Sample Efficient Language Model to Generate Hinglish Corpus

Sakshi Singh sakshisi@usc.edu
Abhinav Prakash ap42546@usc.edu
Aakriti Shah shahaakr@usc.edu
Chaitanya Sachdeva csachdev@usc.edu
Sanjana Dumpala dumpala@usc.edu

1 Description

The aim of this project is to develop a low-resource, sample-efficient language model that generates a Hinglish corpus in response to English input. Hinglish, a blend of Hindi and English commonly used in informal online communication, remains underrepresented in natural language processing (NLP) datasets. We plan on utilizing multilingual pretrained language models, along with strategies like synthetic data generation, continuous pretraining, finetuning, and prompt-based learning, to minimize the need for large datasets while producing high-quality Hinglish outputs.

Our key goal is to develop a model capable of generating Hinglish responses using small parameter training models like ELCBert and MLSM. Our work will contribute to cross-lingual NLP research by leveraging low-resource models for efficient code-mixed language generation, expanding publicly available Hinglish datasets, and making these models more accessible in low-resource environments, fostering inclusivity in NLP.

2 Background Information

Large-scale multilingual language models (LLMs) have demonstrated remarkable performance across multiple languages; however, they remain computationally expensive and often perform sub-optimally on low-resource languages. To overcome these challenges, we propose using a lightweight, small parameter language model optimized for Hinglish generation.

Hinglish is increasingly used across the Indian subcontinent, especially on social media and in conversational text. Despite its growing prevalence, it remains largely underrepresented in NLP research, primarily due to the lack of annotated datasets. Most existing models focus on standard Hindi-English translation, neglecting the complexities of code-mixed language structures.

Our proposed approach offers a more efficient alternative to high-computational-cost LLMs, enabling real-time Hinglish generation with minimal data, while maintaining robust cross-lingual understanding. This research bridges the gap in code-mixed language processing and contributes to advancements in low-resource NLP.

3 Literature Review

Recent advancements in low-resource neural machine translation demonstrate promising approaches for languages with limited data availability. [5] Matzopoulos et al. (2025) showed that a small parameter LM architectures (ELC-BERT and MLSM) achieved significant performance gains on isiXhosa (which is low resource language) with only 13M words of training data. Their ELC-BERT implementation delivered a +3.2 F1 improvement on NER tasks while requiring 70% less training time than alternative models, establishing a small parameter LM as a viable foundation for resource-efficient NLP in morphologically complex languages.

In parallel, [6] Raviraj et al. developed Nemotron-Mini-Hindi 4B, a bilingual small language model focusing on Hindi-English language pairs. Through continued pre-training and synthetic data augmentation on 400B tokens, their model achieved state-of-the-art results on Hindi benchmarks while maintaining strong English performance, demonstrating that targeted pretraining enhances factual accuracy and language understanding in low-resource contexts.

[9] Shen et al. (2024) introduced BAMBINO-LM, a continual pretraining strategy inspired by bilingual child language acquisition. Their approach alternates between two languages during training and uses PPO-based perplexity rewards to boost performance in the heritage (low-resource) language. This method for code-mixed language processing shows us how models can develop capa-

bilities in multiple languages while balancing dominant and low-resource languages. Our project differentiates itself by applying small parameter LM, a significantly smaller and more computationally efficient model, to specifically Hinglish-to-English translation. However, we can apply the alternating method of this paper to help with our challenge of cross-lingual NLP for a mixed-language context.

4 Project Plan

4.1 Phase One: Data Collection & Augmentation

1. **Gather Existing Datasets:** We primarily plan on using publicly available datasets that include Hinglish text, like Hinglish-TOP [8], and expand this dataset with additional we-scraped data. We will also consider other relevant datasets such as FLORES [3], which includes multilingual data from low-resource languages, and PHINC [2] as well.

2. **Web Scraping:** Extract Hinglish text from popular social media platforms like Twitter, Reddit, and other user-generated content sources to build a more diverse dataset. This data will help capture a wide range of modern conversational and informal Hinglish expressions.

3. **Synthetic Data Generation:** Utilize LLMs (e.g., ChatGPT) to generate conversational Hinglish-to-English parallel data.

4. **Data Enhancement:** Balance underrepresented Hinglish patterns and capture grammar nuances and code-switching behavior by using synthetic samples generate by SMOTE (Synthetic Minority Oversampling Technique) and GANs (Generative Adversarial Networks).

4.2 Phase Two: Model Selection & Fine-Tuning

1. **Model Selection:** Evaluate and select the most appropriate pretrained cross-lingual language models (e.g., mT5-small [7], XLM-RoBERTa-base [1], DistilmBERT [4]) based on their existing capabilities with Indian languages and code-switching scenarios. We will prioritize models with demonstrated strength in Hindi and English representation.

2. **Comparative Analysis:** Conduct benchmark testing of candidate models on small Hinglish samples to determine which architecture best captures the linguistic features of Hinglish code-switching without additional pretraining.

3. **Direct Fine-Tuning:** Skip pretraining and directly fine-tune the selected model on our Hinglish dataset from Phase One, leveraging the model's existing cross-lingual knowledge representations.

4. **Fine-Tuning Strategy:** Fine-tune the model using sequence-to-sequence architectures with attention mechanisms, ensuring that the model can efficiently translate Hinglish text. Leverage **QLoRA** (Quantized Low-Rank Adaptation) for better efficiency in low-resource settings.

5. **Parameter-Efficient Fine-Tuning (PEFT):** Use parameter-efficient fine-tuning methods like **LoRA** (Low-Rank Adaptation) or **Adapters** to reduce the number of parameters needed to be trained, making it feasible for low-resource environments without sacrificing translation quality.

6. **Progress Reporting:** We will submit the Project Status Report on April 3rd, detailing our model selection rationale, fine-tuning approach, initial results, what we have done so far, and what we still have left to complete.

4.3 Phase Three: Evaluation & Prompt Engineering

1. **Performance Evaluation:** Evaluate the model's performance using standard NLP metrics like **BLEU**, **ROUGE**, **METEOR**, and **TER**. These will help assess the accuracy and fluency of Hinglish-to-English translations .

2. **Prompt Engineering Experiments:** Conduct prompt engineering experiments to test the model's performance in zero-shot, one-shot, and few-shot learning scenarios. By varying the prompts, we can determine the most effective strategies for generating high-quality Hinglish responses.

3. **Impact Analysis:** Analyze the impact of different prompts on translation quality to determine which approach works best for maximizing model performance. This step will guide the final prompt engineering strategy.

4. **Final Documentation:** Compile our findings, analysis, and documentation into the Project Final Report by April 24th.

References

[1] Facebook AI. 2020. Xlm-roberta: A multilingual pretrained transformer for 100 languages. Accessed: 2025-03-05.

[2] Mrutyunjay Biswal. 2020. Phinc parallel hinglish corpus - machine translation. Kaggle. Accessed: 2025-03-03.

[3] Open Language Data. 2024. Flores+ dataset. https://huggingface.co/datasets/openlanguagedata/flores_plus. Accessed: 2025-03-05.

[4] Hugging Face. 2020. Distilbert. https://huggingface.co/docs/transformers/en/model_doc/distilbert. Accessed: 2025-03-05.

[5] Alexis Matzopoulos, Charl Hendriks, and Liezl Marais. 2025. BabyLMs for isiXhosa: Data-Efficient Language Models. *arXiv preprint arXiv:2501.03855*.

[6] Raviraj, Kanishk, Anusha, Raunak, Rakesh, Utkarsh, Sanjay, Niranjan, and Eileen. 2024. Adapting Multilingual LLMs to Low-Resource Languages using Continued Pre-training and Synthetic Corpus. *arXiv preprint arXiv:2410.14815*.

[7] Google Research. 2021. mt5-small. https://huggingface.co/google/mt5-small. Accessed: 2025-03-05.

[8] Google Research. 2024. Hinglish-top dataset. https://github.com/google-research-datasets/Hinglish-TOP-Dataset. Accessed: 2025-03-05.

[9] Zhewen Shen, Aditya Joshi, and Ruey-Cheng Chen. 2024. BAMBINO-LM: (Bilingual-) Human-Inspired Continual Pretraining of BabyLM. *arXiv preprint arXiv:2406.11418*.