ROMANSETU: Efficiently unlocking multilingual capabilities of Large Language Models models via Romanization

Jaavid Aktar Husain[1] **Raj Dabre**[2] **Aswanth Kumar**[3]
Ratish Puduppully[4] **Anoop Kunchukuttan**[5]

IIT Madras, India[1,5] IIIT D&M Kancheepuram[1] Flipkart, India[3]
National Institute of Information and Communications Technology, Kyoto, Japan[2]
Institute for Infocomm Research (I²R), A*STAR, Singapore[4] Microsoft, India[5] AI4Bharat[1,5]
[1]jaavidaktar@gmail.com [2]raj.dabre@nict.go.jp [5]ankunchu@microsoft.com

Abstract

This study addresses the challenge of extending Large Language Models (LLMs) to non-English languages, specifically those using non-Latin scripts. We propose an innovative approach that utilizes the romanized form of text as an interface for LLMs, hypothesizing that its frequent informal use and shared tokens with English enhance cross-lingual alignment. Focusing on Hindi, we demonstrate through Hindi-to-English translation and sentiment analysis tasks that romanized text not only significantly improves inference efficiency due to its lower fertility compared to native text but also achieves competitive performance with limited pre-training. Additionally, our novel *multi-script* prompting approach, which combines romanized and native texts, shows promise in further enhancing task performance. These findings suggest the potential of romanization in bridging the language gap for LLM applications, with future work aimed at expanding this approach to more languages and tasks.

1 Introduction

Large language models (LLMs) demonstrate remarkable proficiency across a broad spectrum of natural language processing (NLP) tasks, as evidenced by various studies (Liu et al., 2023; Chung et al., 2022; Chowdhery et al., 2022; Wei et al., 2022; Goyal et al., 2022). They excel not only in tasks for which they were explicitly trained but also in those they were not trained for. This achievement is mainly due to the availability of corpora (Wenzek et al., 2019; Abadji et al., 2021; Suárez et al., 2019) as well as the advancements in LLMs that leverage these datasets for pretraining (Touvron et al., 2023; Workshop et al., 2022; Chowdhery et al., 2022). Despite their proficiency in English, these models typically demonstrate reduced effectiveness when applied to non-English languages, highlighting a significant challenge in extending their benefits to non-English languages.

The English-heavy LLMs (Touvron et al., 2023; Jiang et al., 2023; Zhang et al., 2022) still have some representation from other languages due to data leakage while creating the pre-training dataset, particularly for languages which use the same script as English i.e. Latin script. This script sharing enables cross-lingual transfer and bestows some of these LLM capabilities to these languages. For languages using non-Latin scripts, the data representation is very limited to non-existent. Tokenization of text in the languages exhibits high fertility (Ács, 2020) and byte-level representation (Artetxe et al., 2019). Hence, these LLMs perform poorly on most of these languages, and the inefficient tokenization also leads to high inference latency. This disparity in performance raises a critical question: *how can we extend the capabilities of LLMs to the languages written in non-Latin scripts?*

A widely explored solution is extension of the tokenizer vocabulary to incorporate new languages and continual pre-training on native language data (Cui et al., 2023; Nguyen et al., 2023; Minixhofer et al., 2021). This approach is computationally demanding, since the models need to be continually pre-trained long enough to effectively learn the new embeddings and align the representations of English and the new language. Furthermore, this approach requires the availability of large volumes of text corpora.

In this work, we explore an alternative approach for more efficient knowledge transfer. Instead of utilizing the native script, we use the romanized form of text as the interface to the LLMs. The adoption of romanized representation is justified for several reasons. In many languages, romanized text has been frequently used in informal settings and on social media in recent history. This usage creates the potential for the inclusion of some romanized data representation in the pre-training

[1]Work done during internship at AI4Bharat, IIT Madras.

corpus. Additionally, code-mixing with English is a common occurrence, and the romanized form shares tokens with English. This leads us to hypothesize that the romanized form is better aligned with English than the native script, thereby facilitating a more effective transfer from English.

We present our initial results using romanization to extend the capabilities of LLMs to languages not represented in the Latin script. Taking the case of Hindi, an Indo-Aryan language written in the Devanagari script and spoken by over 500 million people, we conduct experiments in two tasks: Hindi-to-English translation and sentiment analysis. Our experiments and analysis indicate that:

- The fertility of the romanized text is 2x times lower than the native text, making the romanized form far more efficient than the native script.
- Continually pre-training on romanized data is key to unlocking the performance on non-Latin script languages. A model continually pre-training with limited romanized data is competitive with the base model using native text. Hence, inference efficiency improves significantly without any significant negative effect on task performance.
- Romanized representation can complement the native representation. We propose a *multi-script* prompting approach that jointly prompts with romanized and native text to improve task performance.

We plan to extend our experiments to more tasks and languages to provide stronger validation of our hypothesis.

2 Related Work

Transliteration refers to the conversion of text written in one script to another. Romanization is a specific instance of transliteration, where the target script is the Roman/Latin script. Romanization has special significance since Roman script is by far the most widely adopted script in the world and many language models are primarily trained for English, which is written in the Roman script. Transliteration is typically used to represented different languages in the same script to enable cross-lingua transfer.

Transliteration has been used to improve cross-lingual transfer in NMT. Amrhein and Sennrich (2020) show that transliteration shows improvements for low-resource languages with different scripts by transferring from related high resource languages that use different scripts. Goyal et al. (2020) show that transliteration helps even when only contact relationship exists between the languages involved. Song et al. (2020) show that transliteration during pre-training stage for NMT also helps cross-lingual transfer.

Transliteration has also been used for cross-lingual transfer in the context of pre-trained language models. Some works (Khemchandani et al., 2021; Dhamecha et al., 2021; Moosa et al., 2022; Purkayastha et al., 2023) employ transliteration to a common script during the pretraining phase to enable cross-lingual transfer. Other studies (Dabre et al., 2022; Muller et al., 2021; Chau and Smith, 2021) adopt transliteration during the fine-tuning. Transliteration could be done to a common non-Latin script (Khemchandani et al., 2021; Dhamecha et al., 2021; Dabre et al., 2022; Doddapaneni et al., 2023) or to the Latin script (Muller et al., 2021; Moosa et al., 2022; Purkayastha et al., 2023).

While transliteration has been explored for language modeling as described above, our work differs from previous work in some impoertant aspects:

- Previous work explored cross-lingual transfer using transliteration with multilingual language models. We focus our attention on English LLMs, and try to achieve cross-lingual transfer via romanization using English LLMs. This is a challenging scenario, since the language can be unrelated to English, and very little (if any) native or romanized data in the language under study might be seen during pre-training of the English LLM. At the same time, this is a very practical need since most best performing LLMs are English-heavy and hence cross-lingual transfer via romanization is an important direction to explore.
- We investigate the utility of transliteration in decoder-only language models, which is the standard architecture for LLMs. In contrast, all previous work explored cross-lingual transfer with transliteration in the context of encoder-only models (the exception is Dabre et al. (2022) which use encoder-decoder models).
- Additionally, we delve into multi-script

prompting, using both native and romanized inputs, along with continual pretraining and supervised fine-tuning, in a language model pretrained on mostly-English data.

To the best of our knowledge, no prior research has investigated the leveraging of romanization for mostly-English LLMs in the context of cross-lingual transfer to non-English languages.

3 Utilizing Romanized Data to make LLMs Multilingual

Our proposed approach aims to enhance the capabilities of mostly-English LLMs for non-English languages by leveraging romanized data. We first continually pretrain the LLM with both romanized and English data to create a base LM that is romanization-aware. Subsequently, we extend the process by fine-tuning the continually pretrained model for the specific task. The framework of our approach is illustrated in Figure 1, encompassing the stages of data romanization, pre-training, and fine-tuning.

3.1 Romanization Scheme

A variety of romanization schemes are available, each driven by multiple considerations. One key factor is the resemblance of the romanized representation to the way people typically write romanized text. This is particularly advantageous if the pre-training data includes romanized text, as it might aid in aligning with English. Another important aspect is the fertility achieved by the romanization scheme when considering the original LLM's tokenizer. A third consideration is whether the transliteration scheme is lossy or lossless. A lossless scheme is preferable when the objective is to convert the output back to the native script. Typically, deterministic transliteration schemes are lossless, whereas natural transliteration schemes tend to be lossy.

In this work, we evaluated two romanization schemes for Hindi: (a) the extended ITRANS scheme (Kunchukuttan, 2020), which defines a fixed, reversible mapping between Devanagari and Roman characters, and (b) the IndicXlit scheme (Madhani et al., 2023), which generates romanizations as commonly used by Hindi speakers in informal contexts and is learned from parallel transliteration corpora. These mappings are inherently lossy. The IndicXlit romanization demonstrates lower fertility compared to ITRANS romanization (2.98 *vs.* 3.91). Preliminary prompting experiments with the LLaMa2 model also indicated that IndicXlit outperforms ITRANS in romanized Hindi to English translation. Consequently, we selected IndicXlit as our romanization scheme for this project. However, it is important to note that the transliterations are not reversible. With continued pre-training, ITRANS transliterations might eventually achieve similar task performance to IndicXlit while ensuring script reversibility. We leave this exploration for future work.

3.2 Continual Pretraining

To render the base model aware of romanization, we continue to pretrain the base LLM with romanized document-level data. To prevent any catastrophic forgetting of English capabilities, we also incorporate an equal amount of English data into the pre-training mix. Additionally, we explore the inclusion of native script data in the pre-training mixture as part of our multi-script prompting approach. It is important to note that native script pre-training is not necessary for romanized prompting. We maintain the use of the original tokenizer embedding and do not expand the vocabulary for either the romanized or native script text.

3.3 Supervised Fine-tuning

The LLM is subsequently fine-tuned using task-specific data pairs (input, output) to align the model for performing the specified task. Following standard supervised fine-tuning practices, the cross-entropy loss is calculated solely on the output tokens. In this work, we fine-tune the model with romanized SFT data. Moreover, we also undertake fine-tuning with native script SFT data and multi-script SFT data. In the current work, we fine-tune the model separately for different tasks. The exploration of multi-task fine-tuning is reserved for future work.

3.4 Prompting

In this study, we investigate both zero-shot and few-shot prompting techniques. For the romanization-aware models, we employ prompts in romanized text. In the case of other model variations, native script prompting is also explored for comparative purposes. Additionally, we introduce the concept of *multi-script* prompting. In this method, the model receives prompts in both the native script and its romanized form. The rationale behind this approach is that the native script may have been

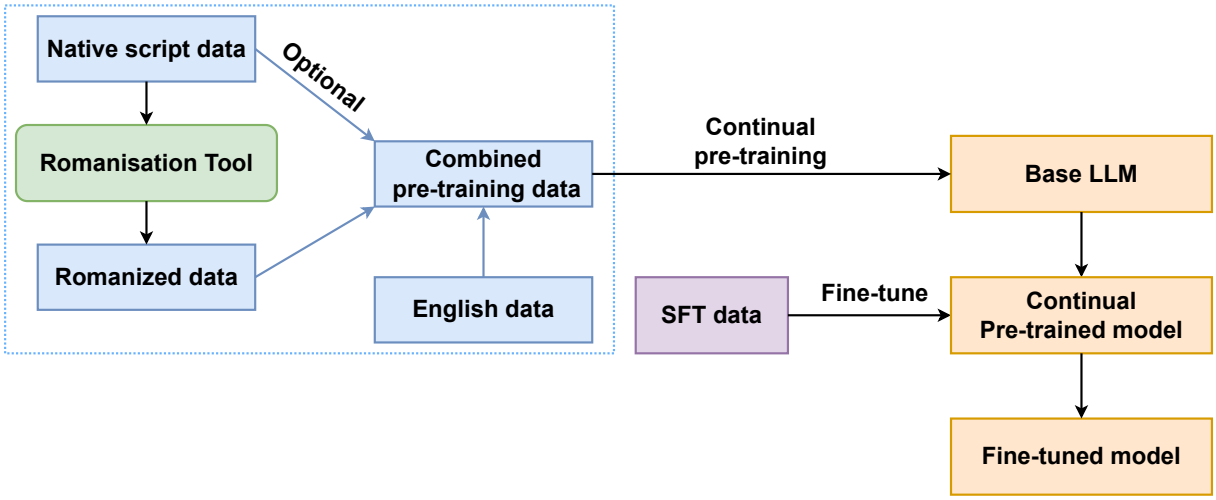

Figure 1: Overview of our proposed approach consisting of romanization, continual-pretraining and fine-tuning.

more prevalent in the original pre-training data, while the romanized script potentially offers better alignment with English. Our aim is that these two representations will mutually enhance the model's performance. Figure 2 depicts the various prompting methods under exploration.

4 Experimental Settings

We conducted experiments using LLaMA2 7B (Touvron et al., 2023) as the base model under various settings. To assess the effectiveness of romanization, we conducted experiments for two tasks: Hindi-to-English machine translation and Hindi sentiment analysis.

4.1 Datasets

Continual Pretraining: For continual pretraining, we sourced approximately 100 million words of document-level data from web-crawled Hindi corpora (Doddapaneni et al., 2023). To generate the romanized dataset, we transliterated this Hindi dataset using the *IndicXlit* model (Madhani et al., 2023). Both the original Hindi dataset and its romanized counterpart were then used for continual pretraining.

Supervised Fine-tuning: In the supervised fine-tuning phase for machine translation, Hindi-English pairs were obtained from the *BPCC-H-Wiki* and *BPCC-H-Daily* seed data within the BPCC corpus (Gala et al., 2023), comprising roughly 40,000 parallel sentences. For sentiment analysis, we employed the SST2 dataset (Socher et al., 2013), which includes about 67,300 instances of sentiment analysis data in the training split. It is important to note that for sentiment analysis, we fine-tuned the model using English data but evaluated its performance on Hindi.

Evaluation Data: The FLORES-200 test set (Costa-jussà et al., 2022) was used for evaluating machine translation, and the IndicSentiment (Doddapaneni et al., 2023) for sentiment analysis. For machine translation, we employed the *dev* set for model checkpoint selection and the *devtest* set for evaluation. The first three examples from the *dev* set were used as few-shot exemplars. A similar approach was applied to the *dev* and *test* sets of IndicSentiment.

4.2 Training and Finetuning Details

We adapt the *open-instruct*[1] (Wang et al., 2023) for our continual pre-training and fine-tuning experiments. The models are fully-finetuned during continual pre-training as well supervised finetuning. Detailed information about the hyperparameters used for both continual pretraining and supervised fine-tuning can be found in Appendix A. The optimal hyperparameters were identified based on their performance in the validation set.

4.3 Decoding Details

For machine translation generation, we use greedy decoding with the bfloat16 precision. Throughout all experiments, we maintained a consistent batch size of 8 to ensure uniformity and reliability across various experimental setups.

[1] https://github.com/allenai/open-instruct/tree/main

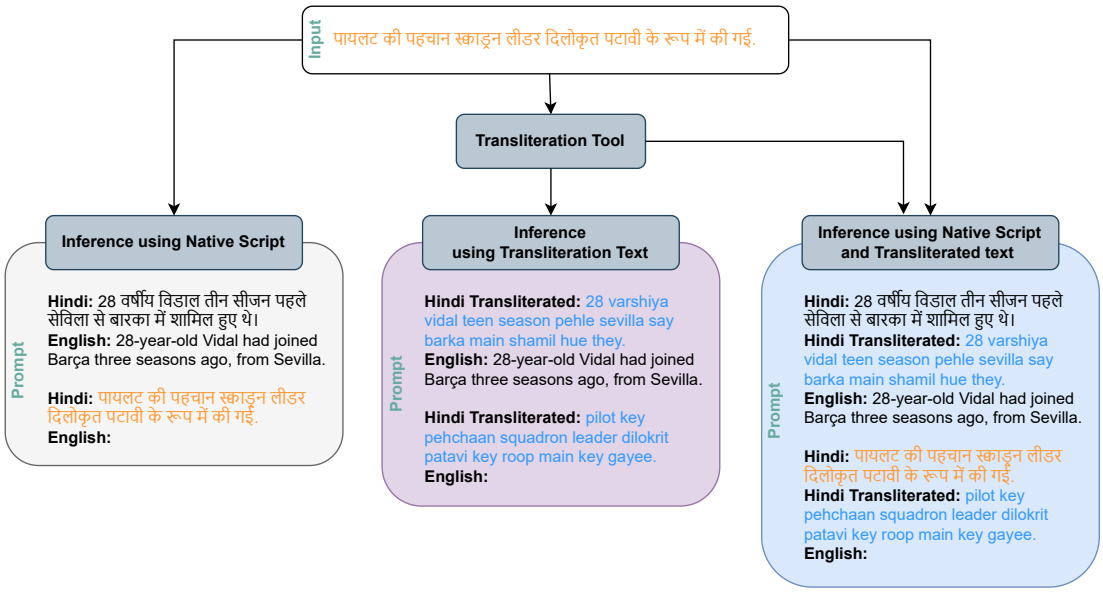

Figure 2: Overview of our romanization-based multi-script prompting. The setup on the left shows the usual inferencing using the native script, the middle setup utilizes the transliteration tool and constructs prompts using transliterated text. On the right is a multi-script setup that constructs prompts using both native and transliterated scripts. This is a *1-shot* with a newline as a delimiter between in-context examples and input.

4.4 Evaluation Metrics

The evaluation metric for machine translation in our experiments is BLEU (Papineni et al., 2002) computed using the SacreBLEU toolkit (Post, 2018).[2] For the sentiment analysis task, we report accuracy scores.

4.5 Models compared

We use the 7B parameter LLAaMA2 base model as our starting point for all our experiments. We evaluate the base model and 2 continually pre-trained models: (a) using romanized + English data, (b) romanized + native script + English data. We also explored 4 variants of SFT models, depending on the script configuration of the Hindi SFT data: (a) romanized examples, (b) native script examples, (c) multi-script examples, (d) aggregating examples of all the three types (a-c).

4.6 Prompting Methods compared

We experimented with *0-shot*, *1-shot* and *3-shot* prompting. We also experimented with native script, romanized and multi-script prompting. Below is a prompt template for the k-shot setting, incorporating both native and transliterated text for machine translation task:

```
Translate the following sentences from
Hindi to English. The output should be
in English and no other language.
   Hindi: [x_1]
   Hindi Transliterated: [t_1]
   English: [y_1]

   ...

   Hindi: [x_k]
   Hindi Transliterated: [t_k]
   English: [y_k]

   Hindi: [x]
   Hindi Transliterated: [t]
   English:
```

The prompt templates for various prompting methods and the sentiment analysis task are provided in the Appendix B.

5 Results

Table 1 shows the results for the machine translation experiments from Hindi to English, while Table 3 presents the results for the sentiment analysis task in Hindi.

Few-Shot Prompting of Base Model In the few-shot prompting of the base LLaMA2 model, prompting with native scripts outperforms prompting with romanized data (see Rows 1 and 2). A

[2] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1

#	CPT Data	SFT Data	#shots	Prompting Data		
				H	RH	H+RH
1	✗	✗	1	16.7	6.4	7.2
2			3	16.2	6.5	17.0
3		H→E	0	25.2	-	-
4		RH→E	0	-	26.0	-
5		H+RH→E	0	-	-	**28.4**
6		All	0	22.6	25.4	27.4
7	RH+ E	✗	1	-	15.7	-
8			3	-	15.4	-
9		H→E	0	25.3	-	-
10		RH→E	0	-	28.6	-
11		H+RH→E	0	-	-	**28.8**
12		All	0	23.7	26.6	28.0
13	H+ RH+ E	✗	1	24.1	16.7	23.0
14			3	22.3	17.0	23.5
15		H→E	0	26.5	-	-
16		RH→E	0	-	28.1	-
17		H+RH→E	0	-	-	**29.8**
18		All	0	24.9	26.5	27.3

Table 1: BLEU scores for the machine translation of Hindi (H) to English (E) using different types of continual pre-training (CPT) data, supervised fine-tuning (SFT) data, number of shots (#shots), and the type of script used, namely, native Hindi (N), Romanized Hindi (RH) and multi-script (H+RH) methods. The highest scores, for each type of CPT data used, are in **bold** text. We prompt non-SFT models with 1 and 3 shots, whereas we prompt SFT models in 0 shot manner.

combination of native script and romanized data in a multi-script approach enhances the effectiveness over using native data alone for 3-shot prompting. The comparatively lower performance when using solely romanized Hindi is understandable, given that the base LLaMA2 model is less familiar with the vocabulary in the romanized script compared to that in the native script.

Continual Pre-training on Romanized Data Enhances Task Performance We see that continual pre-training on limited amount of romanized data significantly enhances translation quality in few-shot prompting scenarios. The Continual Pre-Training (CPT) model using romanized data prompting achieves performance comparable to that of the base model prompted with native script text, as demonstrated by the comparison between Rows 7 and 8 with Rows 1 and 2.

Efficiency Gains with Romanized Prompting Table 2 presents the fertility scores for romanized and native script text in the LLaMA2 model. It is observed that the sequence length for native script text is nearly twice that of the romanized text, indicating that processing romanized text is significantly more efficient than native text. Coupled with the previous results, this suggests that Continual Pre-Training (CPT) with romanized data not only achieves translation quality comparable to native script but also offers substantial efficiency gains at inference time.

Language	Fertility Score
Hindi	7.36
Romanized Hindi (IndicXlit)	2.98

Table 2: Fertility scores of LLaMA2 model computed on FLORES-200 dev set dataset.

Supervised Fine-tuning on Romanized Data Supervised fine-tuning of the base model using romanized text has been found to significantly enhance translation quality, achieving equivalence to native text translation (see Rows 3 and 4). Furthermore, supervised fine-tuning (SFT) on the Continual Pre-Training (CPT) model with romanized data results in further improvements in translation performance (refer to Rows 9 and 10). In fact, romanized translation outperforms native script translation, suggesting that romanized model can better benefit from the English LLM due to better cross-lingual overlap. Consequently, romanized SFT presents itself as an efficient and effective alternative to native script SFT.

Multiscript Enhances Performance of Mostly-English LLMs A substantial improvement in translation quality has been observed through the use of multiscript prompting. This finding indicates that including transliterated text along with

#	CPT Data	SFT Data	#shots	Prompting Data		
				H	RH	H+RH
1	✗	✗	1	0.51	0.51	0.51
2			3	0.75	0.52	0.69
3		E→S	0	**0.91**	0.89	0.88
4	RH+E	✗	1	0.51	0.51	0.52
5			3	0.62	0.72	0.77
6		E→S	0	0.89	0.89	**0.93**
7	H+RH+E	✗	1	0.51	0.51	0.51
8			3	0.75	0.56	0.84
9		E→S	0	0.94	**0.95**	0.91

Table 3: Accuracies for sentiment analysis of Hindi (H) data to get sentiments (S) using different types of continual pre-training (CPT) data, number of shots (#shots), and the type of script used, namely, native Hindi (N), Romanized Hindi (RH) and multi-script (H+RH) methods. Supervised fine-tuning (SFT) is done on English(E) sentiment data. The highest scores, for each type of CPT data used, are in **bold** text. We prompt non-SFT models with 1 and 3 shots, whereas we prompt SFT models in 0 shot manner.

native script significantly boosts the model's ability to produce superior machine translations. Notably, this method surpasses native script prompting by as much as 3.3 BLEU points (see Rows 5, 11 and 17 versus preceding rows in their corresponding block), emphasizing the beneficial role of transliterated text in enhancing machine translation performance.

SFT on multiple data formats If we finetune the model on multiple prompting formats, we see an decline in performance compared to a model finetuned on just a single format (comparison between Rows 5, 11 and 17 with Rows 6, 12, and 18.)

Sentiment Analysis Performance In sentiment analysis, we observed that supervised fine-tuning (SFT) with an English dataset enhances performance across various settings. This includes the base LLaMA2 model, the model continually pretrained with both romanized and English scripts, and the model pretrained with a combination of romanized, native, and English scripts (Rows 3, 6 and 9). The romanized and native script performances are comparable, suggesting that cross-lingual transfer work well for romanized data inputs as well. This is in line with results for machine translation presented earlier.

6 Conclusion and Future Work

In this study, we proposed the use of romanization to enhance the performance of LLMs primarily trained in English. Our approach successfully unlocks LLM capabilities for non-English languages while maintaining task performance. We have empirically demonstrated the effectiveness of this strategy through experiments involving few-shot prompting, continual pretraining, and supervised fine-tuning in tasks such as Hindi-to-English machine translation and Hindi sentiment analysis. Additionally, our results indicate that leveraging romanized data significantly improves inference efficiency.

Looking forward, we aim to expand our experiments to encompass more languages and explore a broader range of NLP tasks, particularly generation tasks. Our primary focus will be on training models using larger text corpora that span multiple languages. This will involve a special emphasis on cross-lingual transfer and cross-task transfer, broadening the scope and impact of our research.

Limitations

Our proposed approach involves utilizing romanization to unlock the capabilities of mostly-English LLMs for non-English languages. While we have demonstrated the effectiveness of romanization using the LLaMA2 model in conjunction with continual pretraining and fine-tuning, the generalizability of our findings to other multilingual language models remains uncertain. Additionally, the pretraining data utilized in our experiments constitutes only a small portion, and the model might benefit further from exposure to a larger dataset. Resource constraints led us to limit our experiments to a 7B LLaMA model, but obtaining a more extensive understanding could be possible with a larger model. Evaluation on a wider set of tasks will help to better understand the generalization of this approach.

Ethics Statement

We use romanization as a way to align English with other languages using non-Latin scripts. Given the current state of LLMs, this seems like a practical direction to extend capabilities of the best LLMs to other languages. The intention is not to supplant the use of a native script (which is widely adopted and has a rich literary tradition) with romanized script. Further advancments are needed to extend this line of research to improve native script performance efficiently.

This work does not involve any new data collection and does not employ any annotators for data collection. We utilize publicly available datasets for the experiments reported in this work. Some of these datasets originate from web crawls, and we do not explicitly attempt to identify any biases within these datasets, using them in their original form.

References

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2021. Ungoliant: An optimized pipeline for the generation of a very large-scale multilingual web corpus. In *CMLC 2021-9th Workshop on Challenges in the Management of Large Corpora*.

Chantal Amrhein and Rico Sennrich. 2020. On Romanization for model transfer between scripts in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2461–2469, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*.

Ethan C. Chau and Noah A. Smith. 2021. Specializing multilingual language models: An empirical study. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 51–61, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh Khapra, and Pratyush Kumar. 2022. IndicBART: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863, Dublin, Ireland. Association for Computational Linguistics.

Tejas Dhamecha, Rudra Murthy, Samarth Bharadwaj, Karthik Sankaranarayanan, and Pushpak Bhattacharyya. 2021. Role of Language Relatedness in Multilingual Fine-tuning of Language Models: A Case Study in Indo-Aryan Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8584–8595, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain, and Amit Bhagwat. 2020. Contact relatedness can help improve multilingual NMT: Microsoft STCI-MT @ WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 202–206, Online. Association for Computational Linguistics.

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*.

Yash Khemchandani, Sarvesh Mehtani, Vaidehi Patil, Abhijeet Awasthi, Partha Talukdar, and Sunita Sarawagi. 2021. Exploiting language relatedness for low web-resource language model adaptation: An Indic languages study. 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 1312–1323, Online. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Khapra. 2023. Aksharantar: Open Indic-language transliteration datasets and models for the next billion users. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 40–57, Singapore. Association for Computational Linguistics.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2021. Wechsel: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. *arXiv preprint arXiv:2112.06598*.

Ibraheem Muhammad Moosa, Mahmud Elahi Akhter, and Ashfia Binte Habib. 2022. Does transliteration help multilingual language modeling? *arXiv preprint arXiv:2201.12501*.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, et al. 2023. Seallms–large language models for southeast asia. *arXiv preprint arXiv:2312.00738*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Sukannya Purkayastha, Sebastian Ruder, Jonas Pfeiffer, Iryna Gurevych, and Ivan Vulić. 2023. Romanization-based large-scale adaptation of multilingual language models. *arXiv preprint arXiv:2304.08865*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Haiyue Song, Raj Dabre, Zhuoyuan Mao, Fei Cheng, Sadao Kurohashi, and Eiichiro Sumita. 2020. Pre-training via leveraging assisting languages for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 279–285, Online. Association for Computational Linguistics.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. How far can camels go? exploring the state of instruction tuning on open resources.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

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*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Judit Ács. 2020. Exploring BERT's vocabulary. http://juditacs.github.io/2019/02/19/bert-tokenization-stats.html.

A Hyperparameter Details

Hyperparameter	Value
Batch Size (tokens)	1M, 2M
Learning Rate	0.00005
Number of Epochs	1, 2, 3, 4
Maximum Sequence Length	2,048

Table 4: The range of hyperparameters used for continual pretraining.

Hyperparameter	Value
Batch Size (examples)	128
Learning Rate	0.00005
Number of Epochs	1, 2, 3, 4
Maximum Sequence Length	2,048[3]

Table 5: The range of hyperparameters used for supervised fine-tuning.

[3]Although this is the maximum permissible length, most examples fall far below this length during fine-tuning.

B Prompts

Translate the following sentences from Hindi to English. The output should be in English and no other language.
Hindi: [सोमवार को, स्टैनफ़ोर्ड यूनिवर्सिटी स्कूल ऑफ़ मेडिसिन के वैज्ञानिकों ने एक नए डायग्नोस्टिक उपकरण के आविष्कार की घोषणा की जो कोशिकाओं को उनके प्रकार के आधार पर छाँट सकता है: एक छोटी प्रिंट करने योग्य चिप जिसे स्टैण्डर्ड इंकजेट प्रिंटर का उपयोग करके लगभग एक अमेरिकी सेंट के लिए निर्मित किया जा सकता है.]
English: [On Monday, scientists from the Stanford University School of Medicine announced the invention of a new diagnostic tool that can sort cells by type: a tiny printable chip that can be manufactured using standard inkjet printers for possibly about one U.S. cent each.]

Hindi: [दस्तावेजों से पता चलता है कि चौदह बैंकों ने अमीर ग्राहकों को कर और अन्य नियमों से बचने के लिए अरबों अमेरिकी डॉलर की संपत्ति छिपाने में मदद की।]
English:

Example of *one-shot* prompting used in the machine translation task using native script.

Translate the following sentences from Hindi to English. The output should be in English and no other language.
Hindi Transliterated: [somwar quo, stanford university school off medicine key vaigyaanikon nay ack nae diagnostic upkaran key aavishkaar key ghoshana key zoo koshikaon quo unake prakaar key aadhaar para chhaant sakta hai: ack chhoti print karane yogya chip jise standard inkjet printer kaa upayog karke lagbhag ack american cent key lie nirmit kiya jaa sakta hai.]
English: [On Monday, scientists from the Stanford University School of Medicine announced the invention of a new diagnostic tool that can sort cells by type: a tiny printable chip that can be manufactured using standard inkjet printers for possibly about one U.S. cent each.]

Hindi Transliterated: [dastavejon say patta chalta hai kii chaudah banks nay ameer grahakon quo curr our any niyamon say bachane key lie arabon american dollar key sampatti chhipaane main madad key.]
English:

Example of *one-shot* prompting used in the machine translation task using transliterated script.

Translate the following sentences from Hindi to English. The output should be in English and no other language.
Hindi: [सोमवार को, स्टैनफ़ोर्ड यूनिवर्सिटी स्कूल ऑफ़ मेडिसिन के वैज्ञानिकों ने एक नए डायग्नोस्टिक उपकरण के आविष्कार की घोषणा की जो कोशिकाओं को उनके प्रकार के आधार पर छाँट सकता है: एक छोटी प्रिंट करने योग्य चिप जिसे स्टैण्डर्ड इंकजेट प्रिंटर का उपयोग करके लगभग एक अमेरिकी सेंट के लिए निर्मित किया जा सकता है.]
Hindi Transliterated: [somwar quo, stanford university school off medicine key vaigyaanikon nay ack nae diagnostic upkaran key aavishkaar key ghoshana key zoo koshikaon quo unake prakaar key aadhaar para chhaant sakta hai: ack chhoti print karane yogya chip jise standard inkjet printer kaa upayog karke lagbhag ack american cent key lie nirmit kiya jaa sakta hai.]
English: [On Monday, scientists from the Stanford University School of Medicine announced the invention of a new diagnostic tool that can sort cells by type: a tiny printable chip that can be manufactured using standard inkjet printers for possibly about one U.S. cent each.]

Hindi: [दस्तावेजों से पता चलता है कि चौदह बैंकों ने अमीर ग्राहकों को कर और अन्य नियमों से बचने के लिए अरबों अमेरिकी डॉलर की संपत्ति छिपाने में मदद की।]
Hindi Transliterated: [dastavejon say patta chalta hai kii chaudah banks nay ameer grahakon quo curr our any niyamon say bachane key lie arabon american dollar key sampatti chhipaane main madad key.]
English:

Example of *one-shot* prompting used in the machine translation task using multi script.

Classify the sentiment of the below English sentence into Positive or Negative. The output should be either 'Positive' or 'Negative'.
Hindi: [रेट्स प्रतिस्पर्धी हैं, लगभग मार्किट में सबसे बेस्ट हैं।]
Output: [Positive]

Hindi: [काफी भीड़ वाली जगह है ये!]
Output:

Example of *one-shot* prompting used in the sentiment analysis task using native script.

Classify the sentiment of the below English sentence into Positive or Negative. The output should be either 'Positive' or 'Negative'.
Hindi Transliterated: [rates pratispardhi hain, lagbhag market main sabse best hain.]
Output: [Positive]

Hindi Transliterated: [kaafee bheed vaali jagah hai yeye!]
Output:

Example of *one-shot* prompting used in the sentiment analysis task using transliterated script.

Classify the sentiment of the below English sentence into Positive or Negative. The output should be either 'Positive' or 'Negative'.
Hindi: [रेट्स प्रतिस्पर्धी हैं, लगभग मार्किट में सबसे बेस्ट हैं।]
Hindi Transliterated: [rates pratispardhi hain, lagbhag market main sabse best hain.]
Output: [Positive]

Hindi: [काफी भीड़ वाली जगह है ये!]
Hindi Transliterated: [kaafee bheed vaali jagah hai yeye!]
Output:

Example of *one-shot* prompting used in the sentiment analysis task using multi script.