Fine-Tuning LLaMA-3.2-3BIT on Bengali Hate Speech Dataset for Sentiment Classification

Abu Mukaddim Rahi
Electrical and Computer Engineering, NSU
Bashundhara R/A, Dhaka, Bangladesh
abu.rahi@northsouth.edu

Mariam Binte Bashir
Electrical and Computer Engineering, NSU
Bashundhara R/A, Dhaka, Bangladesh
mariam.binte@northsouth.edu

Md. Khurshid Jahan
Electrical and Computer Engineering, NSU
Bashundhara R/A, Dhaka, Bangladesh
khurshid.jahan@northsouth.edu

Maher Ali Rusho*
Department of Lockheed Martin Engineering Management, University of Colorado
Boulder, CO 803009, USA
maher.rusho@colordo.edu

Abstract—**In this digital age, analyzing human language and delivering appropriate responses to their performance ability through text classification systems is very basic. This paper proposes a fine-tuning technique in the LLaMA model in natural language processing (NLP) for text classification, which detects "hate" speech and "normal" speech. Though it was a significant challenge for us to train LLaMA models for fine-tuning with a limited annotated dataset and the complexity of detecting hate speech using a low-resource language dataset, results show remarkably higher accuracy. The dataset from Kaggle, which contains 30k Bengali comments that are accumulated from Facebook and YouTube, including 10k marked as "hate speech" and 20k as"normal speech." For instance, the LoRA (Low-Rank Adaptation) with 4-bit quantization for parameter-efficient fine-tuning enables domain-specific adaptation with minimal computational overhead to improve the implementation of our model and enhance it effectively. As a result of model fine-tuning, our system achieved balanced recall, precision, and F1 scores in both classes to significantly reduce misclassification. The aim is to diminish errors, improve accuracy, and extend the approach to other low-resource languages with larger datasets.**

Index Terms—**LLaMA, LoRA, NLP, fine-tuning, classification, data preprocessing**

I. Introduction

In today's world, machines' ability to analyze and influence human behavior and scope has played a great role due to the rapid rise of artificial intelligence [1]. To organize a set of programs to organize input statements automatically is known as speech classification, which has become a very important tool. In today's world, a growing issue is hate speech. It spreads societal tensions, fosters brutality, and even prompts violence. The process of specifying and classifying speech that encourages violence against individuals or groups is known as hate speech detection. Real-life challenges like mental illness, dangers to public safety, and societal threats occur due to the preponderance of online hate speech [2]. It is crucial to enable a safer online environment by detecting hate speech. However, Bengali faces substantial challenges as a low-resource language, while significant languages like English and Chinese receive aid from vast NLP resources, with over 237 million speakers, and as the world's seventh most spoken language [3]. Bengali has no potent dataset. Nevertheless, in automating the classification task of hate speech, the AI model plays a vital role. Nevertheless, AI models play a crucial role in automating the classification task of hate speech. Substantially, by fine-tuning and leveraging pretrained models like LLaMA, typical tasks like Bengali hate speech detection can be accomplished. Fine-tuning critically improves accuracy and computational efficiency and works well with a small dataset [4]. Our aim is to develop a hate speech detection system that specializes in the Bangla language, which will ensure a resilient, robust, and safer online environment. Accordingly, our goal is to diminish the spread of toxic speech, enhance content moderation, and improve people's experience on digital media. The following are the significant contributions of our project:

Enhancing higher Accuracy and Efficiency: Considering the current accuracy, which is quite low, we fine-tuned the LLaMA-3.2-3B-IT model on a Bengali hate speech dataset, emphasizing achieving high accuracy for the detection of hate speech while retaining computational efficiency.

Bangla language explicit resolution: To address individual linguistic arrangements, such as the use of mixed scripts, colloquialisms, and often generic modes where cultural nuances are neglected, we implemented a solution to bridge the gap in hate speech detection resources for an underdeveloped language like Bangla.

II. Literature Review

Yutian Chen et al. [5], offer several approaches to identifying text sources from large language models (LLMs). By directly fine-tuning, as a next token projection task, they reframed the classification scheme. As the backbone for their experiment, they used the Text-to-Text Transfer Transformer (T5). By comparative analysis, it was shown that their approach outperforms the hidden state-based classification approaches. They collected a dataset with 340k text samples from the LLMs and humans, including LLaMA, GPT-3.5, PaLM, and GPT-2, known as OpenLLMText. As the OpenLLMText dataset is biased toward native English speakers This might misclassify non-native English writing and degrade the performance. Renrui Zhang et al. [6] presented a lightweight adaptation method known as LLaMA-Adapter for efficient instruction tuning of LLaMA. This method requires less than one hour for fine-tuning with the 7B frozen LLaMA model with only 1.2M parameters.They also proposed a learnable zero gating, which injects the instructional cues into LLaMA, contributing to a stable training process. They verified their proposed approach with zero-initialized tools for fine-tuning different pre-trained models. Yazhou Zhang et al. [7] proposed an adaptive boosting framework for text classification, designed to specialize LLMs by iteratively fine-tuning the base learners with adjusted training distributions. A comparative analysis of the four benchmarks showed that RGPT outperforms by 1.36% improvement. The many rounds of fine-tuning LLMs in their boosting-based mechanism might lead to high computational costs. Chun Liu et al. [8] explore an effective and simple transfer learning strategy for LLM-based text classification known as LLMEmbed. For training the classifier, they used various lightweight LLMs to improve robustness. When compared to methods based on larger LLMs like GPT-3, the LLMEmbed achieves remarkable performance using the lightweight LLM backbones with low training. Without any fine-tuning and only using 1.8% electricity consumption, 4% model parameters, and 1.5% runtimes, their LLMEmbed achieves good enough accuracy.However, as no fine-tuning is performed on the LLM, the LLMEmbed might need further optimization to improve the performance. Jeyoon Yeom et al. [9], introduce a novel application with advanced generalization capabilities of LLMs in the technology-commercialization field for the specific domain, TC-LLaMA 2. Their model is customized by instruction tuning using the Korean-English Datasets. Matching and generation tasks are also introduced to verify the model's performance. As a result, LLM provides a new approach to developing specialized sectors. In this paper, Aristides Milios et al. [10] explore a pre-trained rank recovery model to bypass the challenges faced by LLM models for the tasks due to the little with many labels due to the limited context window. With the recent LLMs like OPT and LLaMA, they set their stage performance on three common intent classification datasets. On fine-grained sentiment classification, they also surpass fine-tuned performance. They lastly analyzed the model's use by running several ablations. Their experiments are all performed on English-language data and with a small number of runs due to limitations on computing. A method has been offered by Yiming Cui et al. [11] for understanding LLaMA and generating Chinese text by adding 20,000 Chinese tokens to LLaMA's existing vocabulary. To enhance the model's ability, they fine-tune it with Chinese instruction datasets and perform secondary pre-training on Chinese instruction datasets. Hence their result shows proficiency and competitive performance on the C-Eval dataset. Though their model can abandon the corrupt, their models might generate misalignment with human values due to the incapacity of the model to perceive exact outputs in the contexts. According to Hao Yu et al. [12] the selection of models based on task requirements is very important. They highlight some questions in the context of using eight datasets for classification by evaluating entity recognition, misinformation detection, and political party prediction. By fine-tuning, this can contend with the closed-source images, as larger LLMs often lead to improved performance, also in many datasets compared to generative LLMs. RoBERTa can achieve similar or even greater performance. Zongxi Li et al. [13] offer a label-supervised adaptation for LLMs. Their goal is to finetune the model with discriminant labels. Based on LLaMA-2-7B, they evaluate this approach with Label Supervised LLaMA (LS-LLaMA), which can be finetuned on a single GeForce RTX4090 GPU. They finetuned the model using LoRA to minimize loss. Comparatively, in text classification, LS-LLaMA outperforms LLMs tenfold and demonstrates consistent improvements compared to BERT-Large and RoBERTa-Large. Moreover, LS-unLLaMA attains excellent rendition by removing the causal mask from decoders in named entity recognition (NER). Subal Chandra Roy et al. [14] explore a fine-tuned BERT-Bangla model for domain-specific applications. By fine-tuning with the BERT-Bangla model, they developed a question-answering system in a closed domain. They collected the dataset from relevant texts and from the KUET website, which was trained with 2500 question-answer pairs. They achieved promising results by attaining an F1 score of 74.21% and an Exact Match (EM) of 55.26%. Despite having been trained, their model needs further refinements to improve performance for more comprehensive queries.

III. Proposed System

Our project aims to develop an effective and precise Bengali hate speech detection system using advanced machine learning techniques. To ensure data quality, handle imbalanced classes, and improve the model's robustness, we began with the accomplishment of a well-organized Bengali hate speech dataset. Then, we preprocessed it using methods such as text cleaning, tokenization, label encoding, and data augmentation techniques. Afterward, using the standard holdout validation technique, the dataset was split into three technique sets. For training the model, a pre-trained transformer model (LLaMA) was leveraged, as transformer architecture can handle complex language tasks. Subsequently, to adapt the model for the hate speech detection task and facilitate efficient parameter adjustments, a fine-tuning approach was used along with Low-Rank Adaptation (LoRA) without requiring large-scale retraining. Additionally, to control potential overfitting, we used dropout and gradient accumulation techniques. Later on, to ensure it can discern well on unseen data, our model was tested further.

Figure 1 shows the complete workflow of the proposed hate speech detection system, including preprocessing,

processing, fine-tuning, and model evaluation.

A. Dataset Description

For our project, we used the text dataset from the ”Bengali hate speech dataset”. The dataset has been collected from Facebook and YouTube and contains 30k Bengali comments, with 20k Non-Hate speech comments and 10k hate speech comments. Here are comments from 7 different classes to make it diverse, such as Politics, Sports, Religion, Celebrity, Entertainment, Crime, TikTok & memes. The dataset has been labeled in a specific category of the original post. For example, if a video of sports is found, it will comment as “sports.” This helps to analyze and understand the contexts of the data. Also, most of the dataset is in Bengali sentences mixed with the English alphabet, along with some numbers, emojis, and punctuation. We marked the text data as ”hate” or ”normal” speech after uploading the dataset from Kaggle [14]. For fine-tuning the model, we split the original dataset into training, validation, and test sets.

B. Data preprocessing

For fine-tuning the LLaMA model, we will deliver an overview of the system to assemble the Bengali text dataset. To make the data clean, uncluttered, refined, and consistent, data preprocessing is one of the fundamental stages in any machine learning pipeline, which directly impacts the model’s performance. Firstly, we loaded the dataset and split the dataset into training, validation, and testing. To improve the quality of the input of the model and to ensure that only relevant text remains, we removed the irrelevant characters, unessential punctuation, special characters, and white spaces. The next step is tokenization, which is effectively processed by machine learning to split the clean text into smaller units, words, and subwords. Tokenization can analyze the model in a specific manner in the text and helps to break down complicated sentences into simple parts. After tokenization, it is very important to convert categorical labels into numerical values, where the text undergoes label encoding. This makes the text classification task compatible with the model’s requirements, as ML algorithms operate on numerical data. To make the dataset machine-readable, we encoded labels like ”hate” and ”normal” as 0 and 1, respectively, which helps the model predict accurately. We implemented a custom format for preprocessing to organize the data, which plays a crucial role in preparing the dataset for fine-tuning the model. We embedded a template that helps the model learn to specify the task accurately and give the expected output, which is to determine the text as Normal Speech or Hate Speech and return the expected comment label. For example:

Text: "আপনি বেচে আছেন স্যার ?"
Label: ”normal”

This slang word has been uttered to express anger in a way that incites hate toward any individual. Therefore, it is labeled as hate speech.

Text: "এ হালার ঘরের হালা"
Label: ”hate”

This sentence is a sarcastic question but doesn’t promote hate or dehumanize any individual, so it is normal speech.

After completing these preprocessing steps, we split the data for fine-tuning into training, validation, and test sets. To manage and process large datasets, we implemented the datasets library from Hugging Face. Moreover, we used back-translation to address dataset imbalance and data augmentation techniques, which involve converting sentences to another language and back to the original, adding significant differences while maintaining the purpose. As a result, the preprocessing pipeline makes sure that the data is organized, clean, and precise for model training, increasing the performance and applicability of the model. Hence, we generated these steps to increase the data diversity and improve model stability.

Table 01 shows the data splitting after preprocessing, with training data for fine-tuning, validation data for evaluating performance, and test data for final evaluation.

TABLE I: Data Distribution for speech classification model training

Label	Training data	Validation data	Test data
Normal	1200	4000	4000
Hate	6000	2000	2000

C. Model Fine-Tuning

During the time of pre-training, our system leveraged pipelines, models, and tokenizers from the Hugging Face transformers library for the fine-tuning process by using the hate speech detection task. Initially, by utilizing the AutoModelForCausalLM class, the LLaMA 3.2 model is loaded, which is a pre-trained model. We have selected this model because, after fine-tuning, this model can handle casual language modeling tasks for text classification, also allowing easy access to pre-trained models from the transformers library, which do not need to train the model from the beginning.

Additionally, classifying the text into ”hate” or ”normal” categories is fine-tuned as the model is designed for random or autoregressive language modeling tasks and text generation. Besides, we used Low-Rank Adaptation (LoRA) for efficient fine-tuning and to enhance the training process, as it can apply low-rank matrices to the weight matrices of the pre-trained model, which allows for reducing some parameters that are updated and can learn task-specific patterns.

During the process of fine-tuning, our model has been trained on a Bengali-prepared hate speech dataset, where some hyperparameters such as the number of epochs, learning rate, gradient accumulation steps, and batch size are interpreted using the TrainingArguments class. Consequently, to ensure stable merging, we set the learning rate at 2e-4, which is used for large language models for fine-tuning without exceeding the optimal point. In addition, to balance the training time and the model performance, we ran the model for 2 epochs. Furthermore, before executing an update, the model gathers gradients over 8 mini-batches as gradient accumulation has been applied. Hence,

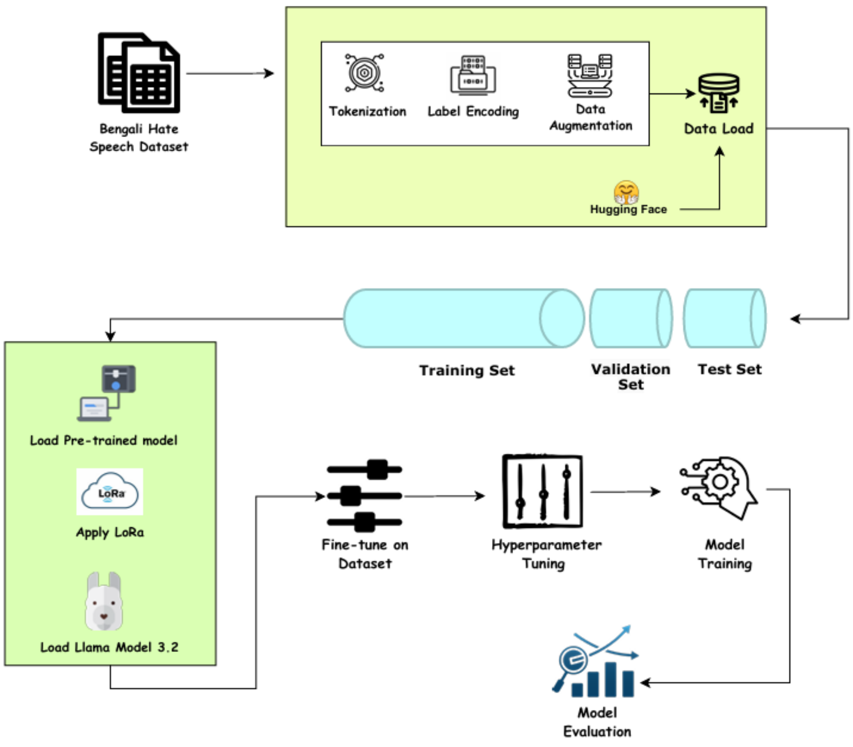

Fig. 1: Workflow of the proposed hate speech detection system

we used this technique especially for preventing memory overflow and for ensuring steady training while working with small batch sizes. In addition, to save memory, we enabled training by storing intermediate activations as gradient checkpointing only when necessary.

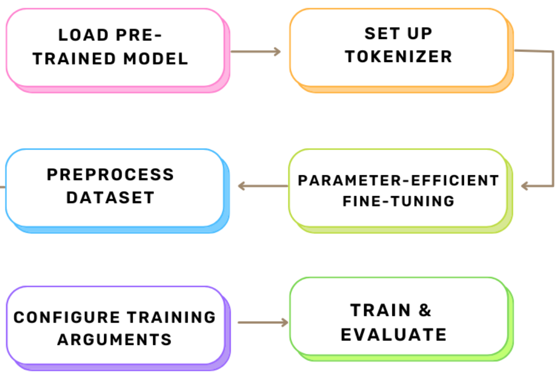

Fig. 2: Shows the fine-tuning steps of our system.

D. Model Architecture

LLaMA 3.2-3b: To handle a vast range of real language processing functions, a wide-ranging language-scale transformer-based language model has been developed, known as LLaMA 3.2-3B. Among the transformer architectures, LLaMA implements multi-layer attention mechanisms to capture long-scale dependencies and related information in text data, which allows the model to generate and operate with remarkable coherence, assurance, and accuracy.

In particular, the architecture of LLaMA 3.2-3B consists of 3 billion parameters, which utilize the decoder-only architecture and optimize modeling tasks such as sequence labeling, classification, and generation of text. Again, for efficiently attending to the input sequence and for focusing on main critical patterns like hate speech detection, the attention mechanism has been designed in the model. Moreover, the model can handle large datasets by using its massive capacity and can perform diverse tasks for learning and generality [16].

For instance, the model architecture starts with our system's Bengali hate speech text tokenization to provide text data to the transformer. Consequently, to convert Bengali sentences into token embeddings, we used a pre-trained tokenizer in the Embedding Layer, where the output comes in a sequence of vectors appearing as the tokens in a sentence. However, we labeled the comment as two classes: ”hate” and ”normal” and used the softmax activation as the final layer, where the label with the highest possibility will be chosen as the prediction. Furthermore, to identify the specific comment, an attention mechanism was used between tokens. In addition, the Feed Forward Layer was used to capture complex patterns like hate speech detection in the text data. Consequently, we applied the RMS normalization after the attention and feed-forward layers to stabilize the model's training.

Figure 3 shows the architecture of the LLaMA model. Which employs feed-forward layers, normalization, and a softmax layer to classify text as 'hate' or 'normal.'

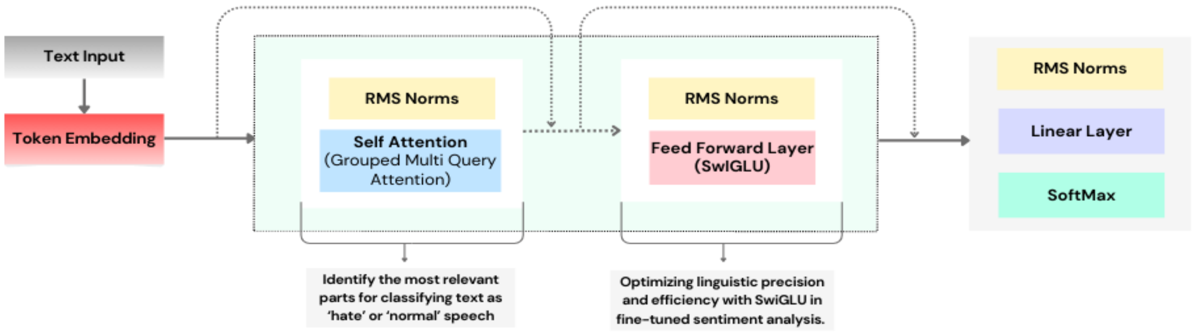
Text Input
Token Embedding
RMS Norms
Self Attention (Grouped Multi Query Attention)
RMS Norms
Feed Forward Layer (SwiGLU)
RMS Norms
Linear Layer
SoftMax
Identify the most relevant parts for classifying text as 'hate' or 'normal' speech
Optimizing linguistic precision and efficiency with SwiGLU in fine-tuned sentiment analysis.

Fig. 3: The architecture of the LLaMA model.

IV. RESULT AND ANALYSIS

A. Model Evaluation

We trained the LLaMA model with 2 epochs and an excellent result. Achieved an exemplary result of 90.4% accuracy after fine-tuning the model. Our system evaluates the model's performance using several key metrics, including accuracy, precision, recall, F1 score, and confusion matrix, explicitly focusing on the "hate" and "normal" categories. Consequently, the model is first evaluated using the initial, untrained LLaMA model. The results show a low accuracy of 0.205, with significant imbalances between the two classes, such as higher accuracy for "hate" speech and low for "normal" speech. Before fine-tuning, we achieved 0.33% precision for hate speech and a recall of 0.46%, which are moderately low. With that, recall is extremely poor for the "normal" class at 8%, despite a high precision of 93%. Thus, this imbalance shows that before fine-tuning, the model was biased toward predicting "normal" incorrectly as "hate." Contrarily, after fine-tuning, we significantly improved across all the metrics. For instance, after fine-tuning, both classes achieve a high precision of 86% for "hate" and 93% for "normal." Besides, for "hate," we achieved 85% recall and 93% recall for "normal." Hence, after observing the graph, we achieved much better results after fine-tuning than before fine-tuning.Figure.3 Shows the precision, recall, and F1 score for both classes, "hate" and "normal" before and after fine-tuning.

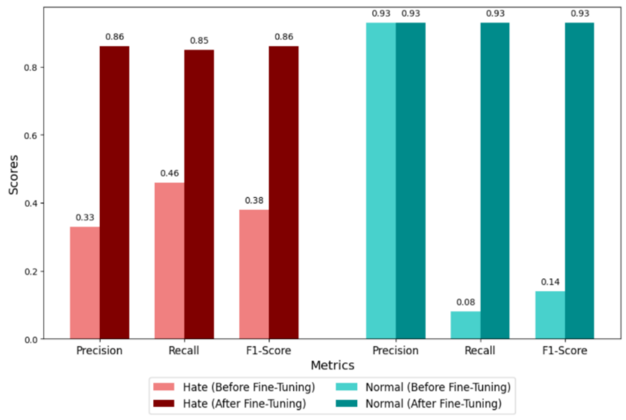

Fig. 4: The precision, recall, and F1 score before and after fine-tuning.

B. Accuracy Comparison Table

The following table highlights the overall accuracy before and after fine-tuning, along with the accuracy for individual classes "Hate" and "Normal." It is worth mentioning that the result after fine-tuning is much higher than before fine-tuning. As we can notice, before fine-tuning, the overall score was quite low, which is 20.5%. Contradictorily, after fine-tuning, the accuracy improves significantly by 90.4% overall. Moreover, the model achieved 46% accuracy for the "hate" class and 7.7% for the "normal" class. On the other hand, after fine-tuning, both classes' accuracy improves profoundly by 85% and 93% for the "hate" and "normal" classes, respectively. Thus, we can say that after fine-tuning, the model's ability to classify both types of speech increased dramatically.

TABLE II: Model Accuracy Before and After Fine-tuning

Accuracy	Before Fine-tuning	After Fine-tuning
Total	20.5%	90.4%
Hate	42.0%	85.2%
Normal	7.70%	93.%

C. Classification Result

For instance, predictions can be correct or wrong. Again, despite achieving high accuracy, we observed some errors during the evaluation. Before fine-tuning, 919 hate speech and 24 normal speech were classified correctly. However, 310 normal speech were misclassified as hate speech, and 1,863 hate speech were misclassified as normal. On the other hand, after fine-tuning, approximately 278 normal speeches were misclassified as hate speech and 294 hate speeches were misclassified as normal speech. Additionally, 1,706 hate speech and 3,722 normal speech were classified correctly. Hence, after fine-tuning, the performance improves significantly with fewer misclassifications of normal speech as hate speech, making it more suitable for the hate speech classification task. The model can now accurately classify both hate speech and normal speech, making it a reliable tool for real-world applications in detecting hate speech in Bengali text. Figure 5 represents the confusion matrix of before fine-tuning and after fine-tuning.

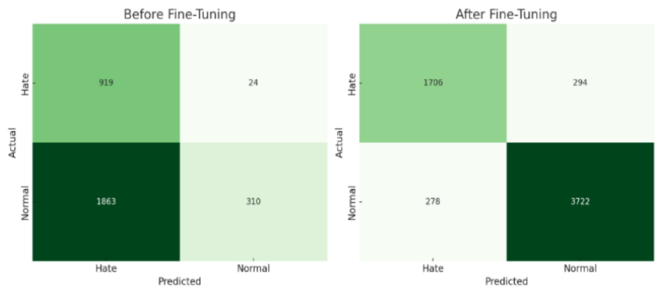

Fig. 5: The confusion matrix for LLaMA before and after fine-tuning.

V. Conclusion

Our main objective of this system is to fine-tune a LLaMA model for sentiment analysis, specifically focusing on classifying Bengali text as hate speech and normal speech. This effort aims to address the crucial need for efficient hate speech detection in regional languages. Our findings aim to enhance understanding and bring ease to everyday activities. Our research is dedicated to making everyday tasks more efficient and effective. The use of our system in Bangla hate and normal speech classification is novel and exclusively present. Our research highlights the potential of advanced language models in identifying harmful content and promoting healthier online interactions. By leveraging this system, individuals and organizations can monitor and manage online platforms more effectively, fostering a safer digital environment. This work will be a foundational consideration for future researchers exploring provincial language processing and ethical AI, offering understanding into fine-tuning large models for specific NLP tasks. The proposed system provides 90.4% accuracy in classifying the text as hate or normal. Thus, our model shows that the proposed method is highly robust in classifying text and is reliable and adaptable due to its low time complexity. Using LLaMA 3.2 in this project highlights its exceptional ability to handle complex NLP tasks, particularly for regional languages such as Bengali. Its adaptability and efficiency during fine-tuning allow it to capture linguistic nuances effectively, making it ideal for sentiment analysis and hate speech detection. This demonstrates the potential of advanced AI models in encouraging more unassailable digital spaces and advancing regional language AI research. If this system were available commercially, it would significantly ease the burden of manual moderation, saving time and resources while ensuring accurate and unbiased hate speech detection. Furthermore, the fine-tuned model's robust performance demonstrates advanced AI's adaptability to regional languages, setting a benchmark for further innovations in linguistic AI research. This project bridges a critical gap in regional language NLP, authorizing communities with tools to combat hate speech and promoting a safer online ecosystem.

References

[1] OpenAI, "Fine-tuning," Oct 16,2024. [Online]. Available: https://platform.openai.com/docs/guides/fine-tuning. [Accessed: Jan. 04, 2025].

[2] Bruegel, "The dark side of artificial intelligence: manipulation of human behaviour," 02 February 2022. [Online]. Available: https://www.bruegel.org/blog-post/dark-side-artificial-intelligence-manipulation-human-behaviour?utm_source=chatgpt.com. [Accessed: Jan. 04, 2025].

[3] M. S. Jahan and M. Oussalah, "A systematic review of hate speech automatic detection using natural language processing," *Neurocomputing*, vol. 546, p. 126232, Aug. 2023, doi: https://doi.org/10.1016/j.neucom.2023.126232.

[4] GeeksforGeeks, "Top 20 Most Spoken Language In The World 2024," 08 Feb, 2024. [Online]. Available: https://www.geeksforgeeks.org/most-spoken-languages-in-the-world/. [Accessed: Jan. 04, 2025].

[5] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, and B. Raj, "Token prediction as implicit classification to identify LLM-generated text," *arXiv preprint arXiv:2311.08723*, 2023.

[6] R. Zhang, J. Han, C. Liu, A. Zhou, P. Lu, Y. Qiao, H. Li, and P. Gao, "LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention," in *Proc. Twelfth Int. Conf. Learning Representations*, 2024.

[7] Y. Zhang, M. Wang, C. Ren, Q. Li, P. Tiwari, B. Wang, and J. Qin, "Pushing the limit of LLM capacity for text classification," *arXiv preprint arXiv:2402.07470*, 2024.

[8] C. Liu, H. Zhang, K. Zhao, X. Ju, and L. Yang, "LLMEmbed: Rethinking lightweight LLM's genuine function in text classification," *arXiv preprint arXiv:2406.03725*, 2024.

[9] J. Yeom, H. Lee, H. Byun, Y. Kim, J. Byun, Y. Choi, S. Kim, and K. Song, "Tc-llama 2: Fine-tuning LLM for technology and commercialization applications," *Journal of Big Data*, vol. 11, no. 1, p. 100, 2024. [Online]. Available: Springer.

[10] A. Milios, S. Reddy, and D. Bahdanau, "In-context learning for text classification with many labels," in *Proc. 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, 2023, pp. 173–184.

[11] Y. Cui, Z. Yang, and X. Yao, "Efficient and effective text encoding for Chinese llama and alpaca," *arXiv preprint arXiv:2304.08177*, 2023.

[12] H. Yu, Z. Yang, K. Pelrine, J. F. Godbout, and R. Rabbany, "Open, closed, or small language models for text classification?," *arXiv preprint arXiv:2308.10092*, 2023.

[13] Z. Li, X. Li, Y. Liu, H. Xie, J. Li, F. Wang, Q. Li, and X. Zhong, "Label supervised llama finetuning," *arXiv preprint arXiv:2310.01208*, 2023.

[14] S. C. Roy and M. M. H. Manik, "Question-answering system for Bangla: Fine-tuning BERT-Bangla for a closed domain," *arXiv preprint arXiv:2410.03923*, 2024.

[15] Kaggle, "Bengali hate speech dataset," Jan. 20, 2021. [Online]. Available: https://www.kaggle.com/datasets/naurosromim/bengali-hate-speech-dataset,[Accessed: Jan. 04, 2025].

[16] Meta, "Model Cards and Prompt Formats for LLAMA 3.2," 2022. [Online]. Available: https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2,[Accessed: Jan. 04, 2025].