet Mathur, Rajiv Shah, and Raj Singh. 2018. Exploring and learning suicidal ideation connotations on social media with deep learning. In *Proceedings of the 9th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 167–175.

Annika Schoene, John E. Ortega, Rodolfo Joel Zevallos, and Laura Ihle. 2025. Lexicography saves lives (lsl): Automatically translating suicide-related language. In *Proceedings of the 31st International Conference on Computational Linguistics*. International Committee on Computational Linguistics.

Annika Marie Schoene and Nina Dethlefs. 2016. Automatic identification of suicide notes from linguistic and sentiment features. In *Proceedings of the 10th SIGHUM workshop on language technology for cultural heritage, social sciences, and humanities*, pages 128–133.

Faisal Muhammad Shah, Farsheed Haque, Ragib Un Nur, Shaeekh Al Jahan, and Zarar Mamud. 2020. A hybridized feature extraction approach to suicidal ideation detection from social media post. In *2020 IEEE Region 10 Symposium (TENSYMP)*, pages 985–988. IEEE.

Hajime Sueki. 2015. The association of suicide-related twitter use with suicidal behaviour: a cross-sectional study of young internet users in japan. *Journal of affective disorders*, 170:155–160.

Michael M Tadesse, Hongfei Lin, Bo Xu, and Liang Yang. 2019. Detection of depression-related posts in reddit social media forum. *Ieee Access*, 7:44883–44893.

Ilija Tavchioski, Marko Robnik-Šikonja, and Senja Pollak. 2023. Detection of depression on social networks using transformers and ensembles. *arXiv preprint arXiv:2305.05325*.

Kid Valeriano, Alexia Condori-Larico, and Josè Sulla-Torres. 2020. Detection of suicidal intent in spanish language social networks using machine learning. *International Journal of Advanced Computer Science and Applications*, 11(4).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

M Johnson Vioules, Bilel Moulahi, Jérôme Azé, and Sandra Bringay. 2018. Detection of suicide-related posts in twitter data streams. *IBM Journal of Research and Development*, 62(1):7–1.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Hannah Yao, Sina Rashidian, Xinyu Dong, Hongyi Duanmu, Richard N Rosenthal, and Fusheng Wang. 2020. Detection of suicidality among opioid users on reddit: machine learning–based approach. *Journal of medical internet research*, 22(11):e15293.

Tianlin Zhang, Annika M Schoene, and Sophia Ananiadou. 2022. Automatic identification of suicide notes with a transformer-based deep learning model elsevier, vol. 25.

Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. 2019. CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 24–33, Minneapolis, Minnesota. Association for Computational Linguistics.

CrossIn: An Efficient Instruction Tuning Approach for Cross-Lingual Knowledge Alignment

Geyu Lin[♡], **Bin Wang**[♡], **Zhengyuan Liu**[♡], **Nancy F. Chen**[♡,†]
[♡]Institute for Infocomm Research (I^{2}R), A*STAR, Singapore
[†]Centre for Frontier AI Research (CFAR), A*STAR, Singapore
{lin_geyu, wang_bin,liu_zhengyuan,nfychen}@i2r.a-star.edu.sg

Abstract

Multilingual proficiency presents a significant challenge for large language models (LLMs). English-centric models are usually suboptimal in other languages, particularly those that are linguistically distant from English. This performance discrepancy mainly stems from the imbalanced distribution of training data across languages during pre-training and instruction tuning stages. To address this problem, we propose a novel approach called `CrossIn`, which utilizes a mixed composition of cross-lingual instruction tuning data. Our method leverages the compressed representation shared by various languages to efficiently enhance the model's task-solving capabilities and multilingual proficiency within a single process. In addition, we introduce a multi-task and multi-faceted benchmark to evaluate the effectiveness of `CrossIn`. Experimental results demonstrate that our method substantially improves performance across tasks and languages, and we provide extensive insights into the impact of cross-lingual data volume and the integration of translation data on enhancing multilingual consistency and accuracy.[1]

1 Introduction

The advancement of large language models (LLMs) like ChatGPT (Achiam et al., 2023) and Gemma (Team et al., 2023) has been a game-changer in the field of natural language processing (NLP), revolutionizing tasks such as language generation and commonsense reasoning (Naveed et al., 2024). Nevertheless, most state-of-the-art LLMs are English-centric, and their performance on non-English languages is usually suboptimal, especially on languages that are dissimilar to English (Blevins and Zettlemoyer, 2022; Mehrabi et al., 2022; Gao et al., 2024). This challenge mainly stems from the imbalanced distribution of multilingual data at both the pre-training and instruction tuning stages. The exposure bias toward major languages results in an imbalanced capability, where models excel in languages with plentiful data while under-performing in those with limited resources (Dac Lai et al., 2023; Feng et al., 2023). Bridging the language gap is a fundamental step to unlock the full potential of these general-purpose models and ensure that the benefits are accessible to people across the linguistic spectrum (Zhu et al., 2023a).

Efforts to improve the multilingual capabilities of English-centric LLMs have included continue pre-training using extensive language-specific datasets. Yet, mastering languages through additional pre-training could require vast amounts of data and significant computational resources (Workshop et al., 2022). On the other hand, despite the limited proportion of non-English data at the pre-training stage, their absolute volume builds a solid knowledge base of various languages. In each iteration, LLMs are exposed to samples in several languages simultaneously, and the compressed representation encourages models to share linguistic features and generalize across different languages (Workshop et al., 2022). However, this ability is not fully retained through the use of datasets that only include English in follow-up tuning steps.

In this work, we propose an efficient approach based on a mixed composition of cross-lingual instruction tuning data to exploit LLMs' underlying multilingual capability, which particularly improves the cross-lingual knowledge alignment (Qi et al., 2023; Wang et al., 2023). Instruction tuning is to boost the task solving capability of pre-trained language backbones (Taori et al., 2023; Luo et al., 2023; Touvron et al., 2023b). The task and prompt diversity are crucial in both data preparation and the training process, and a small high-quality set is sufficient to achieve state-of-the-art zero-shot performance (Ouyang et al., 2022). However, the language diversity of instruction tuning

[1]Our datasets and models will be released after the anonymity period.

Proceedings of the 2nd SUMEval Workshop: Scaling Up Multilingual & Multi-Cultural Evaluation, pages 12–23
January 20, 2025.

is often overlooked in English-centric LLMs. We thus aim to enrich instruction tuning from the language perspective. Since all languages share the compressed representation space, cross-lingual instruction tuning can efficiently boost the model's task-solving and multilingual capabilities within a single process. Unlike previous work that involved a multi-task setting by adding machine translation and mixing monolingual samples of each language (Zhu et al., 2023b), we integrate two languages at the sample level and combine various languages at the corpus level. Moreover, we compare various mixing strategies to identify the impact of different data formulations.

To extensively evaluate the cross-lingual knowledge alignment (Qi et al., 2023; Wang et al., 2023), we establish a benchmark of three tasks (i.e., reading comprehension, commonsense question-answering, and logic reasoning). Consistency is measured by analyzing an LLM's responses to the same question in different languages, and our benchmark encompasses multiple ability aspects and difficulty levels. Moreover, since exact match and F1 score cannot precisely evaluate system outputs in the generative setting, we unify all three tasks in a multiple-choice format for quantitative and reproducible evaluation. The experimental results demonstrate that our mixed cross-lingual tuning can significantly improve performance in all aspects (up to 40% relative gain), followed by a detailed analysis of the influence of data quantity on language consistency and knowledge accuracy.

The main contributions of our research are:

- **A Multi-faceted Benchmark.** We present a multi-lingual, multi-capability benchmark for assessing the cross-lingual knowledge consistency of language models. In particular, we build a parallel multiple-choice version of the XQuAD dataset (Artetxe et al., 2019) - Cross-XQuAD for machine comprehension, and combining it with commonsense QA and logic reasoning.

- **Mixed Cross-Lingual Instruction Tuning.** We introduce `CrossIn`, a cross-lingual instruction tuning approach aimed at aligning knowledge across languages to stimulate the model's full multilingual capability after pretraining. It offers a more efficient way of improving the model's performance in various linguistic contexts.

- **`CrossIn` Data Insights.** We conduct extensive experiments with representative LLMs on three tasks, and show the effectiveness of our proposed approach. We provide detailed analysis to study the optimal amount of cross-lingual data and the necessity of sample translation in enhancing models' cross-lingual consistency.

2 Related Work

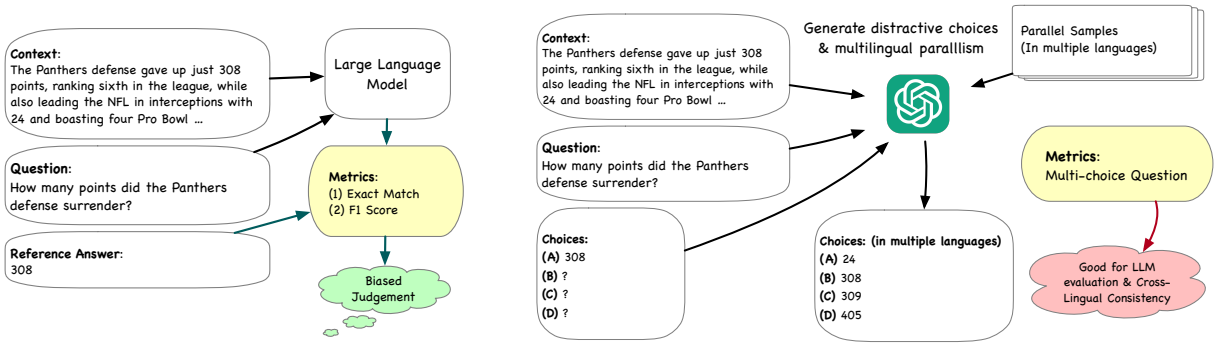
2.1 Multilingual Large Language Model

Multilingual Large Language Models (MLLMs) have experienced significant advancements in recent years. Recently, Qin et al. (2024), as a comprehensive review, summarizes various methodologies for training MLLMs. BLOOM (Workshop et al., 2022), Jais (Sengupta et al., 2023), and Sailor (Dou et al., 2024) are representative models that target improved multilingualism in the pretraining stage. For fine-tuning, ChatGLM employs a reward model trained under a multilingual setting (Zeng et al., 2022), while the x-LLM utilizes a translated version of the Alpaca dataset, combined with supervised translation data and instruction finetuning, to enhance the model's multilingual capabilities (Zhu et al., 2023b).

Instruction tuning on English datasets can introduce zero-shot capabilities in other languages as well (Wei et al., 2022; Chung et al., 2022). Further studies have explored the use of diverse training sets in multiple languages can improve cross-lingual generalization, suggesting that incorporating data from various languages can significantly enhance the model's ability to generalize across linguistic boundaries (Muennighoff et al., 2023; Kew et al., 2023; Shaham et al., 2024). In our work, we build upon these findings and focus on improving multilingual consistency through targeted instruction finetuning. By refining the instruction processing mechanism, we aim to enforce the alignment across different languages to improve multilingual capabilities.

2.2 Multilingual Evaluation Benchmark

Evaluating the multilingual capabilities of LLMs is crucial for their global applicability, as it ensures that these models can understand and generate text effectively across different languages. Benchmarks such as MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2021) have been developed to access the general capability of the



(a) Original XQuAD Dataset

(b) Cross-XQuAD Dataset Creation

Figure 1: An illustration of the dataset construction process of the Cross-XQuAD dataset. The original XQuAD dataset, although multilingual, is not adapted specifically to evaluate LLMs and their cross-lingual consistency.

LLMs in English. XQuAD (Artetxe et al., 2019) and MLQA (Lewis et al., 2019) are popular extractive question-answering datasets that have been developed to evaluate the models' multilingual performance. However, they focus on language-specific performance without considering the knowledge-sharing capabilities. Recently, Cross-MMLU and Cross-LogiQA (Wang et al., 2023) are proposed to assess the multilingual capability of LLMs with an emphasis on cross-lingual consistency. However, the number of samples is limited which could generally lead to less stable evaluation results.

3 Cross-Lingual Consistency Benchmark

Since traditional multilingual evaluations often fail to cater specifically to LLMs or overlook the assessment of cross-lingual consistency in multilingual contexts, in this section, we present a targeted multilingual evaluation benchmark for cross-lingual knowledge alignment.

3.1 Datasets and Metrics

Even though there are multilingual evaluation datasets with parallel samples including MLQA (Lewis et al., 2019) and XQuAD (Artetxe et al., 2019), they are tailored for supervised extractive question-answering tasks and are unsuitable for less structured outputs of LLMs (Schuster et al., 2023). Therefore, recently, two evaluation datasets have been developed for multilingual evaluation with cross-lingual consistency measures (Wang et al., 2023). Specifically, Cross-MMLU and Cross-LogiQA are designed to use multiple-choice questions, presenting parallel samples to assess the knowledge alignment capability of LLMs. These datasets focus on commonsense question answering and logical reasoning. However, as they are crafted by humans, the number of parallel samples they offer is relatively limited due to the high cost of human labor involved. This limitation could lead to less robust evaluation results.

Considering this, in our work, we enhance the cross-lingual consistency evaluation benchmark by introducing another task type: reading comprehension. Furthermore, we utilize existing high-quality parallel datasets to automatically generate new ones that are tailored for LLM evaluation. Table 1 summarizes the complete benchmark.

For evaluation metrics, we leverage the same concept as presented in Wang et al. (2023). In addition to assessing the overall accuracy of each language, we also integrate cross-lingual consistency metrics, measured by "Consistency" and "AC3". The consistency score is designed to determine whether the model provides consistent responses to parallel questions across different languages. A higher consistency score suggests that LLMs can apply common knowledge across languages and deliver uniform responses, regardless of correctness. Specifically, for the Cross-XQuAD dataset that spans four languages, the multilingual consistency metric is defined as

$$M_{\{l_1,l_2,...,l_s\}} = \frac{\sum_{i=1}^{N} 1\{a_i^{l_1} = a_i^{l_2} = ... = a_i^{l_s}\}}{N} \quad (1)$$

where $a_i^{l_s}$ is the answer for sample index i from language s. Then, the consistency is computed as:

$$Consistency_s = \frac{\sum_{\{l_1,l_2,...,l_s\} \in C(s,g_i)} M_{\{l_1,l_2,...,l_s\}}}{C_4^s} \quad (2)$$

Similar to Wang et al. (2023), we use $s = 3$ as the default tolerant for consistency metrics, where

Dataset	MCQs	Number of Samples	Supported Language	Consistency Metric
MLQA (Lewis et al., 2019)	✗	5,500 (36×)	7 - Eng, Zho, Spa, Vie, ...	NA
XQuAD (Artetxe et al., 2019)	✗	1,190 (7.9×)	10 - Eng, Zho, Spa, Vie, ...	NA
Cross-MMLU (Wang et al., 2023)	✓	150 (1×)	7 - Eng, Zho, Spa, Vie, ...	✓
Cross-LogiQA (Wang et al., 2023)	✓	176 (1.2×)	7 - Eng, Zho, Spa, Vie, ...	✓
Cross-XQuAD (ours)	✓	1,190 (7.9×)	4 - Eng, Zho, Spa, Vie	✓

Table 1: A list of multilingual datasets. Multi-choice questions (MCQs) are more suitable for quantitative evaluation of large language models and evaluation for multilingual consistency. Traditional metrics such as the F1 score or Exact Match for extractive question answering can introduce unintended biases in evaluating large language models.

the consistency between any three languages is computed. AC3 enhances the traditional accuracy metric by incorporating consistency, offering a more comprehensive evaluation. This approach is adopted because relying solely on consistency or accuracy does not yield a robust assessment.

$$AC3_s = 2 \cdot \frac{Accuracy \cdot Consistency_s}{Accuracy + Consistency_s} \quad (3)$$

By converting the datasets into MCQ (Multiple Choice Question) format, we can better quantify the model's ability to select the correct answer from a set of options, thereby offering a clearer measure of its understanding and reasoning capabilities.

3.2 Cross-XQuAD Construction

Figure 1 indicates the process of constructing the Cross-XQuAD dataset from the original XQuAD dataset. It involves three steps, 1) English MCQ construction with distractive choices, 2) Parallel MCQ construction, and 3) Post-processing and quality check.

First, the original ground-truth answer from the XQuAD dataset can directly be used as the correction choice. As the XQuAD is for an extractive question-answer task, we extract the incorrect options from the provided context corpus as much as possible. Otherwise, the solution would be highly trivial with simple matching techniques. To achieve this, we prompt *ChatGPT-3.5* to get the other three choices as shown in Figure 1b.

Second, using the prepared English sample as a base, we prompt the generation of equivalent samples in the other languages. We discovered that direct translation without specific context can result in deviated interpretations due to polysemy, potentially leading to a biased evaluation. To counter this, we prompt the model with the English sample alongside its contextual counterpart in the target language to generate new samples. This approach has resulted in samples that are highly aligned across multiple languages.

Third, although LLMs can perform as a reasonable automated method for creating parallel samples (Li et al., 2023), we found that human intervention is essential to ensure higher accuracy. Consequently, each constructed sample undergoes a round of human review to confirm its integrity.

Following the above procedure, we construct the Cross-XQuAD dataset with 1,190 parallel samples in four languages which results in 4,760 samples in total. It is by far the largest multilingual evaluation dataset with cross-lingual consistency assessment capabilities.

4 `CrossIn` Method

To address the imbalance across languages in English-centric LLMs pre-training and fine tuning, we explore strategies to enhance multilingual proficiency through cross-lingual instruction tuning. Typically, instruction tuning relies on monolingual training samples (e.g., English) (Zhu et al., 2023a), which limits the potential of massive multilingual exposure at the pre-training stage. Therefore, we aim to enrich instruction tuning from the language perspective. Since all languages share the compressed representation space, cross-lingual instruction tuning can efficiently boost the model's task-solving and multilingual capabilities within a single process. Specifically, we propose `CrossIn` approach to mix language compositions at the sample level to enforce the information flow across languages. We hypothesize that our method can further boost multilingual performance by encouraging language-level generalization.

The training data can be divided into three main aspects: **`Base`**, **`CrossIn`**, **`Trans`**.

- **`Base`**: This part includes the foundational instruction tuning datasets, where the model is relying on to learn all basic capabilities. Given that English datasets are the most resource-rich and of the highest quality, we focus

Type	Instruction	Output
Original Sample	*Explain what a "food desert" is and why they are a problem.*	*A "food desert" is an area with little or no access to affordable, healthy food options....*
CrossIn$_{en2x}$	*Explain what a "food desert" is and why they are a problem.*	"食物沙漠" 是一个几乎没有或根本没有负担得起的健康食品选择的地区...
CrossIn$_{x2en}$	解释什么是"食物沙漠"，以及为什么它们是一个问题	*A "food desert" is an area with little or no access to affordable, healthy food options....*
CrossIn$_{x2x}$ (zho-spa)	解释什么是"食物沙漠"，以及为什么它们是一个问题	*Un "desierto alimentario" es un área con poco o ningún acceso a opciones de alimentos saludables y asequibles...*
CrossIn$_{x2x}$ (eng-spa)	Explain what a "food desert" is and why they are a problem.	*Un "desierto alimentario" es un área con poco o ningún acceso a opciones de alimentos saludables y asequibles...*
CrossIn$_{x2x}$ (spa-zho)	Explique qué es un "desierto alimentario" y por qué son un problema.	"食物沙漠" 是一个几乎没有或根本没有负担得起的健康食品选择的地区...
Translation	*Translate the following sentence into English.* 解释什么是"食物沙漠"，以及为什么它们是一个问题	*Explain what a "food desert" is and why they are a problem.*

Table 2: One example from the Alpaca dataset. It is further transformed into cross-lingual instruction tuning datasets and translation tasks.

on using common English instruction tuning datasets.

- **CrossIn**: It comprises cross-lingual instruction tuning datasets, where instruction and output are featured in two different languages. The dataset aims to align representations across languages, thus enforcing the knowledge-sharing ability of LLMs.
- **Trans**: It consists of translation pairs for instructions. We hypothesize that if the model concurrently learns these translation tasks, it could facilitate the transfer of knowledge between languages.

For **Base**, we leverage existing datasets, we create the **CrossIn** and **Trans** datasets, where we use the Alpaca (Taori et al., 2023) dataset as the source. Examples are shown in Table 2.

For **CrossIn** dataset, we create three variants as the following recipes:

- CrossIn$_{en2x}$: Instructions are provided in English, and we choose the output language randomly. Given the rich prior knowledge available in English, this approach aims to transfer English knowledge to other languages.
- CrossIn$_{x2en}$: Instruction language is chosen randomly, and output is fixed in English. This approach aims to unify multilingual instructions into responses centered around English.
- CrossIn$_{x2x}$: The languages for both the instruction and the output are selected randomly. This approach seeks to facilitate bi-directional alignment across all languages.

Algorithm 1 CrossIn$_{x2x}$ with translation

$\mathcal{S} \leftarrow$ Total number of samples
$\mathcal{L} \leftarrow$ {"English", "Spainish", "Chinese", "Vietnamese"}
$\mathcal{D} \leftarrow$ Seed Parallel Instructions Dataset
$\mathcal{C} \leftarrow \emptyset$
$\mathcal{T} \leftarrow \emptyset$
$t_p \leftarrow$ Translation Prompt
for $i \leftarrow 1$ to $\mathcal{S}$ **do**
  $s \leftarrow$ Random sample from $\mathcal{D}$
  $l_{in}, l_{ot} \leftarrow$ Random sample from $\mathcal{L}$
  $\mathcal{C} \leftarrow \mathcal{C} \cup (\mathcal{D}[l_{in}][s], \mathcal{D}[l_{ot}][s])$
  $l_t \leftarrow$ Random sample from $\mathcal{L}$
  $\mathcal{T} \leftarrow \mathcal{T} \cup (t_p, D[l_t][s], D[\text{“}English\text{”}][s])$
end for

Previous work shows that incorporating sample translation helps map English to other languages, allowing the model to generalize English knowledge in a broader space (Zhu et al., 2023b). For an extensive comparison, we also investigate how adding a separate translation task might enhance the multilingual abilities of LLMs, compared with using cross-lingual instruction tuning alone. More specifically, aside from the CrossIn data, we add a direct translation task of instructions from English to other languages. The influence on model performance of additional instruction translation is discussed in Section 5.3.

Algorithm 1 illustrates the complete algorithm to create CrossIn$_{x2x}$ with translation dataset, where $\mathcal{S}$ is the desired number of samples to be added with the **Base**. $\mathcal{C}$, $\mathcal{T}$, l_{in} indicate **CrossIn**, **Trans** and the sampled language, respectively.

Models	Cross-XQuAD			Cross-MMLU			Cross-LogiQA		
	Acc	Consis	AC3	Acc	Consis	AC3	Acc	Consis	AC3
General LLMs									
ChatGPT-3.5	90.6	83.7	87.0	66.8	51.8	58.4	53.3	40.5	46.0
LLaMA-2-7B-Chat (Touvron et al., 2023b)	74.9	67.5	71.1	40.1	42.0	41.1	36.8	43.5	39.9
Mistral-7B-Instruct-v0.2 (Jiang et al., 2023)	84.6	72.2	77.9	49.0	26.2	34.1	46.0	38.5	41.9
LLaMA-7B (Touvron et al., 2023a)	40.3	21.5	28.0	29.8	27.8	28.8	27.6	23.0	25.1
m-LLaMA-7B (Zhu et al., 2023b)	46.8	41.1	43.8	26.7	22.3	24.3	28.1	22.0	24.7
Base Model: *Gemma-2B* (Team et al., 2024)									
Tuning w/ Alpaca	42.0	49.7	45.5	36.0	59.8	45.0	28.3	63.8	39.2
Tuning w/ Platypus	60.8	55.8	58.2	36.5	29.7	32.7	36.4	47.9	41.3
CrossIn$_{en2x}$	**60.1**	62.8	**61.5**	39.2	43.0	41.0	39.5	37.8	38.6
CrossIn$_{x2en}$	54.2	**64.7**	59.0	**41.2**	**57.8**	**48.1**	36.8	**48.3**	41.8
CrossIn$_{x2x}$	53.3	64.3	58.3	37.0	54.5	44.1	**39.6**	46.2	**42.6**
Base Model: *Mistral-7B-v0.1* (Jiang et al., 2023)									
Tuning w/ Alpaca	62.2	52.9	57.2	36.2	43.5	39.5	35.7	33.8	34.7
Tuning w/ Platypus	61.1	33.2	43.0	38.8	20.2	26.5	47.9	29.8	36.8
CrossIn$_{en2x}$	74.9	64.0	69.0	41.0	41.5	41.2	44.6	40.1	42.2
CrossIn$_{x2en}$	77.4	63.8	69.9	34.8	**47.2**	40.1	45.3	42.5	43.8
CrossIn$_{x2x}$	**78.6**	**67.9**	**72.9**	**41.0**	42.3	**41.7**	**48.9**	**48.3**	**48.6**

Table 3: Experimental results on three cross-lingual consistency datasets: Cross-XQuAD, Cross-MMLU, Cross-LogiQA. Three metrics presented are Accuracy (ACC), Consistency (Consis), and AC3 as introduced in Section 3.

5 Experiments

5.1 Experimental Setting

In our experiments, we selected four languages: English, Chinese, Vietnamese, and Spanish across all three datasets. We utilized two representative open LLM as base model: *Mistral-7B-v0.1* (Jiang et al., 2023) and *Gemma-2B* (Team et al., 2024). For base models, we employed the Platypus (Lee et al., 2023) corpus as the **Base** dataset for instruction tuning, since previous work shows that it can enable models' higher diverse and robust generalization capabilities than the Alpaca dataset.

For the **CrossIn** instruction tuning data, we utilize the Alpaca (Taori et al., 2023) corpus as the seed dataset. This dataset is expanded into a multilingual format to four languages using an off-the-shelf translation engine, producing a total of (52k×4) samples. From the enriched datasets, both the **CrossIn** and **Trans** parts can be formulated in a variant number of samples. While the Alpaca dataset lacks the complex problem-solving capabilities of the **Base** set from Platypus, it contains English instructions without complex elements like coding and math, which results in a higher translation quality. Meantime, this setup allows us to investigate whether a dataset of simple instructions can adequately support effective knowledge alignment across languages.

In model training, we leverage LoRA (Hu et al., 2022) with $rank = 64$ as a parameter-efficient way to train LLMs. For fair comparison, we fine-tune base models with either the Platypus or Alpaca dataset with the same set of hyperparameters. Besides, following standard benchmarks, we also compared several representative general-purpose LLMs including *ChatGPT-3.5*, *LLaMA-2-7B-Chat*, *Mistral-7B-Instruct-v0.2*, *m-LLaMA-7B* and its base model, *LLaMA-7B*.

5.2 Main Results and Analysis

Table 3 shows the benchmark results of current general LLMs and models tuned with Alpaca, Platypus and different CrossIn variants. Our findings can be summarized as follows.

English-centric LLMs do not perform well on our multi-lingual benchmark. First, we evaluate the performance of representative LLMs using our benchmarks and observed that *ChatGPT-3.5* exhibits outstanding performance across all three test-sets, indicating strong multilingual capabilities and consistency. For open-source models, we observe that models after instruction tuning (e.g.,*LLaMA-2-7B-Chat*, *Mistral-7B-Instruct-v0.2*) significantly outperform the non-tuned models (e.g., *LLaMA-7B*, *m-LLaMA-7B*) on all fronts, while their accuracy and cross-lingual consistency lag behind that of *ChatGPT-3.5*. Moreover, *m-LLaMA-7B* demon-

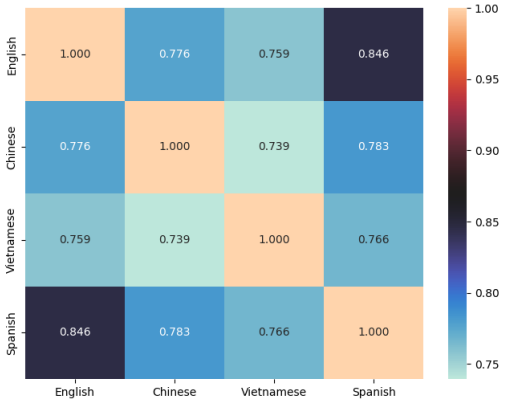


Figure 2: Consistency score between languages on Cross-XQuAD with $\texttt{CrossIn}_{x2x}$ method

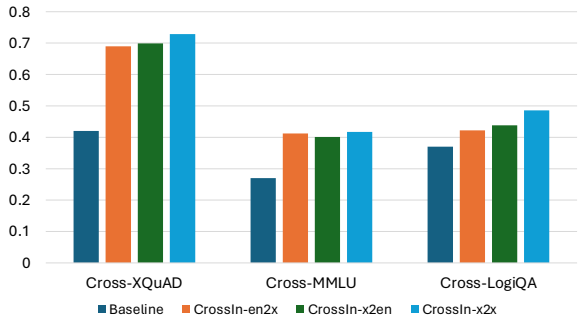


Figure 3: Results of different cross-lingual instruction tuning methods compared with baseline.

strated some improvements over *LLaMA-7B* in the Cross-XQuAD dataset, but it only managed to achieve similar results on the Cross-MMLU and Cross-LogiQA. This suggests that a purely monolingual data mix may not be adequate for training models on complex multilingual tasks, highlighting the importance of our proposed approach.

English-centric instruction tuning is limited. We analyzed the performance of base models fine-tuned on different original instruction datasets (i.e., Alpaca and Platypus). Our findings indicate that models exhibit distinct characteristics depending on the instruction tuning corpus. Fine-tuning with Platypus results in higher accuracy, potentially due to the diversity of tasks in the dataset. Conversely, models fine-tuned with Alpaca shows a higher consistency across most benchmark datasets, albeit with marginally lower accuracy. These observations suggest that Alpaca may be less effective than Platypus in augmenting LLMs with task-solving and reasoning. In addition, fusing a wide range of knowledge in English could potentially lead to a forgetting of information in other languages, thus affect the consistency. This results show a trade-off between accuracy and consistency from fine-tuning on different English-centric instruction tuning corpora. We aim to bridge the gap of both datasets, thereby enhancing both accuracy and consistency.

`CrossIn` is simple but effective. We further review the results from our `CrossIn` instruction tuning method, which leverages the strengths of both the English-centric Platypus and the diverse Multilingual Alpaca datasets. By implementing the `CrossIn` augmentation, we successfully raised the AC3 score by 30% on the Cross-XQuAD benchmark and by about 12% on both the Cross-MMLU and Cross-LogiQA testsets. This improvement was achieved using the $\texttt{CrossIn}_{x2x}$ approach with the Mistral-7B-v0.1 as the foundational model. Enhancements were evident in the model's accuracy and consistency across various languages, contributing to the higher AC3 scores. Our findings highlight the effectiveness of the `CrossIn` method in enriching a model's performance on multilingual tasks. By starting with a task-diverse, strong instruction set from the Platypus dataset and integrating simpler, language-varied data from Alpaca, we crafted a cross-lingual knowledge base that significantly improve accuracy and consistency in multilingual understanding.

Language discrepancy affects consistency. We investigate the consistency scores across all pairs of languages. As shown in Figure 2, Spanish and English exhibit the highest consistency, potentially due to their linguistic similarities, among all other language pairs. On the other hand, Chinese and Vietnamese have the lowest correlation, which may be attributed to their completely distinct character sets. Apart from the linguistic discrepancies, this could also stem from language bias during the pre-training phase of language models. When considering the consistency score between English and other languages, Vietnamese, typically categorized as a low-resource language in pre-training, shows the least consistency with English. This points to the importance of diversifying the data used in training language models to ensure fair and effective language representation, particularly for languages that are typically categorized as low-resource.

5.3 Ablation Study

We conduct three comprehensive ablation studies to systematically assess the effects of various data formations, the integration of translation data, and

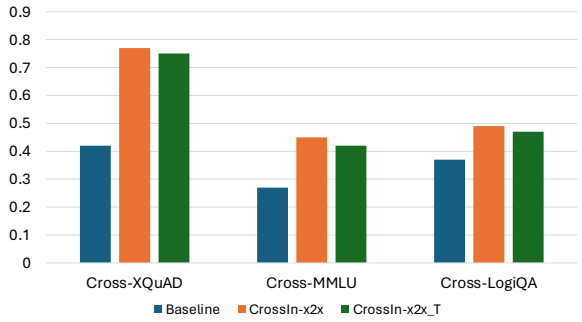


Figure 4: Comparison of AC3 score of adding translation data in cross-lingual instruction tuning.

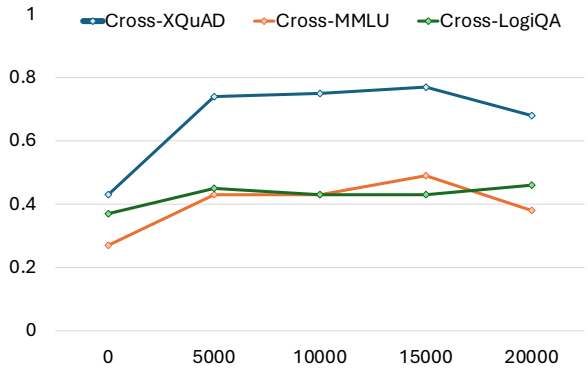


Figure 5: Comparison of AC3 score by adding different numbers of `CrossIn` data. Base model: *Mistral-7B-v0.1*

the influence of different alignment dataset sizes on the performance of our models, aiming to identify key factors that enhance or inhibit their effectiveness.

Data Formulation Comparison. Figure 3 shows the AC3 scores from three tests when the language backbone is the Mistral-7B-v0.1. The results make it clear that methods designed for cross-lingual instructions work better than the basic method, which only uses English-centric instruction tuning data from Platypus or Alpaca. In particular, the $\texttt{CrossIn}_{x2x}$ method does much better than the $\texttt{CrossIn}_{en2x}$ and $\texttt{CrossIn}_{x2en}$ methods. This suggests that fully mixing multiple languages ($\texttt{CrossIn}_{x2x}$) can make the most of what the Mistral-7B-v0.1 model offers by effectively using data from different languages. The mixed composition in training examples seems to help the model understand and apply knowledge from one language to another, leading to more accurate and consistent results.

Efficacy of Translation Data. Figure 4 compares the performance of the $\texttt{CrossIn}_{x2x}$ method with the $\texttt{CrossIn}_{x2x\_T}$ strategy, which adds translations to the Alpaca samples (as described in Algorithm 1). The experimental results indicate that additional translation pairs does not bring performance gains. We speculate that this is because tasks included in our benchmark focus on understanding and reasoning, and the cross-lingual instruction tuning approach stimulate both of them under a multilingual setting. Additionally, the translations used here may be too basic, especially compared to larger datasets like WikiMatrix. This suggests that improving multilingual knowledge alignment may be better achieved through a mixed-language approach at the sample level rather than by incorporating simple translation data.

Essential Cross-Lingual Data Quantities. Figure 5 shows the AC3 score of the LLMs with different quantity of cross-lingual alignment data. It can be shown that adding 5000 alignment data could already achieve a good result of cross-lingual consistency, there are not much improvement trend if we add more data. The observation that only a small amount of cross-lingual alignment data is required to achieve satisfactory consistency in LLMs can be attributed to its efficient learning mechanism. This characteristic allows the model to quickly assimilate and generalize from limited data, making it particularly adept at few-shot learning scenarios. Additionally, the model's pretraining on diverse linguistic corpora might have already equipped it with a foundational understanding of various languages, thereby reducing the need for extensive alignment data to bridge linguistic gaps. This efficient use of data not only demonstrates the model's robustness but also highlights its practicality in situations where data availability is constrained.

6 Conclusion

In this paper, we presented a study on improving cross-lingual knowledge alignment of multilingual large language models, and contributed to both evaluation benchmarks and methodologies. We built a machine comprehension dataset that is a robust resource for extensive multilingual evaluation, emphasizing cross-lingual consistency in compensation with previous datasets. Our cross-lingual instruction tuning method `CrossIn` brought significant improvements in knowledge accuracy and consistency across languages, highlighting the potential of efficient tuning to create more robust multilingual large language models.

Limitations

Our approach depends on the availability of high-quality translation and cross-lingual data, which may not be accessible for all languages. Addressing these data availability challenges is essential for further research on enhancing multilingual consistency in large language models.

In this study, we did not examine the impact of our cross-lingual data formulation on the pretraining stage of large language models. Pre-training is crucial as it significantly shapes the model's foundational knowledge and capabilities. Considering the larger scale of pretraining compared to fine-tuning, exploring whether our method could improve the efficiency and effectiveness of pretraining multilingual language models is a vital direction for future research. However, conducting such an ablation study on the pre-training stage is computationally demanding and may not be feasible with limited resources.

Acknowledgment

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme, Ministry of Digital Development and Information (MDDI) and Infocomm Media Development Authority under National Large Language Models Funding Initiative, AI For Mother Tongue Language Learning and Public Sector Translational R&D Grant Funding Initiative (TRANSGrant). The aim of TRANSGrant is to tap on the research community to solve public sector challenges with innovative use of digital technologies. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, Ministry of Digital Development and Information (MDDI) and Infocomm Media Development Authority.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of english pretrained models.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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. *arXiv e-prints*, pages arXiv–2307.

Longxu Dou, Qian Liu, Guangtao Zeng, Jia Guo, Jiahui Zhou, Wei Lu, and Min Lin. 2024. Sailor: Open language models for south-east asia.

Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair nlp models.

Changjiang Gao, Hongda Hu, Peng Hu, Jiajun Chen, Jixing Li, and Shujian Huang. 2024. Multilingual pretraining and instruction tuning improve cross-lingual knowledge alignment, but only shallowly.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Tannon Kew, Florian Schottmann, and Rico Sennrich. 2023. Turning english-centric llms into polyglots: How much multilinguality is needed?

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023. Coannotating: Uncertainty-guided work allocation between human and large language models for data annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1487–1505.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

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.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2022. A survey on bias and fairness in machine learning.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. Cross-lingual consistency of factual knowledge in multilingual language models.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers.

Tal Schuster, Adam D. Lelkes, Haitian Sun, Jai Gupta, Jonathan Berant, William W. Cohen, and Donald Metzler. 2023. Semqa: Semi-extractive multi-source question answering.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, et al. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint 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, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Bin Wang, Zhengyuan Liu, Xin Huang, Fangkai Jiao, Yang Ding, Ai Ti Aw, and Nancy F. Chen. 2023. Seaeval for multilingual foundation models: From cross-lingual alignment to cultural reasoning.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

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

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023a. Multilingual machine translation with large language models: Empirical results and analysis.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023b. Extrapolating large language models to non-english by aligning languages.

A Appendix

A.1 Prompt for Building Cross-XQuAD Data

Initial Prompt:
Given the following context: {context}.

Question: {question}
Could you generate 4-multiple-choice with 3 incorrect options. Please try to make the incorrect options challenging by extracting them from the context as much as possible. No explanation. Just give me the 4 options. The options should start with (A), (B), (C), (D).
The correct answer is: (A) {answer}

Figure 6: Prompt For Generating English Choice

Translation Prompt:
Given the following context: {context}

Could you translate the above 4 choices into {language}. No explanation. Just give me the 4 options. The options should start with (A), (B), (C), (D).
The correct answer is: (A) {answer}

Figure 7: Prompt to Translate English Choice

A.2 Fine-tuning Parameters

Hyperparameter	Value
learning_rate	1e-4
batch_size	16
epochs	1
lora_rank	64
lora_alpha	128
lora_trainable	p_proj, k_proj, v_proj, o_proj, gate_proj, down_proj, up_proj
modules_to_save	embed_tokens, lm_head
lora_dropout	0.05
warmup_ratio	0.03
weight_decay	0
optimizer	Adam
bf16	True

Table 4: Fine-tuning Hyperparameters

Evaluating Dialect Robustness of Language Models via Conversation Understanding

Dipankar Srirag[1] **Nihar Sahoo**[2] **Aditya Joshi**[1]
[1]University of New South Wales, Sydney, Australia
[2]Indian Institute of Technology Bombay, India
{d.srirag, aditya.joshi}@unsw.edu.au nihar@cse.iitb.ac.in

Abstract

With an evergrowing number of LLMs reporting superlative performance for English, their ability to perform equitably for different dialects of English (*i.e.*, dialect robustness) needs to be ascertained. Specifically, we use English language (US English or Indian English) conversations between humans who play the word-guessing game of 'taboo'. We formulate two evaluative tasks: target word prediction (TWP) (*i.e.*, predict the masked target word in a conversation) and target word selection (TWS) (*i.e.*, select the most likely masked target word in a conversation, from among a set of candidate words). Extending MD3, an existing dialectic dataset of taboo-playing conversations, we introduce M-MD3, a target-word-masked version of MD3 with the en-US and en-IN subsets. We create two subsets: en-MV (where en-US is transformed to include dialectal information) and en-TR (where dialectal information is removed from en-IN). We evaluate three multilingual LLMs–one open-source (Llama3) and two closed-source (GPT-4/3.5). LLMs perform significantly better for US English than Indian English for both TWP and TWS tasks, for all settings, exhibiting marginalisation against the Indian dialect of English. While GPT-based models perform the best, the comparatively smaller models work more equitably after fine-tuning. Our evaluation methodology exhibits a novel and reproducible way to examine attributes of language models using pre-existing dialogue datasets with language varieties. Dialect being an artifact of one's culture, this paper demonstrates the gap in the performance of multilingual LLMs for communities that do not use a mainstream dialect.

1 Introduction

Large language models (LLMs)[1] based on Transformers (Vaswani et al., 2017) are the state-of-the-art in natural language processing (NLP), often reporting superlative performance on several NLP tasks (Zhao et al., 2023). These models predominantly use English language data in their pre-training corpus. However, being a widely spoken language, English takes multiple forms in different parts of the world. These forms, called dialects or national varieties of English, collectively constitute the World Englishes (Bolton, 2012). While research papers introducing LLMs report performance on English language datasets, recent works highlight the performance gap between US English and other dialects of English for several natural language processing tasks (Joshi et al., 2024).

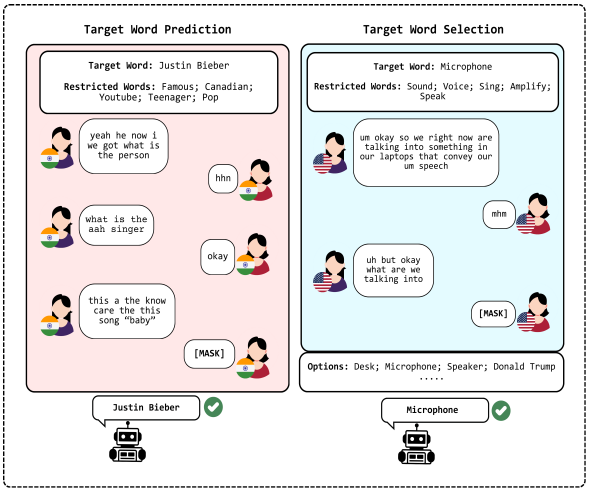


Figure 1: Illustration of the two tasks: Target word prediction (TWP) and Target word selection (TWS). and are the describer and the guesser respectively in a word-guessing game of taboo. and refer to Indian English and US English respectively.

Our paper examines cultural considerations of evaluating LLMs through the prism of dialect robustness via conversation understanding. The choice of conversation understanding as a domain for evaluation emerges from the fact that dialectal features are most visible in free-flowing conver-

[1]We use 'language models' and 'large language models/LLMs' interchangeably in this paper.

Proceedings of the 2nd SUMEval Workshop: Scaling Up Multilingual & Multi-Cultural Evaluation, pages 24–38
January 20, 2025.

sations (Negro and Vietti, 2006). Therefore, we investigate the research question:

> "*In comparison with US English, how effectively can LLMs understand conversations between speakers of other national varieties of English?*"

To address the research question, we use a pre-existing dataset– **MD3** (Eisenstein et al., 2023) that consists of manually transcribed dialogues between pairs of human participants where each pair speaks either Indian English or US English. The participants engage in a focused conversation: they play the word-guessing game based on the game of 'Taboo' (Wikipedia, 2023). In the game, a describer must get a guesser to identify a **target word** but must not use a set of related words known as **restricted words** while describing the target word. Using this dataset of dialectal dialogues, we introduce two tasks to evaluate the dialect-robustness of LLMs to understand conversations. They are: (a) Given an input conversation with the target word masked, can the LLM *predict* the target word? (referred to as **target word prediction**) (b) Given an input conversation with the target word masked along with a set of candidate target words, can the LLM *select* the correct target word? (referred to as **target word selection**). Our approach of masking the target word is similar to Dey and Desarkar (2023), who show that masked word prediction may correlate with automatic dialogue evaluation metrics. Figure 1 shows an example of the two tasks, where the language model predicts '*Justin Bieber*' for target word prediction, and selects '*microphone*' among the set of options for target word selection[2]. For the two tasks, we extend MD3 to create a target-word-**M**asked **M**ulti-**D**ialect **D**ataset of **D**ialogues (**M-MD3**)[3]. M-MD3 consists of (a) conversations between Indian English speakers (en-IN), and conversations between US English speakers (en-US), (b) en-US conversations transformed into en-IN using rule-based perturbations (en-MV), (c) en-IN with dialectal information removed (en-TR). We evaluate the performance of three SOTA large language models (LLMs), one open-source and two closed-source, employing zero-shot prompting on both pre-trained and fine-tuned models (where available). Our evaluation methodology derives from past work that evaluates LLMs by providing a set of task-specific examples (Wang et al., 2023). Of particular relevance is the work by Chalamalasetti et al. (2023), who generate word game conversations using LLMs and evaluate their ability to predict the target word. The contributions of our work are:

- We create M-MD3, an extension of MD3, that deals with two novel evaluative tasks for dialect robustness: target word prediction and target word selection.
- Our evaluation demonstrates a degraded performance in the case of Indian English as compared to US English for all models, supporting existing social disparities between US and Indian culture in the LLM representations (Khandelwal et al., 2024).
- A comprehensive error analysis to identify specific conditions under which fine-tuning enhances the model's performance on Indian English conversations.

Since several LLMs have been deployed as publicly available dialogue agents[4], it is imperative that they can understand the conversations of users belonging to diverse English-speaking subgroups. In the case of our paper, this refers to dialectal variations, considering them as a proxy to culture. The rest of the paper is organized as follows. Section 2 introduces our evaluation methodology. The experiment setup and results are in Sections 3 and 4 respectively.

2 Methodology

We present our method step-by-step, with a detailed overview of our evaluation methodology described in Figure 2. We select two subsets available in MD3: en-IN and en-US, and filter out the conversations where the guesser could not identify the target word. We extend MD3 to include two additional sets of conversations—en-MV and en-TR, and mask the target words in all four subsets to create M-MD3. We ensure that the mask token always appears at the end of the conversation, warranting the use of auto-regressive models. This is done

[2]We run experiments on both the tasks for both US and Indian English conversations. While the examples show expected output, the LLM may or may not produce the same in the case of our experiments. That is the crux of the evaluation.

[3]M-MD3 dataset and the related code will be made publicly available at ANONYMOUS.

[4]ChatGPT https://chat.openai.com/; Accessed on 9th April 2024.

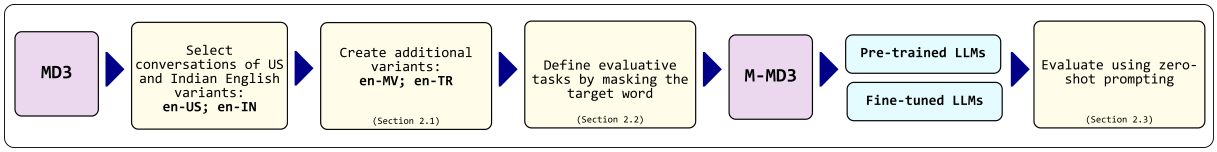


Figure 2: Steps for evaluation of dialect robustness.

by pruning the conversation to the turn where the guesser utters the target word[5].

Transforming text in en-US to other dialectal English text has been explored for low-resource settings (Held et al., 2023; Xiao et al., 2023; Liu et al., 2023). To evaluate the efficacy of synthetically transformed dialogues, we extend the dataset of dialectal dialogues to include two additional sets of conversations– en-MV and en-TR.

en-MV We use Multi-VALUE (Ziems et al., 2023) to transform en-US conversations into en-IN conversations. We call this set of conversations created by rule-based transformations en-MV.

en-TR We prompt GPT-4 Turbo Preview(GPT-4; OpenAI et al. 2024) to remove dialectal information from en-IN. The resultant set of conversations is known as en-TR. The prompt[6] used to generate such conversations is given below:

> "*Normalise the conversation. Remove all exaggerations and dialectal information. Return a neutral response.*"

2.1 Extending MD3

The use of GPT-4 to transform en-IN conversations sometimes leads to the generation of conversation summaries rather than transformed conversations[7]. Due to the varying lengths of speaker turns, transforming en-US conversations using Multi-VALUE occasionally fails to output a result. Such failed transformations are excluded from both the subsets of transformed (en-MV, en-TR) conversations, leading to fewer conversations in en-MV and en-TR as compared to en-US and en-IN, respectively, as shown in Tables 1 and 2.

2.2 Analysis

Table 1 reports some of the constructional statistics of M-MD3. For each subset, it reports the average number of dialogue turns per conversation, the average word count for the dialogues uttered by both the describer and the guesser, and the number of conversations with single-word versus multiple-word reference target words. The target words '*microphone*' and '*Justin Bieber*' in Figure 1 are examples of single-word and multiple-word reference target words, respectively.

We notice a higher number of average turns and words spoken in en-IN conversations compared to en-US conversations. This is due to the en-US speakers being more familiar with the target word compared to en-IN speakers, leading to shorter gameplay time (Eisenstein et al., 2023). The trend is also carried over to the transformed conversations in en-MV (derived from en-US) and en-TR (derived from en-IN).

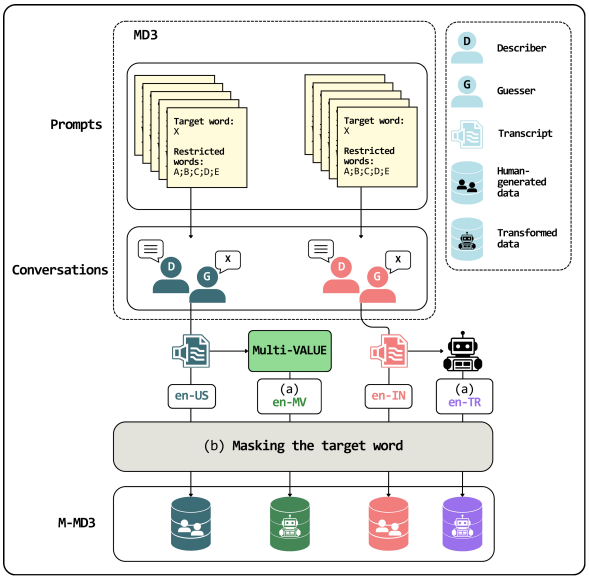


Figure 3: M-MD3 as an extension of MD3: (a) Creation of en-MV and en-TR, and (b) Creation of target-word-masked conversations.

2.3 Task Definition

As shown in Figure 3, we mask the target word in the conversations from all four subsets. The target word occurs in the last dialogue turn of the conversation, which is spoken by the guesser[8]. As a

[5]Details on the masking method with examples are provided in Appendix A.

[6]The forms are experimentally determined using a few test examples.

[7]More details with examples are discussed in Appendix B.

[8]This always holds because of the way we process the conversations.

Subset	Avg. turns	Avg. words	Single	Multiple
en-US	4.1	42.1	308	106
en-IN	6.8	57.4	153	59
en-MV	4.9	35.2	245	87
en-TR	6.3	42.7	121	50

Table 1: Constructional Statistics of M-MD3. *Single* and *Multiple* refer to the number of conversations with single-word and multiple-word reference targets, respectively.

result, we formulate two tasks where the expected output is to fill the correct word at the masked position:

- **Target Word Prediction (TWP)**: Given a conversation with the target word masked, predict the word.
- **Target Word Selection (TWS)**: Given a conversation with the target word masked and a set of candidate target words, select one among the candidate set.

In the case of TWP, the LLM may generate any word within its learned vocabulary, with the expected output being the reference target word. In the case of TWS, we provide the LLM with a masked conversation and a set of all target words in the M-MD3 corpus. The LLM must then select the most likely target word.

We then use prompting on three LLMs to perform both tasks (TWP and TWS). As LLMs, we choose models that have been optimised to follow natural language instructions. In our case, the instruction is to either predict the masked target word or select a word from candidate words. Specifically, we use one open-source model, namely, Llama 3 70B Chat (LLAMA-3; Grattafiori et al. 2024), and two closed-source models, namely, GPT-4 and GPT-3.5 Turbo 0125 (GPT-3.5; Ouyang et al. 2024).

3 Experiment Setup

We report the performance on pre-trained and fine-tuned versions of multilingual LLMs using zero-shot prompting. Fine-tuning is always done '*in-dialect*' in our case, although there is no reason to believe that cross-dialect fine-tuning is not possible.

3.1 Model Parameters

Experiments on GPT-4 and GPT-3.5 are conducted using OpenAI's API[9]. GPT-3.5 is fine-tuned for 5 epochs, separately for every subset. We select top_p as 0.2 to restrict variability in output generation.

LLAMA-3 is fine-tuned for 20 epochs, with a batch size of 16, Paged 8-bit AdamW (Dettmers et al., 2022) as the optimiser and a learning rate of 2e-4. We use QLoRA adaptors, targeting all linear layers, as recommended by Dettmers et al. (2024). All experiments on LLAMA-3 were performed using two A100 GPUs.

3.2 Metrics

We report our results on two metrics: *accuracy* and *similarity*. *Accuracy* is the proportion of conversations where the LLM generated the correct target word. This is a strict metric in that it requires the LLM to generate an exact match to the reference target word. In the case of TWP, the LLM will choose from all the words within its vocabulary, while in the case of TWS, the LLM will choose from the set of candidate target words. Therefore, it is trivial that the accuracy for TWS is expected to be higher than that for TWP. Accuracy metric penalizes models even if the generated target word partially matches with the reference target word in case of multi-word reference target as described in Section 2.2. As *similarity*, we report the cosine similarity between the Sentence-BERT embeddings (Reimers and Gurevych, 2019) of the reference target word and the generated target word. This allows for similar but inexact words generated by the LLM to be acceptable to the similarity score.

3.3 Experiments

We perform experiments on both the tasks (TWP and TWS) using all models({pre-trained and fine-tuned} × {GPT-4, GPT-3.5, LLAMA-3 }). All results are reported only on the test split of each subset of conversations. All fine-tuned models are fine-tuned on the training and validation set using instruction fine-tuning. GPT-4 could not be fine-tuned because doing so is restricted by OpenAI at the time of writing this paper. The statistics of **Train**, **Valid**, and **Test** splits of each subset of M-MD3 are reported in Table 2.

[9]OpenAI API https://platform.openai.com/docs/api-reference; Accessed on 18th April 2024.

Subset	Train	Valid	Test
en-US	62	41	311
en-IN	31	21	160
en-MV	49	33	250
en-TR	23	17	131

Table 2: Statistics of M-MD3.

4 Results

In this section, we compare the performance of three LLMs both quantitatively and qualitatively. Note that the same test split is used to evaluate both pre-trained and fine-tuned versions, ensuring that the results are comparable.

4.1 Quantitative Results

Table 3 shows the results of our experiments on each task specified in Section 2.3. We analyse the results as follows.

en-US versus en-IN The focus of this paper is to evaluate dialect robustness by comparing the performance on en-US and en-IN. All LLMs perform consistently better on en-US as compared to en-IN for all configurations. For example, in the case of LLAMA-3 and TWP, the similarity scores on the fine-tuned model are 78.0 for en-US and 66.3 for en-IN, with the drop in performance of 11.7. Even for all three models, en-US outperforms en-IN on zero-shot performance using the pre-trained model. From all results, it is clearly understood that, on average, the LLMs understand the US English dialect better than the Indian English dialect. Only considering the pre-training setting, GPT-4 outperforms other models for both en-US and en-IN. However, fine-tuning improves the performance of LLAMA-3 on en-IN, achieving better results on both tasks compared to GPT-based models. Interestingly, for LLAMA-3, the performance improvement after fine-tuning on en-IN is greater compared to fine-tuning on en-US (represented by Δ).

Impact of transforming conversations As discussed in section 2.1, we introduced two synthetically transformed subsets, en-MV and en-TR, to assess the importance of dialectal features in LLMs' understanding of conversations. Table 3 shows that, on pre-trained models, en-TR conversations have better performance compared to original en-IN conversations. This suggests that after removing the dialectal information from en-IN, the resulting en-TR conversations are close to the distribution of the dialect that the LLM understands. This behaviour is better reflected in GPT-3.5, potentially, because the LLM has a poor understanding of en-IN as compared to the other two LLMs. Additionally, fine-tuning on en-TR conversations does not improve the task performances in comparison to that on en-IN. This supports the hypothesis that the removal of dialectal information brings the resulting conversation closer to the dialectal distribution that LLMs understand than the original dialect.

In the case of en-MV, the task performances are consistently lower compared to en-US. For example, in the case of GPT-3.5 and TWS, the similarity scores on the fine-tuned model are 80.8 for en-US and 71.5 for en-MV. This degraded performance shows that the rule-based transformation into en-IN from en-US reduced the understanding capacity of LLMs for the resulting conversations, further strengthening our hypothesis that LLMs perform well for US English dialects compared to any other varieties, similar to findings of Ryan et al. (2024).

Shorter turns versus Longer turns A trend appears between the performances of models on each subset of conversations and the constructional properties of these conversations discussed in Section 2.2. Models report their best performances on the subset with the smallest number of average turns in a conversation (en-US), and report the worst performance on the subset with the highest number of average turns in a conversation (en-IN).

TWP versus TWS We now compare the performances of TWP and TWS. As expected, the similarity and accuracy are higher in the case of TWS compared to TWP for all three models, with one exception: the pre-training performance of GPT-3.5 on en-MV, where TWP slightly outperforms TWS. Note that, for pre-trained LLAMA-3, the accuracy on en-IN is 43.8 for TWP and 56.9 for TWS. Across all configurations, fine-tuning consistently improves the performance of both TWP and TWS. GPT-4 performs best (only for pre-trained models) for both TWP and TWS tasks for all subsets.

Model Comparison It can be easily observed from Table 3 that the GPT-4 outperforms the other two LLMs in the pre-training setting. Interestingly, for TWS, GPT-4 pre-training performances are better than fine-tuning performances of GPT-3.5 and LLAMA-3 in most of the cases. Also, GPT-4 performs almost equally well for each subset of M-

Model	Subset	TWP						TWS					
		Similarity			Accuracy			Similarity			Accuracy		
		PT	FT	Δ	PT	FT	Δ	PT	FT	Δ	PT	FT	Δ
GPT-4	en-US	**77.4**	-	-	**67.8**	-	-	**85.7**	-	-	**78.8**	-	-
	en-IN	63.0	-	-	45.6	-	-	79.0	-	-	72.5	-	-
	en-MV	75.6	-	-	60.0	-	-	83.6	-	-	74.4	-	-
	en-TR	62.8	-	-	45.8	-	-	83.4	-	-	77.1	-	-
	δ	-14.4	-	-	-22.0	-	-	-6.7	-	-	-6.3	-	-
GPT-3.5	en-US	**66.3**	**72.2**	5.9	**52.7**	**59.1**	6.4	66.4	**80.8**	14.4	50.8	**71.3**	20.5
	en-IN	53.2	59.1	5.9	34.4	40.0	5.6	61.9	70.7	8.8	47.5	60.6	13.1
	en-MV	57.6	71.3	13.7	40.0	54.4	14.4	52.4	71.5	19.1	31.6	57.6	26.0
	en-TR	59.4	61.0	1.6	39.7	41.2	1.5	**70.7**	73.0	2.3	**57.3**	60.3	3.0
	δ	-13.1	-13.1	-	-18.3	-19.1	-	-4.5	-10.1	-	-21.0	-16.2	-
LLAMA-3	en-US	**70.8**	**78.0**	7.2	**60.5**	**65.3**	4.8	**78.0**	**81.8**	3.8	**67.5**	**74.6**	7.1
	en-IN	59.8	66.3	6.5	43.8	54.4	10.6	68.8	80.8	12.0	56.9	74.4	17.5
	en-MV	68.6	73.8	5.2	54.0	61.6	7.6	72.3	77.6	5.3	58.8	67.2	8.4
	en-TR	60.7	57.5	-3.2	45.8	42.7	-3.1	70.8	79.5	8.7	60.3	72.5	12.2
	δ	-11.0	-11.7	-	-16.7	-10.9	-	-9.2	-1.8	-	-10.6	-0.2	-

Table 3: Performance on the two tasks: TWP and TWS. PT/FT: Pre-trained/Fine-tuned. δ is the difference in performance between en-IN and en-US (en-IN minus en-US). Δ is the difference in performance between FT and PT. The best performance by a model is represented with **bold** numbers. The best performance for a subset of conversations is represented with underlined numbers.

MD3. This shows that GPT-4 and LLAMA-3 are more inclusive for different dialectal variations of English in the pre-training and fine-tuning setting, respectively.

Pre-training versus Fine-tuning Although the pre-training performances of GPT-4 are superlative, Table 3 shows that the fine-tuning also improves the performance of GPT-3.5 and LLAMA-3 across both tasks and four subsets. Fine-tuning is more effective for en-US than en-IN in the case of GPT-3.5, whereas LLAMA-3 shows the opposite trend. For GPT-3.5, the most improvement due to fine-tuning is seen when the models are fine-tuned on en-MV, while LLAMA-3 shows the highest improvement when fine-tuned on en-IN.

4.2 Error Analysis

From **Test** set of each conversation subset, we randomly select 30 conversations that are mislabeled by GPT-4 and LLAMA-3, and manually analyse errors among all model variants across all subsets of conversations. We summarise the six error categories[10] in Table 4. The error types are:

Ambigous Descriptions (AD) This error type is observed when descriptions lack specificity (given the *situational* constraint on the describer), leading to multiple potential answers. For the example target word–'*engine*,' the description provided is–'*What we find in our. cars. in the front part?*'. Although these descriptions provide enough information to guide a human guesser to the right answer, they are often too vague to guide the LLM to a singular, correct interpretation.

Wrong Descriptions (WD) These errors occur when the guesser guesses the target word even before the describer can finish the description completely. In the case of the target word '*surname*,' the model infers '*parent*' when the description provided is–'*beside your. uh. what is your elder? Uh what is*'. While human guessers might use their cognitive bias to guess correctly without the complete description, LLMs lack the ability to understand the target word from such a description.

Broken down description of prompt word (BDD) This error occurs when the describer breaks down the target word into subwords and attempts to explain each separately. Generally, such descriptions involve longer turns. The guesser is then expected to piece together these fragments to deduce the

[10]Additonal examples for each error category are in Appendix C.

Error Type	Config	GPT-4				LLAMA-3			
		en-US	en-IN	en-MV	en-TR	en-US	en-IN	en-MV	en-TR
AD	PT	18 (5)	13 (3)	13 (6)	14 (6)	10 (6)	16 (10)	18 (15)	11 (7)
	FT	- (-)	- (-)	- (-)	- (-)	7 (6)	9 (4)	13 (13)	11 (6)
WD	PT	4 (2)	4 (4)	- (-)	3 (3)	3 (2)	5 (5)	- (-)	3 (3)
	FT	- (-)	- (-)	- (-)	- (-)	2 (2)	5 (4)	- (-)	3 (2)
BDD	PT	3 (2)	16 (5)	- (-)	7 (3)	- (-)	2 (1)	- (-)	3 (2)
	FT	- (-)	- (-)	- (-)	- (-)	- (-)	0 (0)	- (-)	2 (3)
CC	PT	6 (2)	5 (2)	12 (5)	4 (2)	4 (2)	5 (4)	5 (3)	2 (1)
	FT	- (-)	- (-)	- (-)	- (-)	2 (1)	4 (2)	3 (2)	2 (0)
PF	PT	6 (2)	2 (0)	7 (4)	4 (0)	14 (8)	3 (1)	9 (5)	4 (1)
	FT	- (-)	- (-)	- (-)	- (-)	7 (5)	1 (1)	5 (4)	3 (0)
ERR	PT	- (-)	- (-)	6 (1)	- (-)	- (-)	- (-)	4 (3)	- (-)
	FT	- (-)	- (-)	- (-)	- (-)	- (-)	- (-)	3 (1)	- (-)
$\sum$	PT	37 (13)	40 (14)	38 (16)	32 (14)	31 (18)	31 (21)	40 (29)	23 (14)
	FT	- (-)	- (-)	- (-)	- (-)	18 (13)	19 (11)	27 (21)	21 (11)

Table 4: Count of errors of GPT-4 and LLAMA-3 for each subset. PT/FT: Pre-trained/Fine-tuned. 'X (Y)' indicates that there are X errors in TWP and Y errors in TWS. $\sum$ is the sum of errors tagged in the sampled erroneous conversations by a model on a subset across all error types.

original word, as in the case of the target word '*Billie Holiday*,' the describer individually describes the subwords '*Billie*' and '*Holiday*'. In such cases, LLMs sometimes latch onto the descriptions pertaining to later subwords, predicting a partially correct target word.

Shared Cultural Context (CC) These errors arise when the human players use culturally shared notions in a conversation, often due to the describer's lack of familiarity with the target word. For example, an Indian describer explains the word '*idli*' using examples of breakfast items and then asks the guesser to infer '*Adele*'. The model is unable to understand this happening in the conversation.

Public Figure (PF) These errors pertain to inaccurate predictions generated by the model when the descriptions are about a well-known public figure. For example, the describer describes the target word '*Mike Tyson*' as '*Big guy that punched people out and he had a little bit of a lisp*,' but the model generates '*darth*'.

Fallback Error (ERR) While efforts were made to classify every mislabeled conversation into an error category, few generated target words were found to be inexact or inaccurate, even with apt descriptions in the conversations. For example, the target word–'*Rose*' and the description–'*This are the types of that's often given valentine day plant.*', the model generates '*Gift*'. This example description mentions the word *plant* which should have guided the model to a more specific target word than *Gift*.

The error types **AD**, **CC**, and **PF** test the model's ability to predict the target word based on descriptions influenced by the describer's dialect, shared notions with the guesser, and perceived notions about the target word. Also, some of the conversations fall into multiple error categories except in the case of conversations in **ERR** (which is a mutually exclusive label).

Table 4 presents the error cases in 'X (Y)' which indicates that there are X errors in TWP and Y errors in TWS for the corresponding configuration. The benefit of TWS providing options for the target word is seen in **AD**, where the alleviation is almost uniform across all dialects. The presence of direct or indirect references to the prompt word helps the LLM towards a plausible answer, in turn making it easier for them to choose an option. However, this error reduction does not extend to **CC**, which LLMs are unable to detect.

Fine-tuning helps to reduce the errors of **AD** category more for conversations of en-IN dialect compared to en-US. However, after removing the dialectal information, the conversations are insensitive to fine-tuning for the **AD** error cases. Additionally, fine-tuning helps to decrease errors in the **PF** category. As expected, it does not significantly reduce errors in the **WD** category.

5 Related Work

Research in **dialect robustness** stems from the need for language technologies to be equitable and not reinforce any negative sentiments against a specific linguistic subgroup (Blodgett et al., 2020). LLMs perform poorly on several downstream tasks (such as the tasks in the GLUE benchmark) involving dialects other than mainstream US English (Joshi et al., 2024; Faisal et al., 2024).

Similar to our work, the evaluation of language understanding ability of LLMs has been explored using typical **conversation understanding tasks** (Chen et al., 2022) like conversation summarisation (Gliwa et al., 2019; Chen et al., 2021), conversation completion (Sai et al., 2020; Ueyama and Kano, 2023), or NLU tasks (Faisal et al., 2024). Other approaches involve conversation-based question-answering tasks that also evaluate the reasoning abilities of LLMs (Sun et al., 2019; Qin et al., 2021). Tasks like mask-filling were used to evaluate LLM-generated responses, more specifically Dey and Desarkar (2023) do so by making RoBERTa predict masked keyword utterances when given a context of dialogue history along with conditions like persona, topic, and facts. Different from standard language understanding tasks, Chalamalasetti et al. (2023) presents a novel method to evaluate the ability of LLMs to act as '*situational*' language understanding agents (Schlangen, 2023). They do so by assigning roles to LLMs and generate dialogues resembling word games such as taboo, and test the language generating and instruction following abilities of LLMs based on the quality of game-play leading to successful target word prediction.

Although we propose a similar approach to evaluation by utilising conversations of such a word game, our work differs from theirs in two ways: (a) they use LLM-generated conversations while we rely on an existing dataset of conversations; (b) they do not employ dialects in their conversations while the dataset we use contains information about the dialects of the human speakers.

6 Conclusion

Although superlative performances have been reported on LLMs in recent times, recent work shows the performance gap between US English and other dialects of English. Our paper presents a first-of-its-kind evaluation of the multilingual LLMs for their robustness to minority language varieties, using their ability to predict target words in game-playing conversations. We use a dataset of target-word-masked conversations between US English speakers and those between Indian English speakers playing a game of taboo. We evaluate pre-trained and fine-tuned versions of one open-source and two closed-source models, on two tasks: target word prediction (TWP) and target word selection (TWS). Our results show that the LLMs indeed perform better for en-US as compared to en-IN on both tasks, with the average performance being higher by 12.66 and 17.4 on similarity and accuracy scores across all configurations. This shows that the LLMs, although multilingual, marginalise or discriminate against speakers of the Indian dialect. We also observe that pre-trained models report a degraded performance on conversations created using both rule-based (en-MV) and LLM-based (en-TR) transformations, as compared to their source conversations (en-US and en-IN respectively). However, fine-tuning on en-MV yields a greater improvement in the task performances, as compared to that on en-TR. This shows that the transformations that introduce dialectal information about a national variety help in improving the dialect robustness of LLMs more than the transformations that remove the said dialectal information. Finally, our error analysis demonstrates that, while most errors are mitigated by providing options for masked target words (TWS; in both pre-trained and fine-tuned variants), multilingual LLMs struggle to interpret target words based on the shared cultural context between speakers.

Our extension M-MD3 is a dataset for TWP and TWS based on MD3, consisting of four subsets: en-US, en-IN, en-MV, and en-TR. The dataset opens opportunities for future evaluations of dialect robustness using similar conversation-based tasks. Our evaluation methodology can also be scaled up and applied to other existing dialogue and discourse datasets, to evaluate the ability of LLMs on properties other than dialect robustness.

Limitations

The original MD3 paper states that their dataset may be dominated by Western entities to some degree. Therefore, it is possible that Indian speakers faced difficulties with the terms. Having said that, the instances selected for our dataset are the ones where the Indian players guessed the word correctly. We have not performed a detailed quali-

tative analysis of these conversations, except for a cursory sanity check. We also assume that the dialect of English from each locale is homogeneous. Assuming that en-IN is the English spoken in every region of India is an unrealistic generalization of the diversity of dialects of English. In terms of model fine-tuning, our paper also does not cover the impact of quantization and different fine-tuning (including cross-dialect) techniques on the task.

Ethics Statement

We use a publicly available dataset of conversations consisting of human players engaged in a game of taboo. The topics discussed in the dataset are fairly general and are unlikely to cause distress. The error analysis was performed by one of the authors of the paper. The AI-transformed (en-TR) conversations may contain biased output, arising due to inherent properties of GPT-based models.

References

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Kingsley Bolton. 2012. World englishes and linguistic landscapes. *World Englishes*, 31(1):30–33.

Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Zhi Chen, Lu Chen, Bei Chen, Libo Qin, Yuncong Liu, Su Zhu, Jian-Guang Lou, and Kai Yu. 2022. UniDU: Towards a unified generative dialogue understanding framework. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 442–455, Edinburgh, UK. Association for Computational Linguistics.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, New Orleans, LA, USA. Curran Associates Inc.

Suvodip Dey and Maunendra Sankar Desarkar. 2023. Dial-M: A masking-based framework for dialogue evaluation. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 77–84, Prague, Czechia. Association for Computational Linguistics.

Jacob Eisenstein, Vinodkumar Prabhakaran, Clara Rivera, Dorottya Demszky, and Devyani Sharma. 2023. Md3: The multi-dialect dataset of dialogues. In *INTERSPEECH 2023*, pages 4059–4063.

Fahim Faisal, Orevaoghene Ahia, Aarohi Srivastava, Kabir Ahuja, David Chiang, Yulia Tsvetkov, and Antonios Anastasopoulos. 2024. Dialectbench: A nlp benchmark for dialects, varieties, and closely-related languages. In *Proceedings of the 2024 Association for Computational Linguistics (ACL)*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu,

Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta,

Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

William Held, Caleb Ziems, and Diyi Yang. 2023. TADA : Task agnostic dialect adapters for English. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 813–824, Toronto, Canada. Association for Computational Linguistics.

Aditya Joshi, Raj Dabre, Diptesh Kanojia, Zhuang Li, Haolan Zhan, Gholamreza Haffari, and Doris Dippold. 2024. Natural language processing for dialects of a language: A survey. *Preprint*, arXiv:2401.05632.

Khyati Khandelwal, Manuel Tonneau, Andrew M. Bean, Hannah Rose Kirk, and Scott A. Hale. 2024. Indianbhed: A dataset for measuring india-centric biases in large language models. In *Proceedings of the 2024 International Conference on Information Technology for Social Good*, GoodIT '24, page 231–239, New York, NY, USA. Association for Computing Machinery.

Yanchen Liu, William Held, and Diyi Yang. 2023. DADA: Dialect adaptation via dynamic aggregation of linguistic rules. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13776–13793, Singapore. Association for Computational Linguistics.

Silvia Dal Negro and Alessandro Vietti. 2006. The interplay of dialect and the standard in anonymous street dialogues: Patterns of variation in northern italy. *Language Variation and Change*, 18(2):179–192.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang,

Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2024. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, New Orleans, LA, USA. Curran Associates Inc.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. TIMEDIAL: Temporal commonsense reasoning in dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7066–7076, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Michael J Ryan, William Held, and Diyi Yang. 2024. Unintended impacts of LLM alignment on global representation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16121–16140, Bangkok, Thailand. Association for Computational Linguistics.

Ananya B. Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M. Khapra. 2020. Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Transactions of the Association for Computational Linguistics*, 8:810–827.

David Schlangen. 2023. On general language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8818–8825, Singapore. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Ayaka Ueyama and Yoshinobu Kano. 2023. Dialogue response generation using completion of omitted predicate arguments based on zero anaphora resolution. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 282–296, Prague, Czechia. Association for Computational Linguistics.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Wikipedia. 2023. Taboo (game).

Zedian Xiao, William Held, Yanchen Liu, and Diyi Yang. 2023. Task-agnostic low-rank adapters for unseen English dialects. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7857–7870, Singapore. Association for Computational Linguistics.

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. *Preprint*, arXiv:2303.18223.

Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta, and Diyi Yang. 2023. Multi-value: A framework for cross-dialectal english nlp. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada. Association for Computational Linguistics.

A Dataset Construction

Table 5 describes the example conversations from extended MD3 and their corresponding masked versions from M-MD3. We mask the turn where the guesser utters the target word to help with formulating our downstream tasks. We mask the target word by finding the exact match in the conversation as shown in the conversations from Table 5. In case of conversations where an exact match is not found (such as *planets*), we find the utterance that is most similar to the target word using the similarity score[11]. The rest of the conversation is then pruned to make the masked target word (represented by '[MASK]') the last token in the conversation.

Target Word	en-IN	Masked en-IN
Fisherman	Describer: Uh. What do you call if we, what will be there in the water? Guesser: Fishes Describer: Who will catch that? Guesser: **Fisherman**.	Describer: Uh. What do you call if we, what will be there in the water? Guesser: Fishes Describer: Who will catch that? Guesser: **[MASK]**
Target Word	**en-US**	**Masked en-US**
Planet	Describer: These are hard words. um Okay. So there's. the Sun and the Moon and all the rest of them. Guesser: And all the **planet**s? (Describer: Yes.)	Describer: These are hard words. um Okay. So there's. the Sun and the Moon and all the rest of them. Guesser: **[MASK]**

Table 5: Masking conversations from the extended MD3 to create M-MD3. The text such as **this** represents the target word utterance by the guesser which is masked (represented by, **[MASK]** in the M-MD3 version of the conversation. The rest of the original conversation is pruned as represented text in parentheses.

B Transformation Issues

We present examples of transformation issues faced while creating en-TR in Table 6. We create en-TR by prompting[12] GPT-4 to remove exaggerations and dialectal information from en-IN conversations. Table 7 presents examples of similar issues faced while creating en-MV using Multi-VALUE. As mentioned in Tables 6 and 7, a *'typical'* transformed conversation maintains the semantic meaning but only differs from the original conversation grammatically. A *'bad'* example deviates largely from the expected output. An *'erroneous'* example is a result of Multi-VALUE not being able to transform a conversation from en-US. Both *'bad'* and *'erroneous'* examples are excluded from the final set of conversations used in our evaluation.

C Errors

Table 8 describes additional examples for all identified error types[13]. As mentioned, each conversation can be classified under multiple error types. For example, the conversation about the target word–*'Ryan Reynolds'* is classified as **CC**, but can also be classified as **PF**.

[11] Described in Section 3.2 of the main paper.
[12] The exact prompt can be found in Section 2 of the main paper.
[13] Defined in Section 4.2 of the main paper.

Type	en-IN	en-TR
Typical	**Describer:** (Uh). What do you call **if we, what will be there** in the water? **Guesser:** Fish(es) **Describer:** Who **will catch that**? **Guesser:** Fisherman.	**Describer:** (∅) What do you call **the creatures** in the water? **Guesser:** Fish(∅). **Describer:** Who **catches them**? **Guesser:** Fishermen.
Bad	**Describer:** There. is a. there is a character in a movie **Guesser:** um **Describer:** It's a very famous movie and it's a very. where is a where you can see famous dialogue called I am still gorgeous **Guesser:** uh. ok. uh **Describer:** character name. compare like Marvel movie **Guesser:** So. uh **Describer:** very muscular body gives scientifically. injections **Guesser:** ok **Describer:** His body will grow **Guesser:** uh Captain America	The character being described is from a well-known movie and is known for the line "I am still gorgeous." This character is similar to those found in Marvel movies and has a very muscular physique, which is achieved through scientific injections that cause his body to grow. The character being guessed is Captain America.

Table 6: Example *transformations* of en-IN to en-TR. We utilise GPT-4 to generate the response. The text in parentheses refers to the omission/removal of certain filler and exaggerated words, and the text such as **this**, refers to the words or sentences that were rephrased to convey the original meaning.

Type	en-US	en-MV
Typical	**Describer:** Perfect. Oh! (We) earn this. We go to our jobs. **Guesser:** Money	**Describer:** Perfect. Oh! (∅) **[are]** earn**[ing]** this. We **[are]** go**[ing]** to our jobs. **Guesser:** Money
Error	**Describer:** This person. is in. oh films. It's a man. He's um. famous for a fine show in the '80s. **Guesser:** Um. what else is he in? **Describer:** He's in a lot of things. I think he's kind of short. Some people think that he looks nice but I don't. Mhm. Mish Mission Impossible. Mission Impossible. **Guesser:** Tom Cruise	None

Table 7: Example *transformations* of en-US to en-MV. We utilise GPT-4 to generate the response. The text in parentheses refers to the omission/removal of words, and the text such as **[this]**, refers to the dialectal features added using Multi-VALUE.

Type	Target Word	Conversation	Model Prediction
AD	Fisherman	Describer: Okay. Okay. A. guy um wants to um okay. Guy catching something in the water. Guesser: [MASK]	Fish
	Mike Tyson	Describer: Big guy that punched people out and he had a little bit of a lisp. Guesser: [MASK]	Darth Wader
WD	Atlantic Ocean	Describer: One of the. of the of world. Guesser: Of the seventh wonder of the world. Taj mahal? Is it regarding sea? Describer: No no no the. Towards the bottom of India. Guesser: Is it regarding Describer: what we have? Guesser: [MASK]	Kanyakumari
	Beg	Describer: Um so if you don't have any money uh you may stand on the corner. Guesser: [MASK]	Panhandle
BDD	Russian Language	Describer: Ok. Ah Largest continent in the world Guesser: Ok. Describer: Ah like area wise. Which country? Guesser: Largest. vast area. vast area? Russia but. Describer: We need to add N over there at the end. Guesser: Russian Describer: We speak Guesser: What they speak? Describer: Yeah. Ok. Guesser: [MASK]	Russian
	Cold War	Describer: This is a two-word term. The first word is a common illness that causes a runny nose. Guesser: Cold. Describer: Yes that's the first word. The second word refers to a conflict between two countries. Guesser: [MASK]	War
CC	Ryan Reynolds	Describer: It is like. One of the. Pen name. which we used in school school days. Guesser: Cello point pen. Fine Grip Describer: No no no Guesser: Reynolds Describer: Uh yeah yeah Guesser: This is a second word or first word. Describer: Yeah this is second word Guesser: First word is. Name Describer:Yeah name related to the same Guesser: [MASK]	Flair
	Mark Wahlberg	Describer: Okay. Um. He was the original. of the Funky Bunch. But then he stopped music. Guesser: [MASK]	Marky
PF	Steve Jobs	Describer: Ok. He is a famous person and he is a. a. for. what we call? Um now it is a. Its. giving competition to Android. what we call? Guesser: ok. so he is the fond ok sorry Describer: he is a founder of so and so company. Its a U. S. company Guesser: so it is giving competition to Android means Google ok.. So Describer: and he is the founder of that company Guesser: [MASK]	Steve
	Kanye West	Describer: All right. Guesser: How do you wanna skip that one Describer: All right. Now um. This guy he um. He just bought a ranch in Wyoming. Guesser: [MASK]	Clint
ERR	Podium	Describer: Okay um. uh. well I isn't sure I'm not sure but uh letting are seeing. Well it's like preacher are churching. I am standing behind this. uh. in in used for speaker. Guesser: [MASK]	Pulpit

Table 8: Example conversations (*'Conversation'*) for each error type (*'Type'*) along with the reference target word (*'Target Word'*) and the generated target word (*'Model Prediction'*). All model predictions are generated using the pre-trained variants of GPT-4 and LLAMA-3.

Cross-Lingual Document Recommendations with Transformer-Based Representations: Evaluating Multilingual Models and Mapping Techniques

Tsegaye Misikir Tashu and Eduard R. Kontos

Matthia Sabatelli and Matias Valdenegro-Toro
Department of Artificial Intelligence
Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence
University of Groningen, Groningen, 9747AG
t.m.tashu.rug.nl,ediraul2001@gmail.com
m.sabatelli@rug.nl, m.a.valdenegro.toro@rug.nl

Abstract

Recommendation systems, for documents, have become tools for finding relevant content on the Web. However, these systems have limitations when it comes to recommending documents in languages different from the query language, which means they might overlook resources in non-native languages. This research focuses on representing documents across languages by using Transformer Leveraged Document Representations (TLDRs) that are mapped to a cross-lingual domain. Four multilingual pre-trained transformer models (mBERT, mT5 XLM RoBERTa, ErnieM) were evaluated using three mapping methods across 20 language pairs representing combinations of five selected languages of the European Union. Metrics like Mate Retrieval Rate and Reciprocal Rank were used to measure the effectiveness of mapped TLDRs compared to non-mapped ones. The results highlight the power of cross-lingual representations achieved through pre-trained transformers and mapping approaches suggesting a promising direction for expanding beyond language connections, between two specific languages.

1 Introduction

The rapid expansion of online information from diverse sources and the growing multilingual nature of the web underscore the escalating significance of information retrieval (IR) and recommender systems (RS). Today's web is no longer limited to a single language, but is increasingly rich in multiple languages, mirroring the multilingual capacities of its global users (Steichen et al., 2014; Tashu et al., 2023). This diversity highlights the urgent need for cross-lingual recommender systems. Traditional recommender systems often prioritize content in a single language, sidelining a wealth of multilingual documents that may hold valuable insights. This gap leads to the emergence of cross-language information access, where recommender systems suggest items in different languages based on user queries (Lops et al., 2010; Narducci et al., 2016; Salamon et al., 2021).

Machine Learning and Deep Learning, which have significantly impacted language representation and processing, are pivotal to enhancing information retrieval and recommender systems, especially in the realm of document recommendation (Tashu et al., 2023; Feng et al., 2022). With these advancements, documents ranging from historical texts and scientific papers to legal ones can be recommended more accurately. However, current recommender systems falter when content is available in various languages, often recommending documents in only the query language. In multinational contexts such as the European Union, such limitations can hinder effective policy formation.

There are two main strategies to address this gap: on the one hand, one can translate the query into multiple target languages or develop a cross-lingual representation space for documents. While this can be effective, this approach is fraught with challenges, including the need for large-scale data, the computational expense of training, and potential loss in translation, especially in domains like law that require precision. On the other hand, cross-lingual representations, which focus on creating shared embedding spaces for documents across languages, are the focal point of this study (Tashu et al., 2023). By employing mapping-aligned document embeddings and comparing their similarity with the query, it offers a computationally cheaper solution without the need for extensive fine-tuning of pre-trained large language models.

The rest of the paper is organized as follows. Section 2 presents the related works. The proposed methodology is presented in section 3. Section 4 presents the experimental setting and the datasets used in this work. The experimental results will be presented in Section 5, while the results are discussed in Section 6. Finally, the conclusions

Proceedings of the 2nd SUMEval Workshop: Scaling Up Multilingual & Multi-Cultural Evaluation, pages 39–47
January 20, 2025.

will be presented in section 7.

2 Related work

The work towards generating inter-lingual and multilingual representations, which can encapsulate information across multiple languages in a unified form, has gained substantial attention in recent years. This interest spans both word-level and document-level representations. Early observations, such as those introduced by (Mikolov et al., 2013), identified that word embedding spaces across languages possess structural similarities. These insights led to the development of linear mappings from one language embedding space to another, utilizing parallel vocabularies. Subsequent works (Lample et al., 2018; Smith et al., 2017; Xing et al., 2015), have aimed to refine these cross-lingual word embeddings, mainly through modifications in space alignment methods or retrieval techniques. Techniques like averaging word vectors (Litschko et al., 2018) or leveraging cross-lingual knowledge bases like Wikipedia (Potthast et al., 2008) or BabelNet (Franco-Salvador et al., 2014) have been used to learn document-level cross-lingual representation. A notable methodology in this domain is the cross-lingual semantic indexing (CL-LSI) (Deerwester et al., 1990; Saad et al., 2014), which extends the well-known latent semantic indexing (LSI) to encapsulate multiple languages through the singular value decomposition of concatenated monolingual document-term matrices.

An emerging strategy in both word-level, sentence-level and document-level research is the use of neural network architectures. One of the pioneer works in this direction was the work by (Schwenk and Douze, 2017) where they used a deep neural network to directly encode long text passages in a language-independent manner. The work by (Artetxe and Schwenk, 2019) used a multilingual auto-encoder to generate language-independent sentence embeddings. Recently, pre-trained models such as BERT (Devlin et al., 2019) have changed the landscape of cross-lingual representation research. These models have enabled the generation of sentence encoders on multilingual unlabeled corpora without the need for parallel data (Conneau et al., 2020; Feng et al., 2022; Goswami et al., 2021; Litschko et al., 2022). Concurrently, certain studies have leveraged pre-trained multilingual transformers for cross-lingual information retrieval (IR). The work by (Shi et al., 2020) combined mBERT with Google Translate in their information retrieval pipeline, while Litschko et al. (2022) utilized mBERT and XLM for the same purpose, emphasizing the need for fine-tuning for efficient and effective document-level results. Collectively, these studies underscore the potential of transformers in cross-lingual information retrieval, paving the way for alternative methodologies such as mapping over fine-tuning, as explored in the current investigation. While these approaches have shown promise, the study herein differentiates itself by presenting a methodology that uses mapping methods to create inter-lingual representations. The novelty of this work primarily lies in the use of mapping methods to align monolingual representations obtained separately for each language from pre-trained large language models, to produce inter-lingual document-level representations.

3 Methods

In this section, we will introduce the different large language models used in this study and the mapping approaches used to learn interlingual representation from the pre-trained large language models.

3.1 Transformers

Transformers, introduced by Vaswani et al. (2017) have transformed the landscape of natural language processing (NLP). Instead of relying heavily on recurrent or convolutional layers, transformers leverage multiple attention heads to weigh the significance of different parts of an input sequence differently, allowing for parallel processing and the capture of long-range dependencies in data. There exist a plethora of variations within the transformer architecture. In the following sections, we will discuss the specific variants of transformer-based large language models used in the context of this study.

3.1.1 mBERT

Multilingual BERT is an extension of the Bidirectional Encoder Representation from Transformers (BERT) that was introduced by Devlin et al. (Devlin et al., 2019). BERT stands out as a pre-trained model, having undergone training on vast volumes of unlabelled data, primarily focusing on two pre-training objectives:

- **Masked Language Modelling (MLM):** This objective requires the model to predict masked

portions of the provided input. Specifically, 15% of the training data tokens undergo masking. Of these masked tokens, 80% are substituted with the "[MASK]" placeholder, 10% are replaced with a random token, and the remaining 10% are left unaltered.

- **Next Sentence Prediction (NSP):** BERT's versatility allows it to manage tasks that involve pairs of sentences, which may or may not exhibit contextual coherence. During its training phase, BERT was supplied with sentence pairs where 50% of the pairs were contextually sequential from the training dataset, while the remaining 50% constituted random, unrelated sentences.

BERT was originally pre-trained on a strictly monolingual English corpus. Recognizing the limitations of such a unilingual approach, there emerged a demand for a model with broader linguistic capabilities. In response, the Multilingual BERT (mBERT) (Devlin et al., 2019), was conceptualized. This iteration extends the foundational principles of BERT, accommodating text from a diverse array of 104 languages.

3.1.2 mT5

Multilingual(Xue et al., 2020) Text-to-Text Transfer Transformer (mT5) is an encoder-decoder model pre-trained on 101 languages, closely based on the original T5 model from (Raffel et al., 2019). It has been pre-trained on an objective similar to MLM, called MLM span-corruption, where consecutive tokens from the input are masked from the model during pre-training.

mT5 is highly specialised for text-to-text tasks such as machine translation and text generation, however, it can also be used as an encoder model only, which was done for this project. Like BERT, the maximum amount of tokens that were used was 512, with an embedding dimensionality of 768, corresponding to the "base" version.

3.1.3 XLM-RoBERTa

The Cross-Lingual Modelling for Robustly Optimised BERT, colloquially termed XLM-RoBERTa, stands as a notable iteration of pre-trained multilingual transformers. Introduced by Conneau et al. (2019), this model is an evolution of RoBERTa (Liu et al., 2019). Diverging from conventional methodologies, XLM-RoBERTa eschews both the Next Sentence Prediction (NSP) and translation objectives, concentrating exclusively on Masked Language Modelling (MLM). The key innovation lies in refining the training procedure and extending the training duration, measures that synergistically enhance model performance. Adapted to cater to 100 languages, XLM-RoBERTa can function effectively as an encoder-only model. For the purposes of this research, the "base" variant of XLM-RoBERTa was deployed, accommodating a maximum of 512 tokens and featuring an embedding dimensionality of 768.

3.1.4 ErnieM

The Multilingual Ernie (ErnieM) (Ouyang et al., 2021) represents a distinguished pre-trained multilingual transformer. Drawing inspiration from the XLM-RoBERTa, ErnieM's hallmark feature lies in its capacity to synchronize linguistic representations across its embedded languages. This harmonization is operationalized through a cross-lingual semantic alignment, juxtaposing parallel data with its monolingual counterpart. In the spirit of achieving this, the authors put forth two pre-training objectives:

- Cross-Attention MLM (CAMLM): A strategy devised to cohesively align the semantic representation of parallel data across the entire linguistic spectrum.

- Back-Translation MLM (BTMLM): This objective embarks on aligning cross-lingual semantics with monolingual contexts. Through back-translation, it facilitates the generation of novel linguistic tokens from monolingual corpora, and subsequently acquaints the model with their multilingual semantic alignment.

Supplemented by the translation modelling language task (an initiative akin to MLM but marked by the amalgamation of sequences from an array of languages) and the Multilingual MLM (characterized by masking tokens transcending diverse languages), these objectives jointly constitute the pre-training paradigm of ErnieM. Maintaining consistency, this study harnesses the "base" version of ErnieM, with a stipulated threshold of 512 tokens and an embedding dimensionality set at 768.

The models selected for this investigation inherently embrace a multilingual ethos, underpinned by two pivotal reasons: Firstly, the monolingual iterations of these models have not ubiquitously

undergone training across the selected quintet of languages earmarked for this research. More critically, the inherent overlap in the models' embedding space across languages posits a fertile ground to evaluate the potential of leveraging ready-made multilingual models sans the requisite of supplementary mapping or precision-tuning. To draw an illustrative parallel, juxtaposing disparate models of analogous frameworks, each tailored to individual languages (e.g., BERT vis-à-vis its Gallic analogue), might yield embeddings that, owing to divergent training trajectories, manifest disparities too profound to be semantically reconciled.

3.2 Mapping approaches

Given two monolingual document collections, $D_x = \{d_{x,1}, \ldots, d_{x,n}\}$ in language x and $D_y = \{d_{y,1}, \ldots, d_{y,n}\}$ in language y . To embark on a nuanced analysis of these documents, it is imperative first to learn or extract the embedding for each document. To achieve this, we employ the pre-trained large language models introduced in section 3 subsection 3.1. Notwithstanding, it's worth noting that any representation learning algorithm that embeds the document sets D_x and D_y into vectors within the space $\mathbb{R}^k$ can be used.

From the language models, we obtain sets of vectors, respectively, defined as $C_x = \{\hat{d}_{x,1}, \ldots, \hat{d}_{x,n}\} \subset \mathbb{R}^k$ and $C_y = \{\hat{d}_{y,1}, \ldots, \hat{d}_{y,n}\} \subset \mathbb{R}^k$. Conceptually, C_x and C_y can be interpreted as "Conceptual Vector Spaces", encapsulating broader linguistic and thematic abstractions inherent to the original documents. Nevertheless, a salient point to recognize is that even if vectors within C_x and C_y encapsulate analogous concepts transversal to languages, the representation schema might vary. Consequently, a mere direct juxtaposition of $\hat{d}_{x,k}$ and $\hat{d}_{y,k}$might not manifest the underlying content congruencies.

All the mapping methods used in this study are adopted from the works of Tashu et al.(Tashu et al., 2023). In the upcoming section, we will present a summary of three different mappings where more details on each of the methods can be found in (Lenz et al., 2021; Tashu et al., 2023).

3.2.1 Linear concept approximation (LCA)

The motivation is to directly embed the test documents into the space spanned by the training documents in the semantic space using linear least squares (Salamon et al., 2021). This is based on the assumption that the vector space spanned by the parallel training documents is the same in their respective language. Therefore, the coordinates of the test documents in that span would be a good language-independent representation of these documents. Using the representation obtained from the large language models presented in section 3, we can derive low-dimensional representations of each document within $\mathbb{R}^k$. Multiple documents can be concatenated into matrices. If there are n documents available in both languages, we can create the representation/concept matrices $C_x = X^T \in \mathbb{R}^{n \times k}$ and $C_y = Y^T \in \mathbb{R}^{n \times k}$ in which every column is a concept in its respective language.

3.2.2 Linear Concept Compression (LCC)

The motivation behind LCC is to find mappings into an inter-lingual space, EC_x, C_y, such that the comparison of $C_x(\hat{d}_{x,k}), C_y(\hat{d}_{y,k})$, provides a measure of content similarity. For two monolingual representations, we want to find their inter-lingual representations, which encode the same information as the different monolingual spaces do. More precisely, for a given document d and its representations in each respective language, $\hat{d}_{x,k}$ and $\hat{d}_{y,k}$, we want to find mappings C_x and C_y, respectively, such that $C_x(\hat{d}_{x,k}) = C_y(\hat{d}_{y,k})$ and the information of $\hat{d}_{x,k}$ and $\hat{d}_{y,k}$ is preserved. The intuition is to train an Encoder-Decoder approach. The purpose of the Encoder is to encode monolingual representations in a language-independent space. The purpose of the Decoder is to reconstruct the monolingual representations of multiple languages from that encoding(Lenz, 2021).

3.2.3 Neural Concept Approximation (NCA)

In contrast to conventional approaches where mappings are directly derived from given vectors, C_x and C_y, the proposed methodology leverages a Neural Network to approximate these vectors. Specifically, a Feed Forward Neural Network (FFNN). Two distinct models were trained: one mapping from the source language to the target language, and the other in the reverse direction (Tashu et al., 2023).

Both models were defined in the same manner: 1 layer of 500 neurons, using the Exponential Linear Unit (ELU), with the Huber objective function, for a maximum of 250 epochs with the implementation of early stopping and a learning rate of $5 \cdot 10^{-4}$. The network's architecture consists of 3 total layers, one

input layer with dimensionality d (the dimension of a given document), followed by the hidden layer (with dimensionality $d \times 500$), and the output layer with dimensionality $500 \times d$.

4 Experiment

4.1 Data

The JRC-Acquis corpus (Steinberger et al., 2006) was used for this project because of its characteristics. It is a publicly available, sentence-aligned corpus consisting of the 22 official languages of the European Union (EU), containing legal documents pertaining to EU matters from 1958 to 2006. Since this study dealt with language pairs, only five languages were used, those being English, Romanian, Dutch, German, and French, for a total of 20 ordered pairs (i.e. English $\rightarrow$ French and French $\rightarrow$ English are treated as a different pair). Since the documents for each language were not aligned, it was necessary to perform a secondary alignment for the five chosen languages such that documents were shared across the subset, resulting in $6,538$ unique documents. There were also some issues at the character level of some non-English documents from the initial dataset. For instseveralber of French documents presented corrupted letters, meaning that letters with diacritics were instead displayed in XML format (e.g. "é" displayed as "%eacute"). A preprocessing step was as such introduced to replace these corrupted variants with their original form and to remove any additional white space from the documents. The documents, at the same time, were converted from XML to a standard string format to be used by the models. In this study, 60% was used for the training set, 20% for the validation set and 20% for the test set.

4.2 Embeddings

It is necessary to represent the documents in a continuous manner to be able to apply any mapping approach. This was achieved by passing all documents, in each language, through the tokenizer and model modules of the previously discussed transformer models.

An input text undergoes several processing steps while passing through the tokenizer: it is truncated or padded to the maximum length allowed by the models ($N = 512$ tokens), after which the tokens are converted to internal ID representations stored in the vocabulary of the model, and for which the attention mask is computed. The latter part allows the model to look only at the relevant tokens in the sequence, ignoring padding tokens. Since this study only deals with the embeddings of the models and not their decoded outputs, the final hidden state from the encoder part of the models is extracted. The model computed the embedding for each token, and as such, documents are now represented as 512×768 matrices, while it is necessary to obtain a vector of size 768. This was solved by performing a global pooling operation on all of the outputted states, where tokens that were not ignored by the attention mask were averaged together. As such, documents are now represented by vectors with dimensionality 768, to be used in the following section.

4.3 Evaluation metrics

Two evaluation metrics were used to compute the performance of the mapping approaches:

- Mate Retrieval Rate: the retrieval rate of the most symmetric document; this metric evaluates how similar two documents are - the query and retrieved document. If the retrieved document is the same as the query document, that is called a mate retrieval. It is defined as:

$$MR(d) = \arg\max \mathbf{S}_d \cdot \mathbf{T}_d^T$$
$$S(d, d') = \begin{cases} 1 & d = d' \\ 0 & d \neq d' \end{cases} \quad (1)$$

  where S is the similarity between 2 documents d and d', and MR is the mate retrieval for a given document d in the source S and target language T. It can be said that a mate retrieval is successful if d and d' are the same. The equations in 1 can be combined to compute the mate retrieval rate for all documents (D), as seen in equation 2:

$$\text{RetrievalRate} = \frac{1}{|D|} \sum_{d=1}^{|D|} S(d, MR(d)) \quad (2)$$

- Mean Reciprocal Rank: this represents how high-ranked documents are, based on a similarity measure. This has been achieved using cosine similarity, defined below:

$$C(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \cdot \|d_2\|} \quad (3)$$

where the numerator represents the inner product of the vector representations of documents d_1 and d_2, and the denominator is the magnitude product of the two vectors. If the two documents are similar to each other, their cosine will be closer to 1 and will be closer to -1 if they are not similar. This equation can be used to obtain the cosine matrix similarity of all documents.

Furthermore, the rank r of a document can be defined as its cosine similarity compared to other documents obtained from the matrix cosine similarity. If it is most similar to itself in the target language, then its rank will be 1. Finally, these components can be combined to form the mean reciprocal rank:

$$\text{ReciprocalRank} = \frac{1}{|D|}\sum_{d=1}^{|D|}\frac{1}{r_d} \quad (4)$$

5 Results

The performance of the mapped (or not) embeddings was measured using the evaluation metrics defined in the previous section. Due to the large number of results that were obtained (640 total results across four transformer models, three mapping methods and no mapping, for 20 language pairs, for each evaluation metric), the final results have been averaged across models and language pairs. As such, Figures 1 and 2 only present their average evaluation metric for all dimensions. Both figures showcase significant results when comparing mapped and non-mapped embeddings. However, there is also a significant difference between embeddings mapped using NCA and embeddings mapped with the other methods.

The best mapping method across both evaluation metrics was LCA (Retrieval Rate $= 0.937$, Reciprocal Rank $= 0.958$), while the worst mapping method was NCA (Retrieval Rate $= 0.609$, Reciprocal Rank $= 0.696$). Still, all methods performed significantly better than the non-mapped embeddings (Retrieval Rate $= 0.201$, Reciprocal Rank $= 0.279$). Table 1 presents the results across all language pairs for both metrics, broken down for each transformer model and mapping method, and additionally the results obtained by Tashu et al. (2023). Using the same mapping approaches, mBERT embeddings mapped using LCA outperform all other models and mapping combinations, including those from the mentioned paper, across both metrics (RetrievalRate $= 0.963$, ReciprocalRank $= 0.975$).

Model	Mapping	MRRank	MRtRate
mBERT	None	0.2	0.115
	LCA*	**0.975**	**0.963**
	LCC	0.973	0.959
	NCA	0.84	0.781
mT5	None	0.466	0.37
	LCA*	**0.947**	**0.922**
	LCC	0.936	0.907
	NCA	0.814	0.756
XLM-RoBERTa	None	0.114	0.057
	LCA	0.948	0.925
	LCC*	**0.951**	**0.928**
	NCA	0.617	0.499
ErnieM	None	0.443	0.355
	LCA*	**0.965**	**0.949**
	LCC	0.962	0.946
	NCA	0.742	0.67

Table 1: Mean Reciprocal Rank (MRRank) and Mate Retrieval Rate (MRtRate).

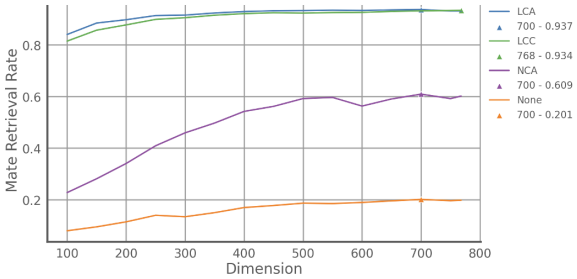


Figure 1: Line plot of the average Mate Retrieval Rate across dimensions for all language pairs and models, using LCA, LCC, NCA, and no mapping.

6 Discussion

From our results, it becomes evident that Transformer Leveraged embeddings combined with mapping methods markedly outperform non-mapped embeddings across all models, as delineated in Table 1. These Leveraged embeddings, in all instances, show significant superiority compared to the non-mapped variants. This underscores that employing an off-the-shelf model devoid of enhancements (e.g., fine-tuning, mapping) results in subpar outcomes, irrespective of the model's type. Figures 1 and 2 further substantiate this, demonstrating that mapped embeddings consistently outpace their non-mapped counterparts across all metrics. Within this context, the NCA mapping method displayed

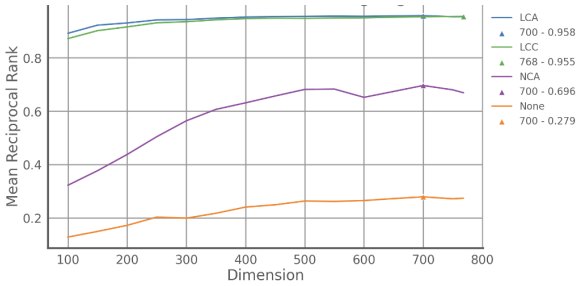


Figure 2: Line plot of the average Mean Reciprocal Rank across dimensions for all language pairs and models, using LCA, LCC, NCA, and no mapping.

the favourableorable performance, overshadowing only the non-mapped embeddings. This could be attributable to the network's architectural design, potentially falling short in capturing the nuanced similarities between documents to establish an effective mapping.

An examination of Table 1 reveals mBERT's dominance over other transformer models across all mapping strategies. Notably, when paired with LCA and LCC-mapped embeddings, mBERT eclipsed all other embedding and mapping combinations as referenced by (Tashu et al., 2023). This superior performance may be credited to the extensive data mBERT trained on, complemented by its pre-training tasks.

Interestingly, both ErnieM and mT5, when aligned with non-mapped embeddings, showcased better performance than their transformer counterparts under identical conditions. The underlying reason might be traced back to the distinctive training data and methodologies employed by these models. In contrast to mBERT and XLM-RoBERT, which utilize MLM (and additionally NSP for mBERT), ErnieM incorporates a broader spectrum of pre-training objectives geared towards cross-lingual alignment. This distinction could elucidate the superior performance of its non-mapped embeddings. mT5's commendable performance can be attributed to its foundational design, being inherently an encoder-decoder model, though this project exclusively utilized its encoding facet.

In general, our study highlights the efficacy of Transformer Leveraged embeddings when synergized with mapping techniques, resulting in a noticeable performance leap over non-mapped embeddings. This aligns with the findings of (Litschko et al., 2022), which accentuate that standalone out-of-the-box models, without refinements or supplementary techniques, are generally less efficient. However, diverging from their study, our research underscores that optimal performance doesn't solely hinge on model fine-tuning. In the realm of IR, integrating mapping techniques can be equally potent in driving commendable results.

7 Conclusion

Document recommendation stands at the forefront of Information Retrieval (IR) systems. Within recommendation frameworks, it efficiently suggests pertinent documents in alignment with a user's query. In our research, we delved into the possibilities of crafting cross-lingual representations by harnessing embeddings from pre-existing multilingual transformers in conjunction with mapping strategies. Using embeddings from these pre-trained multilingual transformers allows for document representation without requiring further training or intricate processing. Nonetheless, our research illuminated that solely depending on the raw embeddings from the transformers fell short in terms of efficacy. A notable enhancement in results was witnessed when the embeddings were synergized with mapping techniques such as LCA, LCC, and NCA. The languages incorporated within our study hold considerable prominence across various linguistic tasks. Consequently, the adopted models and mapping techniques have the potential to foster efficient representations by mapping low-resource languages onto those that are more abundantly represented. It beckons further exploration into how these mapping techniques perform when applied to low-resource languages. Future research might not restrict itself to merely language pairs, as was the focus of this study, but could expand to encompass language tuples—translating from a single source language to multiple target languages. Achieving this might necessitate refining the present mapping methodologies, introducing supplementary steps, or pioneering entirely novel methods. The code of this project is publicly available on GitHub.

References

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

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.

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)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Marc Franco-Salvador, Paolo Rosso, and Roberto Navigli. 2014. A knowledge-based representation for cross-language document retrieval and categorization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 414–423, Gothenburg, Sweden. Association for Computational Linguistics.

Koustava Goswami, Sourav Dutta, Haytham Assem, Theodorus Fransen, and John P. McCrae. 2021. Cross-lingual sentence embedding using multi-task learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9099–9113, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*, pages 1–14.

Marc Lenz. 2021. Learning multilingual document representations.

Marc Lenz, Tsegaye Misikir Tashu, and Tomáš Horváth. 2021. Learning inter-lingual document representations via concept compression. In *Intelligent Data Engineering and Automated Learning – IDEAL 2021*, pages 268–276, Cham. Springer International Publishing.

Robert Litschko, Goran Glavaš, Simone Paolo Ponzetto, and Ivan Vulić. 2018. Unsupervised cross-lingual information retrieval using monolingual data only. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1253–1256, New York, NY, USA. Association for Computing Machinery.

Robert Litschko, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2022. On cross-lingual retrieval with multilingual text encoders. *Inf. Retr.*, 25(2):149–183.

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.

Pasquale Lops, Cataldo Musto, Fedelucio Narducci, Marco De Gemmis, Pierpaolo Basile, and Giovanni Semeraro. 2010. Mars: A MultilAnguage Recommender System. In *Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, HetRec '10, page 24–31, New York, NY, USA. ACM.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Fedelucio Narducci, Pierpaolo Basile, Cataldo Musto, Pasquale Lops, Annalina Caputo, Marco de Gemmis, Leo Iaquinta, and Giovanni Semeraro. 2016. Concept-based item representations for a cross-lingual content-based recommendation process. *Information Sciences*, 374:15–31.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Martin Potthast, Benno Stein, and Maik Anderka. 2008. A wikipedia-based multilingual retrieval model. In *Advances in Information Retrieval*, pages 522–530, Berlin, Heidelberg. Springer Berlin Heidelberg.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Motaz Saad, David Langlois, and Kamel Smaïli. 2014. Cross-lingual semantic similarity measure for comparable articles. In *International Conference on Natural Language Processing*, pages 105–115. Springer.

Vilmos Tibor Salamon, Tsegaye Misikir Tashu, and Tomáš Horváth. 2021. Linear concept approximation for multilingual document recommendation. In *Intelligent Data Engineering and Automated Learning – IDEAL 2021: 22nd International Conference, IDEAL 2021, Manchester, UK, November 25–27, 2021, Proceedings*, page 147–156, Berlin, Heidelberg. Springer-Verlag.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Peng Shi, He Bai, and Jimmy Lin. 2020. Cross-Lingual Training of Neural Models for Document Ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2768–2773, Online. Association for Computational Linguistics.

Samuel L Smith, David HP Turban, Steven Hamblin, and Nils Y Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *arXiv:1702.03859*.

Ben Steichen, M. Rami Ghorab, Alexander O'Connor, Séamus Lawless, and Vincent Wade. 2014. Towards personalized multilingual information access - exploring the browsing and search behavior of multilingual users. In *User Modeling, Adaptation, and Personalization*, pages 435–446, Cham. Springer International Publishing.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. 2006. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *CoRR*, abs/cs/0609058.

Tsegaye Misikir Tashu, Marc Lenz, and Tomáš Horváth. 2023. NCC: Neural concept compression for multilingual document recommendation. *Applied Soft Computing*, 142:110348.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934.

VRCP: Vocabulary Replacement Continued Pretraining for Efficient Multilingual Language Models

Yuta Nozaki*, **Dai Nakashima***, **Ryo Sato***,
Naoki Asaba*, **Shintaro Kawamura***
*Ricoh Company, Ltd.
{yuta.nozaki1,dai.nakashima,ryo.sato4,
naoki.asaba,shintaro.kawamura}@jp.ricoh.com

Abstract

Building large language models (LLMs) for non-English languages involves leveraging extensively trained English models through continued pre-training on the target language corpora. This approach harnesses the rich semantic knowledge embedded in English models, allowing superior performance compared to training from scratch. However, tokenizers not optimized for the target language may make inefficiencies in training. We propose Vocabulary Replacement Continued Pretraining (VRCP), a method that optimizes the tokenizer for the target language by replacing unique (solely available) vocabulary from the source tokenizer while maintaining the overall vocabulary size. This approach preserves the semantic knowledge of the source model while enhancing token efficiency and performance for the target language. We evaluated VRCP using the Llama-2 model on Japanese and Chinese corpora. The results show that VRCP matches the performance of vocabulary expansion methods on benchmarks and achieves superior performance in summarization tasks. Additionally, VRCP provides an optimized tokenizer that balances token efficiency, task performance, and GPU memory footprint, making it particularly suitable for resource-constrained environments.

1 Introduction

Recent advancements in large language models (LLMs) based on transformer architectures (Vaswani et al., 2017) have brought significant progress to the field of NLP. Models such as GPT-4 (OpenAI et al., 2023) and Llama-2 (Touvron et al., 2023) have predominantly been trained on extensive English corpora, leaving a gap in the availability of models optimized for non-English languages. This disparity is due to the relative scarcity of high-quality, large-scale corpora for many non-English languages compared to English. Consequently, this limits the potential improvements based on the scaling laws (Kaplan et al., 2020) of language models for these languages.

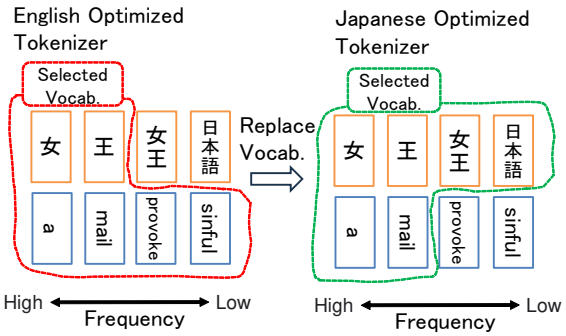


Figure 1: Illustration of VRCP: The English Optimized Tokenizer (left) replaces low-frequency English vocabulary with high-frequency Japanese vocabulary to create the Japanese Optimized Tokenizer (right). This retains common vocabulary while optimizing for Japanese.

One promising approach to developing models specialized for non-English languages involves continued pretraining of extensively pretrained models on the target language corpora (Yong et al., 2023; Wang et al., 2020; Pfeiffer et al., 2021). This approach leverages the semantic knowledge and intelligence of the English-based models, enabling high performance even with relatively small amounts of target language corpora. This approach, akin to DAPT (domain-adaptive pre-training) (Gururangan et al., 2020), allows for the incorporation of new linguistic characteristics while retaining the established knowledge base.

However, this approach presents significant challenges. The vocabulary of tokenizers used in English-based model is optimized for efficient tokenizing of English texts. When applied to texts in target languages, these tokenizers often segment the text into excessively small units. This leads to a considerable reduction in the length of sequences that can be processed in the same batch, thereby increasing the training time for the target

Proceedings of the 2nd SUMEval Workshop: Scaling Up Multilingual & Multi-Cultural Evaluation, pages 48–59
January 20, 2025.

language data and significantly decreasing training efficiency. This inefficiency is particularly problematic in scenarios involving Retrieval-Augmented Generation (RAG), where language models need to process long texts as prompts for seq2seq tasks, e.g., summarization (Lewis et al., 2021). Given the inherent limitations in sequence length imposed by transformer architectures, using a tokenizer that inefficiently for the target language restricts the amount of text that can be included in the prompt. Even when a model supports very long sequence lengths, the computational complexity increases at an $O(n^2)$ rate with the sequence length (Vaswani et al., 2017), leading to poor computational efficiency.

Additionally, languages with non-alphabetic scripts, such as Japanese and Chinese, often encounter issues with frequent out-of-vocabulary words or characters being fragmented at the byte level (Rust et al., 2021).

To address these inefficiencies, a previous approach is to expand the tokenizer's vocabulary with tokens relevant to the target language, reducing the number of tokens needed to represent the same text (Wang et al., 2020; Yao et al., 2021; Liu et al., 2020; Wang et al., 2019; Minixhofer et al., 2022). However, this method inadvertently increases the parameter count of the embedding layer of the model and, consequently, a lager memory footprint.

Maintaining a constant vocabulary size while optimizing for target languages is crucial for preventing an increase in model parameters and controlling memory consumption. This is particularly important for smaller models, where the embedding layer represents a significant portion of the overall model size. For instance, in models like Qwen-2 (0.5B and 1.5B), embedding tying is used to prevent an increase in model size (Yang et al., 2024). Moreover, works on machine translation models have shown that embedding tying can reduce model size while maintaining performance (Press and Wolf, 2017). Additionally, increasing the size of the embedding layer exacerbates communication overhead during distributed training, as highlighted in studies by (Acun et al.). Maintaining a constant vocabulary size allows us to mitigate these issues and improve training efficiency.

We propose a novel method, Vocabulary Replacement Continued Pretraining (VRCP), to enhance token efficiency for the target language in continued pretraining while maintaining the vocabulary size. Our method involves constructing a new tokenizer tailored to the target language and substituting unique (solely available) vocabulary from the source tokenizer with vocabulary from the target language. This enables continued pretraining that leverages the semantic knowledge of the existing model while improving token efficiency. As illustrated in Figure 1, VRCP is a simple method whereby low-frequency words from the target language corpus within the source tokenizer are replaced with high-frequency words from the same target language corpus.

We evaluated VRCP through experiments on Japanese and Chinese texts. The results demonstrated that VRPC matches the token efficiency and task performance of expanding vocabulary methods. By not increasing the model size, VRCP also prevents any additional GPU memory footprint. Notably, for summarization tasks, VRCP showed superior performance compared to previous methods. This indicates its suitability for use cases such as RAG.

Our main contributions are as follows:

- We propose a method, VRCP, to enhance token efficiency for the target language in continued pretraining while maintaining the vocabulary size.
- Through experiments with Japanese and Chinese, we demonstrated that VRCP can achieve token efficiency and task performance comparable to vocabulary expanding methods while maintain vocabulary size.
- We showed that VRCP improves summarization task performance, making it ideal for use cases such as RAG.

2 Formulization

2.1 Tokenizer Definition

A tokenizer performs two primary tasks: segmenting a text into tokens and mapping these tokens to unique IDs using its vocabulary V.

First, the segmentation function S processes a text $text$ and produces a sequence of tokens $\{t_1, t_2, \ldots, t_n\}$, where each token t_i belongs to V:

$$S(text, V) = \{t_1, t_2, \ldots, t_n\} \quad (1)$$

Then, the mapping function M assigns each token t_i an ID id_i within the range $\{0, 1, \ldots, |V| - 1\}$:

$$M(t_i, V) = id_i \quad (2)$$

2.2 Embedding Definition

Each token ID id_i is associated with an embedding vector from the embedding matrix E. This matrix E is of size $|V| \times d$, where $|V|$ is the vocabulary size and d is the dimensionality of the embedding vectors. For a token t_i, its embedding vector $\vec{e}_{id_i}$ is obtained by mapping t_i to its ID id_i using M, and then retrieving the corresponding vector from E:

$$\vec{e}_{id_i} = E[M(t_i, V)] \tag{3}$$

2.3 Vocabulary Expansion Method

To expand the vocabulary of a source tokenizer, a new vocabulary V' is constructed from the target language corpus. This new vocabulary V' is combined with the original vocabulary V to form an expanded vocabulary:

$$V_{\text{new}} = V \cup V' \tag{4}$$

For each new token v'_i in V', its embedding $\vec{e}_{id_{v'_i}}$ is computed by taking the arithmetic mean of the embeddings of its segmented subtokens. Specifically, if v'_i is segmented into $\{t_1, t_2, \ldots, t_n\}$, we first map these tokens to their IDs $\{id_1, id_2, \ldots, id_n\}$ using M, and then retrieve their embeddings $\vec{e}_{id_1}, \vec{e}_{id_2}, \ldots, \vec{e}_{id_n}$ from E. The new embedding vector $\vec{e}_{id_{v'_i}}$ is calculated as the arithmetic mean of these vectors:

$$\vec{e}_{id_{v'_i}} = \frac{1}{n} \sum_{j=1}^{n} \vec{e}_{id_j} \tag{5}$$

These new embeddings are then added to the original embedding matrix E, resulting in an updated matrix:

$$E' = \begin{cases} \vec{e}_{id} & \text{if } id \in V \\ \vec{e}_{id_{v'_i}} & \text{if } v'_i \in V' \end{cases} \tag{6}$$

3 Proposed Method: VRCP

Our proposed method, Vocabulary Replacement Continued Pretraining (VRCP), consists of four main components: Vocabulary Construction, Vocabulary Replacement, Embedding Replacement, and Continued Pretraining.

3.1 Vocabulary Construction

The first step of VRCP is to construct a new vocabulary specialized for the target language. We define the vocabulary size to be equal to that of the source tokenizer and develop the tokenizer using a corpus that combines both the target language and English. Including English corpus ensures that common vocabulary between the target language and English is retained, which helps in effectively utilizing the semantic knowledge of the source model.

3.2 Vocabulary Replacement

Next, we replace the unique vocabulary of the source vocabulary V with those from the constructed vocabulary V'. This process retains the token ID mappings for the common vocabulary between V and V', enabling the model to leverage its knowledge of source model effectively.

The process is carried out using the following steps and equations:

1. **Identifying Common Vocabulary**:

   Identify the common vocabulary between the source vocabulary V and the constructed vocabulary V' by taking their intersection:

   $$V_{\text{com}} = V \cap V' \tag{7}$$

2. **Identifying Unique Vocabulary**:

   Determine the unique vocabulary in V' that are not present in V by taking the difference:

   $$V'_{\text{uni}} = V' \setminus V \tag{8}$$

3. **Constructing the New Vocabulary**:

   Form the new vocabulary V_{new} by combining the common vocabulary V_{com} and the unique vocabulary from V', V'_{uni}:

   $$V_{\text{new}} = V_{\text{com}} \cup V'_{\text{uni}} \tag{9}$$

To preserve the integrity of token IDs, for any vocabulary $v_{\text{com}} \in V_{\text{com}}$, the token ID in the new vocabulary V_{new} remains the same:

$$M(v_{\text{com}}, V_{\text{new}}) = M(v_{\text{com}}, V) \tag{10}$$

This equality holds because v_{com} is a part of the common vocabulary V_{com} and thus its token ID does not change with the new vocabulary V_{new}. By preserving these mappings, the model retains its knowledge associated with the shared tokens, enabling effective utilization of existing knowledge while adapting to new language nuances.

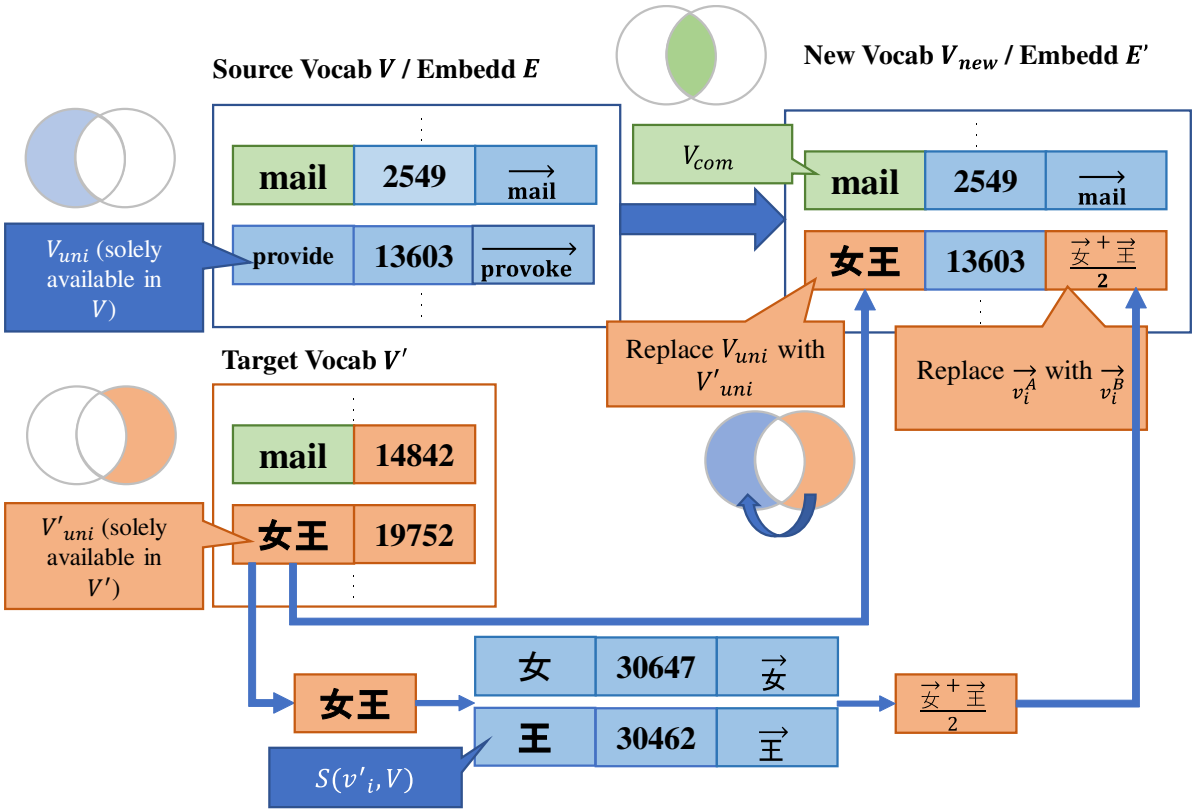


Figure 2: VRCP Process: Source vocabulary V and target vocabulary V' are used to create a new vocabulary V_{new}. Common tokens V_{com} are retained, while unique tokens V_{uni} from V are replaced with unique tokens V'_{uni} from V'.

3.3 Embedding Replacement

Replacing the vocabulary alone does not ensure that the embedding vectors for the unique vocabulary V' (V'_{uni}) align with the embeddings E. This is because the unique vocabulary $v'_i \in V'_{\text{uni}}$ have embedding vectors $\vec{e}_{id_{v'_i}}$ that may not fit well with the source embedding space. To address this, we need to replace these embeddings to maintain semantic consistency. Specifically, the following relation holds:

1. **Token Segmentation and ID Mapping**:

   For each token v'_i in the constructed vocabulary V', use the source tokenizer S and the vocabulary V to segment v'_i and map it to a sequence of token IDs $\{id_1, id_2, \ldots, id_n\}$:

   $$\{id_1, id_2, \ldots, id_n\} = M(S(v'_i, V), V) \tag{11}$$

   where each id_j corresponds to the token t_j in the vocabulary V.

2. **Retrieving Source Embedding Vectors**:

   Retrieve the corresponding embedding vectors $\vec{e}_{id_j}$ from the source embedding matrix E for each token ID id_j:

   $$\vec{e}_{id_j} = E[id_j] \tag{12}$$

3. **Calculating the Arithmetic Mean Embedding Vector**:

   Compute the arithmetic mean of the embedding vectors $\vec{e}_{id_j}$ for all sub-tokens t_j and create the embedding vector $\vec{e}_{id_{v'_i}}$ for the constructed vocabulary v'_i:

   $$\vec{e}_{id_{v'_i}} = \frac{1}{n} \sum_{j=1}^{n} \vec{e}_{id_j} \tag{13}$$

4. **Updating the Embedding Matrix**:

   After calculating the embedding vectors $\vec{e}_{id_{v'_i}}$ for the constructed vocabulary V', update the source embedding matrix E to form the new embedding matrix E':

   $$E' = \begin{cases} \vec{e}_{id} & \text{if } id \in V \\ \vec{e}_{id_{v'_i}} & \text{if } v'_i \in V'_{\text{uni}} \end{cases} \tag{14}$$

   This update ensures that the new embedding matrix E' remains semantically consistent with the existing embedding matrix E.

3.4 Continued Pretraining

In VRCP, the continued pretraining process primarily utilizes the target language corpus. However, to enhance the stability and effectiveness of training, we include a small portion of English corpus, aligning with the Domain-Adaptive Pretraining (DAPT) (Gururangan et al., 2020) strategy.

This inclusion follows approach of DAPT, where maintaining a minor amount of English data helps to mitigate the risk of catastrophic forgetting and prevents abrupt changes in the data distribution from destabilizing the model adaptation process.

4 Experiments

We evaluated the effectiveness of our proposed method using experiments with Japanese and Chinese corpora.

We evaluated the methods based on three key aspects:

- **Token Efficiently (Target/English)**
  - We evaluated each method by tokenizing the test data and measuring the average character length per token (Length Per Token, LPT) of the resulting token sequence.
- **Task performance (Target/English)**
  - We evaluated the models trained with each tokenizer method on benchmark tasks to assess their performance.
- **GPU Memory Footprint**
  - We measured the average GPU memory footprint per device in training.

4.1 Setup

Preparation of Corpora: We prepared mixed corpora of the target language and English for both Japanese and Chinese, ensuring a balanced ratio of 6:5 between the target language and English. The details of the corpora used are as follows:

- **Japanese**: Wikipedia (Japanese), CC100 Conneau et al. (2020) (Japanese), Wikipedia (English)
- **Chinese**: NLP Chinese Corpus (Xu, 2019) (Baike, News, Wikipedia), Wikipedia (English)

Construction of Tokenizer: We built a tokenizer with 32,000 words (V') using SentencePiece (Kudo and Richardson, 2018) and BPE (character_coverage=0.9995, Byte Fallback=True) from the mixed corpus of the target language and English, prepared as described in the previous section. Subsequently, we applied VRCP to replace the unique vocabulary (V_{uni}) from the Llama-2 tokenizer with the unique vocabulary (V'_{uni}) derived from the vocabulary constructed using this mixed corpus.

To evaluate the effectiveness of VRCP, we compared it against three other tokenizer approaches:

- $V \cup V'$ ($|V'| = 16k$): In this approach, we expanded the Llama-2 tokenizer (V) by adding 16,000 vocabulary constructed exclusively from a corpus in the target language.
- $V \cup V'$ ($|V'| = 32k$): In this approach, we expanded the Llama-2 tokenizer (V) by adding 32,000 vocabulary constructed exclusively from a corpus in the target language, similar to the previous approach.
- **Unchanged Llama-2 tokenizer** (V): In this approach, we used the Llama-2 tokenizer as-is, without any replacements or additions.

Continued Pretraining: We conducted further pretraining of Llama-2-7B using a mixed corpus consisting of both the target language and English. For this process, we used the same types of corpora as those used for the construction of Tokenizer, but with a different sampling strategy. Specifically, the ratio of English data in the training dataset was adjusted to under 5%. We describe the detailed training settings in the appendix.

4.1.1 Result and Discussion on Token Efficiency

VRCP significantly improved token efficiency for both Japanese and Chinese compared to the Llama-2 tokenizer, achieving approximately 2.2 times better efficiency for Japanese and 1.8 times better for Chinese. This is a major achievement of our study, particularly because we impose a constraint of not expanding the vocabulary size (see Tables 1 and 2 for detailed efficiency metrics).

When comparing VRCP to the vocabulary expansion method with $|V'| = 16,000$ and $|V'| = 32,000$, we find that VRCP includes more target language vocabulary. Although VRCP shows

Table 1: Tokenization Efficiency for Japanese

Methods	Vocab Size	JA Vocab Size	Common Vocab Size	EN LPT	JA LPT
$V_{\text{com}} \cup V'_{\text{uni}}$ (VRCP)	32,000	15,293	12,137	3.740	1.838
$V \cup V'$ ($\|V'\| = 16k$)	46,312	14,553	32,000	3.738	1.883
$V \cup V'$ ($\|V'\| = 32k$)	61,701	29,273	32,000	**3.758**	**2.081**
Llama 2	32,000	837	32,000	3.523	0.851

Table 2: Tokenization Efficiency for Chinese

Methods	Vocab Size	ZH Vocab Size	Common Vocab Size	EN LPT	ZH LPT
$V_{\text{com}} \cup V'_{\text{uni}}$ (VRCP)	32,000	13,476	12,756	**3.767**	1.257
$V \cup V'$ ($\|V'\| = 16k$)	46,073	14,065	32,000	**3.767**	1.445
$V \cup V'$ ($\|V'\| = 32k$)	61,191	28,284	32,000	3.736	**1.580**
Llama 2	32,000	700	32,000	3.523	0.705

slightly lower token efficiency compared to the vocabulary expansion methods, the Length Per Token (LPT) is almost equivalent or better, especially for English. Even with $|V'| = 32,000$, the improvement in LPT is not significantly greater compared to the efficiency improvement observed when replacing the unique vocabulary in the Llama-2 tokenizer with VRCP's vocabulary (refer to Table 1 and Table 2).

Overall, we emphasize that substantial improvement of VRCP on token efficiency for the target languages, achieved without expanding the vocabulary size, remains highly competitive against vocabulary expansion methods. This suggests that excluding non-target languages from the tokenizer enhances tokenization efficiency for both the target language and English, providing a balanced and efficient approach for multilingual tokenization (see Tables 1 and 2 for a summary of results).

4.2 Evaluation of Task Performance

We evaluated the performance on benchmark tasks using models pretrained with each tokenizer method:

- **English**: ARC, HellaSwag, MMLU, XLSum-EN
- **Japanese**: JCommonsenseQA, JSQuAD, NIILC, XLSum-JA
- **Chinese**: C-Eval, CMMLU, CMRC, XLSum-ZH

4.2.1 Discussion on Task Performance for Japanese and Chinese

Japanese: Overall, the Unchanged method showed the highest average performance across tasks (see Table 3). This may be explained by the fact that the Llama-2 model has been extensively trained with its tokenizer on a vast amount of data, optimizing the model for this tokenizer. This phenomenon has also been reported in previous studies. However, when excluding the Unchanged method, VRCP demonstrated the best performance among the methods tested.

In particular, in summarization tasks, VRCP significantly improved performance compared to the Unchanged method (see Table 3). This improvement was especially notable in the version of VRCP without embedding replacement, which performed better than the vocabulary expansion methods. In fact, the more we expanded the total vocabulary size, the lower the performance tended to be. This may be because modifying the embedding vectors for vocabulary expansion can negatively impact text generation tasks. Even if the initial values of the embeddings do not align perfectly with the meaning of the vocabulary, it has been shown that maintaining these initial embeddings can be advantageous for text generation tasks.

Regarding English tasks, VRCP showed slightly lower performance compared to the vocabulary expansion methods. However, the decrease in performance was not severe enough to suggest a breakdown of the model. Notably, the decrease in performance for English summarization tasks was less pronounced than for other tasks. This indicates that the extensive training of Llama-2 has made the model robust to some changes in the tokenizer, especially for text generation tasks, e.g., summarization (see Table 3 for performance metrics).

Chinese: Similar trends were observed for Chinese tasks. VRCP performed best in summarization tasks, with particularly strong results in the version without embedding replacement. The pattern of performance decreasing as the vocabulary size expanded was also noted in Chinese, indicating

Table 3: Performance for Different Methods (Japanese)

Type	Method	EN		JA		GPU Memory Footprint (GB)
		Avg.	XLSum-EN	Avg.	XLSum-JA	
Vocab Replace **(VRCP)**	$V_{\text{com}} \cup V'_{\text{uni}}$	.627	.900	.672	.734	**52.78**
	$V_{\text{com}} \cup V'_{\text{uni}}$ (Without Embed Replace)	.617	.897	.637	**.737**	**52.78**
Vocab Expand **(Previous)**	$V \cup V'$ ($\|V'\| = 16k$)	**.643**	**.901**	.668	.736	54.57
	$V \cup V'$ ($\|V'\| = 32k$)	.640	**.901**	.658	.717	56.16
Unchanged **(Baseline)**	V	.626	.900	**.691**	.712	**52.78**
Vanilla	Llama 2-7B	.670	.905	.591	.690	N/A

Table 4: Performance for Different Methods (Chinese)

Type	Method	EN		ZH		GPU Memory Footprint (GB)
		Avg.	XLSum-EN	Avg.	XLSum-ZH	
Vocab Replace **(VRCP)**	$V_{\text{com}} \cup V'_{\text{uni}}$	.622	**.902**	**.507**	.625	**52.78**
	$V_{\text{com}} \cup V'_{\text{uni}}$ (Without Embed Replace)	.616	**.902**	.502	**.652**	**52.78**
Vocab Expand **(Previous)**	$V \cup V'$ ($\|V'\| = 16k$)	.639	**.902**	.493	.605	54.51
	$V \cup V'$ ($\|V'\| = 32k$)	**.642**	.901	.488	.596	57.35
Unchanged **(Baseline)**	V	.633	.900	.483	.553	**52.78**
Vanilla	Llama 2-7B	.670	.905	.499	.619	N/A

that expanding the vocabulary may reduce performance, similar to what was seen with Japanese tasks. Additionally, the slightly lower performance in English tasks when using VRCP was observed in both Japanese and Chinese settings, but again, the decrease was not severe enough to compromise the model's effectiveness (see Table 4 for detailed results).

Summary of Task Performance: The experiments indicate that the vocabulary size in the target language does not necessarily impact performance. Both VRCP and the vocabulary expansion methods showed similar average scores across tasks. This suggests that expanding the vocabulary is not always essential to achieve high performance. Instead, the approach of VRCP, which involves replacing unique vocabulary without expanding the total vocabulary size, remains competitive and effective, especially in text generation tasks. Additionally, VRCP's improvement in English summarization tasks supports the benefit of vocabulary replacement for task performance in the target language as well as in English (see Tables 3 and 4 for task performance comparisons).

4.3 Evaluation of GPU Memory Footprint

We evaluated the GPU memory footprint of each method in experiments with Japanese and Chinese. VRCP maintains the same vocabulary size as the Llama-2 model, ensuring a consistent GPU memory footprint. This indicates its effectiveness in resource-constrained environments without the need for extensive vocabulary expansion (refer to Table 3 and Table 4 for memory footprint details).

Vocabulary expansion methods increased memory footprint. Specifically, as the vocabulary size increased, the GPU footprint increased linearly. This result indicates that extensive vocabulary expansions may not be efficient or necessary for improving performance (see Figure 3 for a visual representation of the linear increase in memory footprint).

As shown in Figure 3, even though increasing the vocabulary size does not significantly improve the Length Per Token (LPT), the GPU memory foot-

print continues to increase linearly. This demonstrates that increasing vocabulary size leads to a predictable linear increase in memory footprint, without a corresponding substantial improvement in tokenization efficiency.

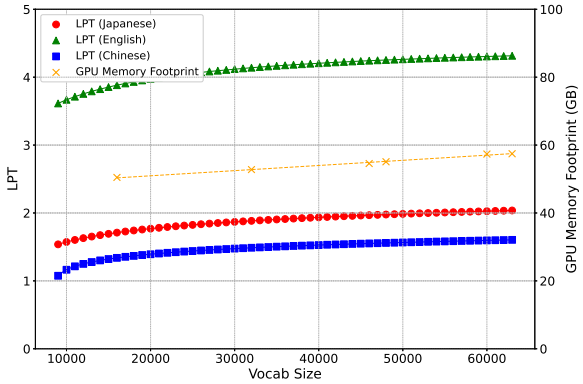


Figure 3: Relationship between vocabulary size, LPT, and GPU memory footprint.

5 Related Works

Several studies have explored enhancing model performance for low-resource languages by expanding vocabulary and embedding layers in continued pretraining. For example, Wang et al. (2020) and Pfeiffer et al. (2021) expanded the vocabulary and embedding layers for mBERT (Pires et al., 2019), which improved performance by incorporating low-resource language corpora. These methods typically involve expanding the vocabulary size, which increases the number of model parameters and the GPU memory footprint. This is because expanding the vocabulary requires adding corresponding embedding vectors. In decoder-only models like Llama-2 (Touvron et al., 2023), embeddings must be placed both after the input and before the output. Therefore, expanding the vocabulary by N_{new} tokens with an embedding dimension of D results in an increase of $2{\times}D{\times}N_{\text{new}}$model parameters. In contrast, our proposed method maintains the source vocabulary size by focusing on vocabulary replacement. This preserves the semantic knowledge of the source model while optimizing for the target language.

Limisiewicz et al. (2023) highlight the significant impact of tokenization on multilingual models, especially the importance of language-specific token coverage for word-level tasks. Their study provides guidelines for selecting tokenizers before expensive model pre-training.

Our research aligns with these insights by addressing the optimization of tokenizers for multilingual models without expanding the vocabulary size. This is particularly beneficial in resource-constrained environments. We provide an alternative strategy that leverages source model knowledge while adapting to the target language, complementing the guidelines suggested by Limisiewicz et al. (2023).

6 Conclusion

We introduced Vocabulary Replacement Continued Pretraining (VRCP), a method that optimizes tokenizers for non-English languages without increasing vocabulary size. VRCP replaces low-frequency English tokens with high-frequency target language tokens, leveraging the semantic knowledge of English models while enhancing token efficiency for the target language.

Our experiments with Japanese and Chinese corpora demonstrated that VRCP matches the performance of traditional vocabulary expansion methods and excels in tasks requiring text generation, such as summarization. This improved performance in generation tasks suggests that VRCP can be effectively applied to language models integrated into retrieval-augmented generation (RAG) frameworks, where generating coherent and contextually accurate summaries is critical. Additionally, VRCP avoids additional GPU memory costs by maintaining the original vocabulary size, making it suitable for resource-constrained environments.

Limitations

One limitation of our work is its language specificity. Our experiments were conducted only on Japanese and Chinese, meaning that the findings may not necessarily generalize to other languages. Since each language has unique characteristics, applying VRCP to other languages may require further adjustments and validation.

Additionally, while VRCP aims to prevent an increase in GPU memory consumption by maintaining a constant vocabulary size, modern distributed training techniques already provide efficient memory management solutions. Frameworks like TensorParallel, PipelineParallel (Narayanan et al., 2021), and ZeRO (Rajbhandari et al., 2020)) offer alternative or complementary strategies for managing resource constraints in large-scale model training.

References

Bilge Acun, Matthew Murphy, Xiaodong Wang, Jade Nie, Carole-Jean Wu, and Kim Hazelwood. Understanding training efficiency of deep learning recommendation models at scale. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 802–814. IEEE.

François Chollet. 2019. On the measure of intelligence. *Preprint*, arXiv:1911.01547.

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.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. A span-extraction dataset for chinese machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5883–5889, Hong Kong, China. Association for Computational Linguistics.

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

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*, pages 4693–4703, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, jiayi lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pages 62991–63010. Curran Associates, Inc.

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. *Preprint*, arXiv:2001.08361.

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.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. Jglue: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966, Marseille, France. European Language Resources Association.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. Cmmlu: Measuring massive multitask language understanding in chinese. *Preprint*, arXiv:2306.09212.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. 2023. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5661–5681, Toronto, Canada. 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. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference*

for High Performance Computing, Networking, Storage and Analysis, SC '21, New York, NY, USA. Association for Computing Machinery.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, et al. 2023. Gpt-4 technical report.

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.

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

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Satoshi Sekine. 2003. Development of a question answering system focused on an encyclopedia (in japanese only). *9th Annual Meeting of the Association for Natural Language Processing*, pages 637–640.

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.

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*, volume 30.

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.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Bright Xu. 2019. Nlp chinese corpus: Large scale chinese corpus for nlp.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. 2021. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 460–470, Online. Association for 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, , et al. 2023. Bloom+1: Adding language support to bloom for zero-shot prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11682–11703, Toronto, Canada. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Choi Yejin. 2019. Hellaswag: Can a machine really finish your sentence? pages 4791–4800.

A Training Settings and Hyperparameters

In our experiments, we used the same settings and hyperparameters for continued pretraining across

all configurations, including VRCP, Vocabulary Expansion and Unchanged models (see Table 5).

Table 5: Training Hyperparameters

Hyperparameter	Value
Global Batch Size (GBS)	256
Sequence Length	4096
Learning Rate (LR)	7.5e-5
Warmup Ratio	0.05
Weight Decay	0.1
DeepSpeed ZeRO Stage (Rajbhandari et al., 2020)	2
AllGather Bucket Size	7e8
Reduce Bucket Size	7e8
GPU	NVIDIA H100 (80GB)
Number of GPUs	8

B Corpora Details

We used different corpora for Japanese and Chinese pretraining, as detailed in Table 6 and Table 7.

Table 6: Japanese Corpora Details

Corpus	Size (MB)
Wikipedia (Japanese)	3,178.95
CC100 (Japanese)	10,405.93
Wikipedia (English)	374.87

Table 7: Chinese Corpora Details

Corpus	Size (MB)
Wikipedia (Chinese)	1,115.09
Baike	1,186.11
News	5,837.07
Wikipedia (English)	374.87

C Evaluation Tasks

We evaluated the models on various benchmark tasks for Japanese, Chinese, and English. Each task focuses on different aspects of language understanding and generation, as summarized in Table 8, Table 9, and Table 10.

For the Japanese evaluation, we utilized the **JCommonsenseQA**, **JSQuAD**, **NIILC**, and **XLSum-JA** datasets.

Table 8: Japanese Evaluation Tasks

Task	Shots	Metric
JCommonsenseQA	4	Exact Match (EM)
JSQuAD	4	Character-level F1
NIILC	4	Character-level F1
XLSum-JA	1	BERTScore

Table 9: Chinese Evaluation Tasks

Task	Shots	Metric
C-Eval	4	Accuracy
CMMLU	4	Accuracy
CMRC	4	Accuracy
XLSum-ZH	1	BERTScore

Table 10: English Evaluation Tasks

Task	Shots	Metric
ARC	25	Normalized Accuracy
HellaSwag	10	Normalized Accuracy
MMLU	5	Accuracy
XLSum-EN	1	BERTScore

- **JCommonsenseQA** evaluates common sense reasoning abilities in Japanese. This dataset is part of the JGLUE benchmark and provides questions that require the model to use background knowledge to choose the correct answer from multiple choices (Kurihara et al., 2022).
- **JSQuAD** is a question-answering task also included in the JGLUE benchmark. It focuses on extracting answers from provided contexts based on Japanese text (Kurihara et al., 2022).
- **NIILC** (National Institute of Informatics Large-scale Encyclopedia Corpus) presents open-ended questions, where the model must generate answers using knowledge embedded within the model. This task assesses the model's encyclopedic knowledge and its ability to produce accurate responses (Sekine, 2003).
- **XLSum-JA** is a summarization task that requires the model to generate concise summaries from Japanese news articles (Hasan et al., 2021).

For evaluating Chinese language capabilities, we

used the **C-Eval**, **CMMLU**, **CMRC**, and **XLSum-ZH** datasets.

- **C-Eval** includes tasks for reading comprehension, text generation, and reasoning based on various domains of knowledge, such as literature, history, science, and technology (Huang et al., 2023).
- **CMMLU** (Massive Multitask Language Understanding in Chinese) encompasses a set of tasks across multiple domains, including comprehension, text generation, classification, translation, and dialogue (Li et al., 2024).
- **CMRC** (Chinese Machine Reading Comprehension) focuses on question-answering by extracting answers from given contexts (Cui et al., 2019).
- **XLSum-ZH** is the Chinese counterpart of the summarization task for news articles, where the model generates brief summaries from longer articles (Hasan et al., 2021).

The evaluation for English language tasks was conducted using the **ARC**, **HellaSwag**, **MMLU**, and **XLSum-EN** datasets, which test various aspects of knowledge and reasoning:

- **ARC** (AI2 Reasoning Challenge) is designed to assess middle school level science reasoning abilities. The dataset includes questions that require the model to apply scientific knowledge and reasoning skills to select the correct answer from multiple choices (Chollet, 2019).
- **HellaSwag** measures ability of the models to perform contextual and common sense reasoning (Zellers et al., 2019).
- **MMLU** (Massive Multitask Language Understanding) covers a wide range of knowledge domains and evaluates the model's capability to apply this knowledge in answering questions accurately (Hendrycks et al., 2021).
- **XLSum-EN** is the English version of the summarization task, where the model must create concise summaries from news articles (Hasan et al., 2021).

Author Index