**Nav**

**Shreyas**

**Paul**

**Anusha**

**Sebastian**

**Motivation**

* LLMs often struggle with non-English languages due to limited training data and lack of resources.
* Romanization is widely used for informal communication and has shared tokens with English, suggesting potential for cross-lingual alignment.
* Tokenization of text in the languages exhibits high fertility (avg number of words a tokenizer splits a word into) and byte-level representation. Hence, these LLMs perform poorly on most of these languages, and the inefficient tokenization also leads to high inference latency.
* Existing research involves
  + Cross lingual transfer using transliteration with multilingual language models opposed to better performing English only models (something that needs to be explored)
  + previous work explored cross-lingual transfer with transliteration in the context of encoder-only models this paper deals with decoder only models

Large Language Models are typically trained on massive amounts of text data, but the vast majority of this data is in English. This leads to challenges when using LLMs for tasks involving non-English languages. ROMANSETU proposes leveraging romanization, a technique used for converting text from a different writing system to the Roman (Latin) script. Romanization is frequently used for informal communication and shares many tokens with English, which could facilitate communication and learning between the LLM and non-English languages.

**Problem description**

* How can we efficiently extend LLMs to handle non-English languages, specifically those which use non-Latin scripts, without requiring massive amounts of non-English training data?
* Can we leverage existing resources and knowledge to improve LLM performance on diverse languages?

The core problem addressed by ROMANSETU is how to make LLMs more multilingual without requiring extensive non-English training data. This is a significant challenge because creating and collecting large datasets in multiple languages can be time-consuming and expensive. ROMANSETU seeks to find a way to leverage existing resources, such as romanized text, to improve LLM performance on non-English languages.

**Design**

**Romanization Schemes**

Evaluated 2 schemes -

(a) the extended ITRANS scheme - defines a fixed, reversible mapping between Devanagari and Roman characters, and

(b) the IndicXlit scheme - generates romanizations as commonly used by Hindi speakers in informal contexts, learned from parallel transliteration corpora.

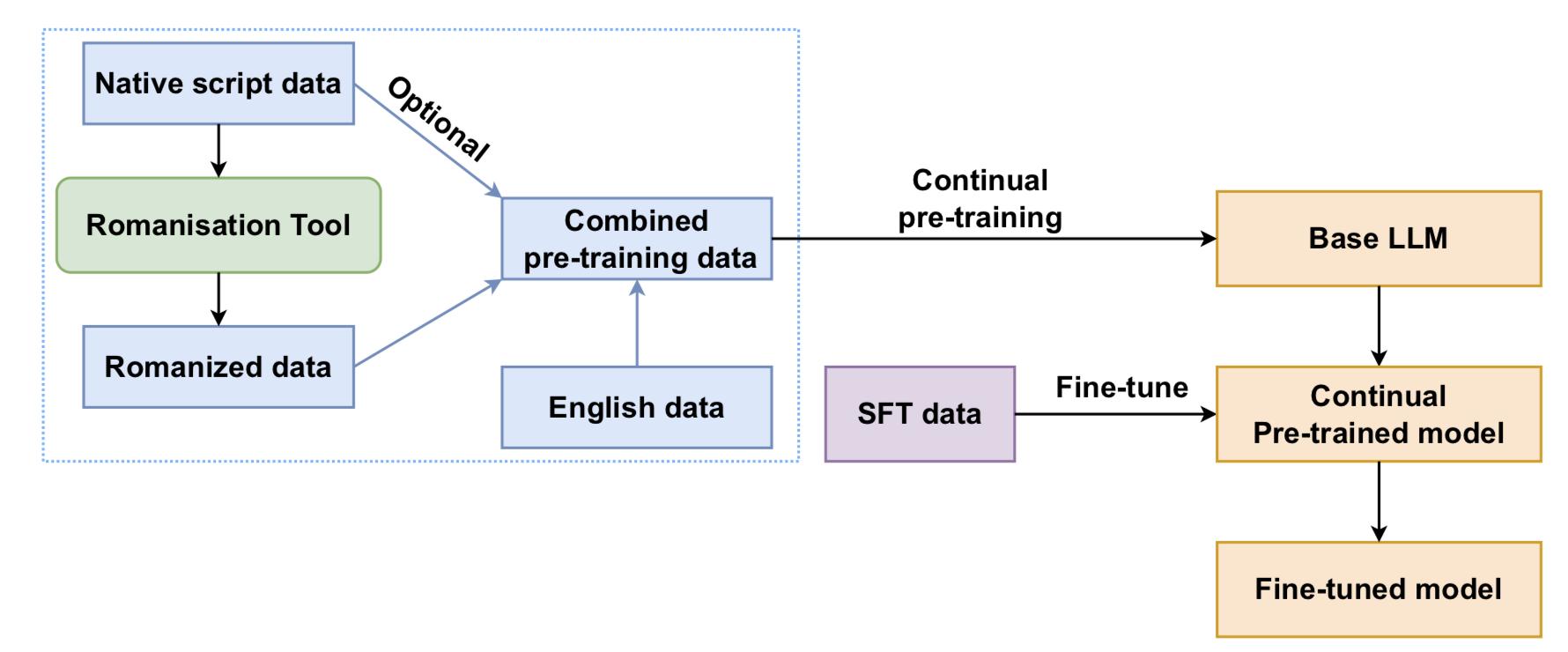
Experimentation revealed that the latter performs better in romanized Hindi to English translation. The authors hence selected IndicXlit. However, they believe that more experimentation can be done here to determine if ITRANS performs better with more pre-training - future work.

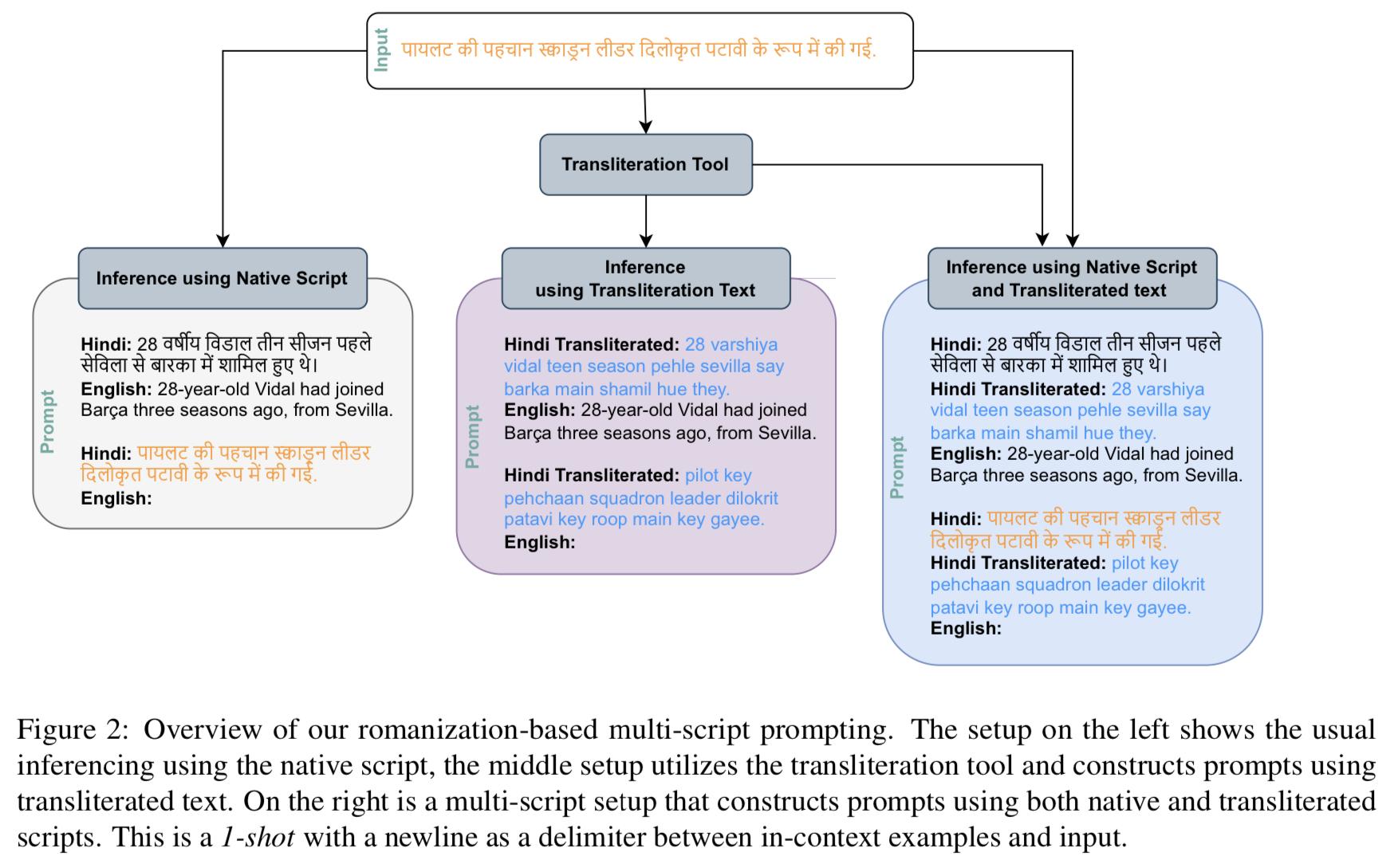
**Tasks performed**

* Hindi to English machine translation
* Hindi sentiment analysis

1. Datasets
   1. Continual pretraining - sourced approximately 100 million words of document-level data from web-crawled Hindi corpora. To generate the romanized dataset, we transliterated this Hindi dataset using the *IndicXlit* model
   2. Supervised finetuning - Hindi- English pairs were obtained from the *BPCC-H- Wiki* and *BPCC-H-Daily* seed data within the BPCC corpus, comprising roughly 40,000 parallel sentences,   
      employed the SST2 dataset (Socher et al., 2013), which includes about 67,300 instances of sentiment analysis data in the training split.
   3. Evaluation data - FLORES-200 test set was used for evaluating machine translation, and the IndicSentiment for sentiment analysis.
2. Training and finetuning
   1. adapt the *open-instruct* for our continual pre-training and fine-tuning experi- ments. The models are fully-finetuned during continual pre-training as well supervised finetuning.
3. Decoding - for machine translation greedy decoding with the bfloat16 precision

Important - Finetuned using english data but evaluated on hindi

* Use romanized text as an interface for the LLM.
* Explore different strategies:
  + Few-shot prompting: Provide a few examples to guide the LLM.
  + Zero shot prompting
  + Continual pre-training: Train the LLM on romanized data in addition to English data. To avoid the model forgetting its core English capabilities, the authors include an equal amount of English data in the pre-training mix.
  + Supervised fine-tuning: Fine-tune the LLM on specific non-English tasks using romanized data.
* Multi-script prompting: Combine romanized and native script text for enhanced performance.

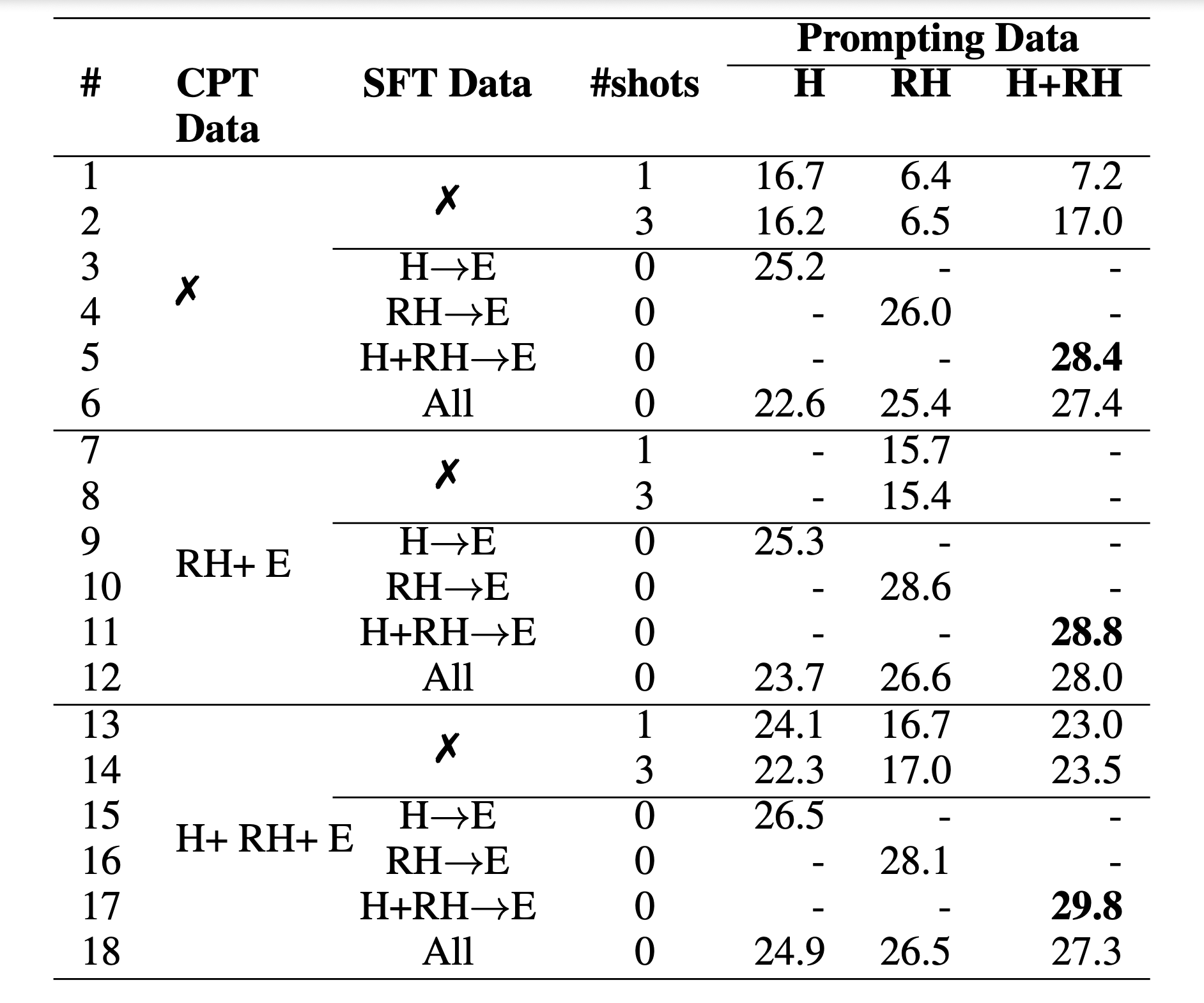


The ROMANSETU approach involves using romanized text as the primary input for the LLM. This leverages the shared vocabulary and structure between romanization and English, potentially improving LLM understanding of non-English languages. The researchers explore different strategies to utilize romanization, including few-shot prompting, continual pre-training, and supervised fine-tuning. They also propose a multi-script prompting approach that combines romanized and native script text for potentially even better results.

**Results**

**Hindi to english machine translation**

Evaluation metric: The chosen evaluation metric for the translation task were the BLEU scores of the predicted output.

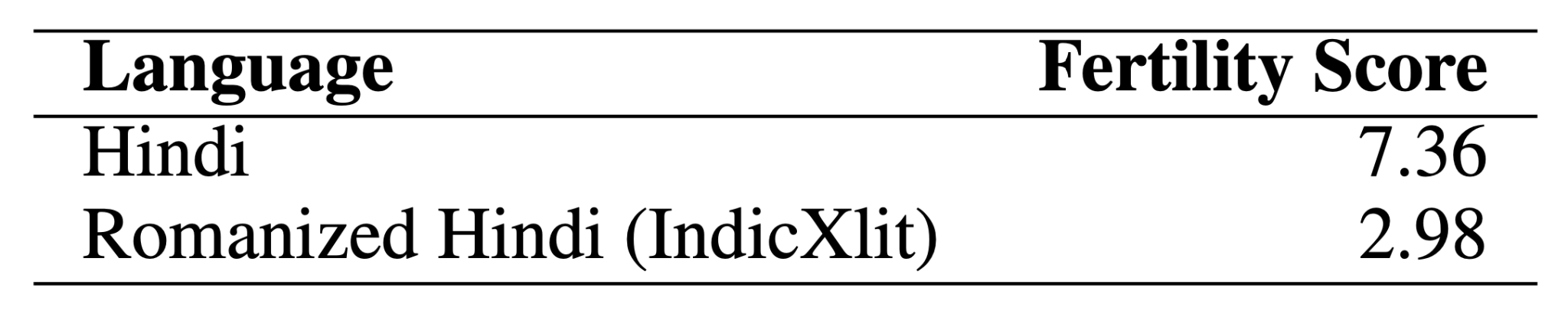
****

The table contains 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. The highest BLEU scores were consistently achieved when using the multi-script input of Hindi and Romanized Hindi. The highest Bleu score was achieved when using multi-script (H+RH+E) continual pre-training data when translating H+RH->E: 29.8

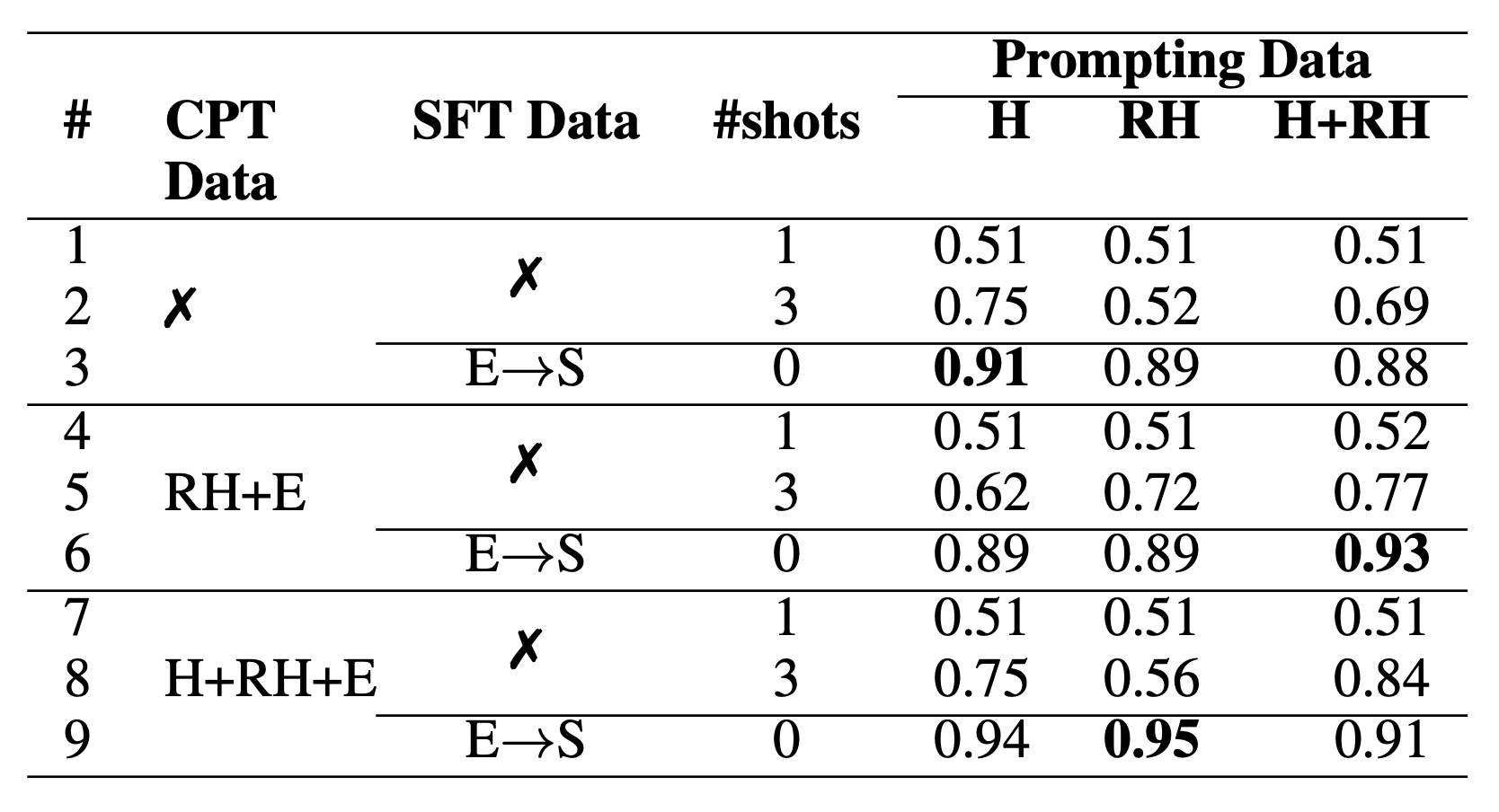
**Efficiency Gains with Romanized Prompting**

In addition to the improvement in BLEU scores, Romanized prompting resulted in an improvement in the efficiency of the LLM model. The reason for this improvement is the fertility achieved by the romanization scheme when considering the original LLM’s tokenizer.



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 which thereby significantly improving inference efficiency.

Hindi sentiment analysis

****

The table contains 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.

**Brief conclusion**

To conclude, the authors have proposed the use of romanization of non-English, non-Latin script languages to enhance the performance of the numerous state of the art LLMs out there, all of which have been trained on English. Their work showcases the effectiveness of their strategy, with experiments involving few-shot prompting, supervised fine-tuning on tasks like machine translation, and more. Their results indicate that leveraging romanized data can significantly improve the efficiency of inferences made using these languages.

Some extra notes: the cost per token isn’t a factor that is discussed here between approaches RH + E or just RH. Using more input tokens may increase the cost of having this system deployed for an application. Also, there are several models that are comparable to the performance of Llama2, like Mistral and the model used here was the 7B version (vs 12B, 70B are also available), bigger llms may have much superior performance and it would be interesting to see such comparison and analyze the tradeoffs of using a smaller model but with this technique VS just raw power of a bigger LLM like gpt4.

**Future Work**

Looking forward, one can aim to expand experiments to encompass more languages and explore a broader range of NLP tasks, particularly generation tasks. The 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 the research.