COE 379L: Final Project Written Report

Sanika Nandpure, Melissa Huang

Project GitHub: https://github.com/sanikanandpure/coe379_project3
Project Slides: slides linked here

I. Introduction and Problem Statement

In an increasingly globalized world, translation has become an integral part of connection and communication. Tools like Google Translate and various AI models have mastered this for common languages, such as Spanish or even Mandarin. However, for some smaller diasporas or communities, these tools are far less ubiquitous. Many state-of-the-art LLMs might demonstrate great general translation capabilities, but fail when addressing languages like Navajo, Yoruba, or Marathi, where pretraining resources and corpora are sparse.

Thus, this project aims to investigate the performance of a fine-tuned English-based model on translations to a lesser-known language, specifically Marathi, with its unique writing style that does not rely on the Latin or Roman alphabet. The objective is to enable the model to produce accurate translations from Marathi to English without having to pretrain an entirely new model, which would be computationally and time-expensive.

We chose Marathi for a few reasons. First, a co-author in this project is fluent in Marathi, allowing us some baseline knowledge of the language in case manual inspections are ever needed. Secondly, we wanted to leverage a lesser-known dialect/language, but needed to balance the amount of proper and cohesive datasets or resources available. We originally wished to conduct our research on indigenous American or African languages, but could not find enough datasets to satisfy the fine-tuning of the model, nor would we be able to verify their accuracy effectively.

II. Data Sources and Technologies

For our project, we will be using this dataset from the Hugging Face Hub (anujsahani01, 2025). It provides around three million pairs of simple English-Marathi translations of short sentences/phrases from various sources. This provides enough samples to pull from for both the fine-tuning and testing phases. Upon an initial inspection of the dataset, it seems to sample quotes from fictional stories (possible television shows or novels), first-person dialogue, news reports, and other online media, providing a large variety of quotations (anujsahani01, 2025).

Our base model is the Microsoft Phi-2 (Microsoft, 2025). We chose this model mainly because it is lightweight, as our project was heavily limited by memory constraints. For its size, it offers relatively strong reasoning and language quality, making it very cost-effective. Additionally, it was immediately available, unlike other models on HuggingFaceHub that required admin approval in order to gain access.

We wrote our code in a Google Colab environment, as this was one of the few ways to get free access to a GPU without an actual hardware setup. It provided an easy interface to rapidly prototype and collaborate on the data preprocessing, fine-tuning, testing, and evaluations.

III. Methods Employed

A. Data Preprocessing

Due to computation constraints, we decided to fine-tune the base model on a dataset of 10k samples instead of the original 3 million. We obtained this subset via random sampling from the original dataset (seed=67 for reproducibility).

Our initial data pre-processing involved first normalizing both the Marathi and English text using NFC (Unicode Normalization Form C) data normalization. NFC ensures that identically looking characters have the same binary code. This is critical for non-English scripts, such as Marathi, where the same characters can be represented in multiple ways in Unicode. For example, Marathi uses "matras", which are marks that combine consonants with vowel sounds. Depending on the way the text is typed out, characters with the matra may sometimes be encoded as a single character, and other times represented as two separate characters (one for the constant, one for the vowel). This can affect the way the text is tokenized, ultimately affecting model training. Thus, we first normalize the text using NFC, ensuring visually/semantically identical characters are encoded the same way. In addition to NFC normalization, we normalized the text whitespace (spaces, tabs, newlines, etc.) to ensure consistency.

Next, we re-formatted the dataset into a form suitable for SFT. Our dataset consisted of pairs of translated English-Marathi text ("`english`", "`marathi`"). We transformed these into "`prompt`" and "`label`", using the encoding structure below:

"prompt": "`Translate from Marathi to English. Marathi: <marathi>. English:`"
"label": "`<english>`"

Next, we tokenized the text using AutoTokenizer, which uses the appropriate tokenizer for a given pre-trained model. In our case, we used Microsoft Phi-2, which uses a Byte Pair Encoding (BPE) Tokenizer (Microsoft, 2025). We added padding on the right to account for variable text inputs; this maintains consistent text vector sizes.

B. Training Methods

For supervised fine-tuning, we utilized Low-Rank Adaptation (LoRA), a form of parameter-efficient fine-tuning (PEFT). This strategy essentially trains "adapters," which are small matrices added to select layers in the base LLM. Rather than modifying all model parameters, LoRA significantly reduces this computational complexity by only training these low-rank matrices. The biggest benefits include faster fine-tuning and easier swapping of adapters if needed. We configured the hyperparameters to keep this fine-tuning process relatively lightweight due to computational constraints. We set the rank of the low-rank matrices to be 32, lora_alpha = 64 (this is the scaling factor; this means LoRA updates are scaled by 64/32 = 2).

Furthermore, we only apply LoRA to the query and value projection matrices in the attention mechanism; this is because attention layers are typically the most important for adaptation as opposed to other layers, such as feed-forward. Finally, we set the dropout to 0.1 to avoid overfitting. This LoRA configuration corresponds to training approximately 0.38% of the total model parameters. We train for 3 epochs with a learning rate of 2e-4, and use an NVIDIA T4 GPU, available through Google Colab.

IV. Results and Analysis

For evaluating our results, we used both Bilingual Evaluation Understudy (BLEU) and BERTScore as our metrics. BLEU is a standard metric for evaluating the translation quality of text, focusing on n-gram structural similarity between machine-translated and ground truth/human-translated text (Papineni et al., 2002). BLEU scores are on a range of 0 to 100, with 0 meaning no overlap between the generated and ground truth text, and 100 meaning the two texts are identical (Papineni et al., 2002).

However, we quickly realized that BLEU may not fully capture the quality of the model's translations, as it penalizes outputs that convey the correct meaning but differ in word choice or sentence structure from the ground truth text. Thus, we additionally compute BERTScore, which accounts for semantic meaning rather than exact word matches (Zhang et al., 2019). This allows for greater flexibility in phrasing, sentence structure, and word choice. This makes BERTScore a better metric for morphologically-rich languages like Marathi, providing a more realistic evaluation of meaning preservation in translation.

BERTScore ranges from 0 to 1 and consists of multiple metrics, including precision, recall, and F1 (Zhang et al., 2019). Precision measures how much of the model's generated translation is semantically aligned with the reference translation. Recall measures how much of the reference translation's meaning is captured by the generated translation. Finally, F1 is the harmonic mean of precision and recall, making it the most informative BERTScore metric.

A. Results Before and After Fine-Tuning

	Base Model (Microsoft Phi-2)	**Base Model + Supervised Fine Tuning**
BLEU	0.39	2.52
BERTScore - Precision	0.75	0.88
BERTScore - Recall	0.85	0.87
BERTScore - F1	0.79	0.87

Table 1. Evaluation of translation accuracy for the base model and the SFT model.

B. Analysis

Our results are summarized in Table 1. The evaluation results demonstrate significant improvement in translation quality following SFT on the base model. We first look at the BLEU metric, which increased from 0.39 to 2.52. However, we note that neither of the two models has a very high BLEU score, likely stemming from differences in sentence phrasing, use of synonyms, etc. between the model-generated and ground-truth English texts. Furthermore, all BERTScore metrics increased with the fine-tuned model. Particularly, the BERTScore F1 metric increased from 0.79 to 0.87, a 10.1% increase in accuracy! Overall, we see that supervised fine-tuning, even on a dataset as small as 10k samples, substantially improves translation quality from Marathi to English.

C. Inference Examples

Below is one inference example pulled from the testing dataset.

```
============================================================
TEST INFERENCE
============================================================

Prompt:
"Input: Translate from Marathi to English:
Marathi: पण त्या मुलाने खरंच मदत केली होती.
English:"

Generated: But they helped each other.

Expected: But has this really helped.
```

D. Limitations

Our most glaring limitation throughout this project was a lack of computational power. First, we tried working on our class-provided virtual machines, but immediately noticed that running on purely CPU power caused our fine-tuning to be extremely slow. Next, we tested training on our local environments, but the issue persisted, given that neither of our devices has a GPU. Finally, Google Colab did work to provide a free GPU environment. However, our progress in training often did not save because running the training portions overnight resulted in disconnected runtimes. Unfortunately, we simply did not have access to enough resources to allow us to achieve our original goal of fine-tuning on 100k of the translation samples.

Another limitation of our findings is domain sensitivity. Although we took a preliminary inspection of the first few pages of the English-Marathi dataset, it would be difficult and nearly impossible to verify the integrity and quality of all 3 million text snippets. The dataset cleaning techniques learned in class are not applicable here, given that the input is inherently qualitative, string-based, and can be morphologically diverse. Since the fine-tuning of the model is highly

dependent on the data, not being able to fully verify the dataset can limit the robustness and validity of our results.

E. Future Steps

While our fine-tuned model has produced promising results, there are multiple possible next steps to explore. First and foremost, if given greater time and resources, we would like to see the base model fine-tuned on a much larger scale, like the original goal of 100k samples. With more data, we would expect a naturally more robust model and more accurate translations. In the same vein, training with more epochs would likely also help improve such results. Other variables to tinker with could be the LoRA arguments and the base model chosen. Finally, we would also want to try fine-tuning with other available Marathi-English datasets to increase general accuracy across multiple data sources. Moreover, this might even allow us to address more cultural nuances, like slang or even dialect variations, which are integral for everyday translations.

Based on the results of this paper, we now anticipate and present some future research directions for other researchers. Given the wide range of text available across the Internet and social media today, one might investigate the robustness of the model against noisy, imperfect text (like slang or typos). Furthermore, if specific domain jargon is involved, we might want to explore how to leverage RAG techniques to maintain the accuracy for highly specialized translations. Such techniques would be critical in specific legal or medical settings where accurate translations are necessary. Another interesting question could involve researching the most common types of translation errors and whether they can be clustered. Finally, our model focuses on a single translation direction (Marathi to English), but exploring fine-tuning on bidirectional translations might lead to even more robust and useful model adaptations.

References

anujsahani01. (2025). *English-Marathi* [Dataset]. Hugging Face. https://huggingface.co/datasets/anujsahani01/English-Marathi

Microsoft. (2025). *Phi-2* [Model card]. Hugging Face. https://huggingface.co/microsoft/phi-2

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). *BLEU: A method for automatic evaluation of machine translation*. In P. Isabelle, E. Charniak, & D. Lin (Eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311–318). Association for Computational Linguistics. https://doi.org/10.3115/1073083.1073135

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2019). *BERTScore: Evaluating text generation with BERT* (arXiv:1904.09675). arXiv. https://doi.org/10.48550/arXiv.1904.09675