IIITDWD-zk@DravidianLangTech-2024: Leveraging the Power of Language Models for Hate Speech Detection in Telugu-English Code-Mixed Text

Zuhair Hasan Shaik[1], **Sai Kartheek Reddy Kasu**[1], **Sunil Saumya**[1], and **Shankar Biradar**[1]
[1]Department of Data Science and Intelligent Systems,
Indian Institute of Information Technology Dharwad, Dharwad, Karnatka, India
(zuhashaik12, saikartheekreddykasu)@gmail.com
(sunil.saumya, shankar)@iiitdwd.ac.in

Abstract

Hateful online content is a growing concern, especially for young people. While social media platforms aim to connect us, they can also become breeding grounds for negativity and harmful language. This study tackles this issue by proposing a novel framework called HOLD-Z, specifically designed to detect hate and offensive comments in Telugu-English code-mixed social media content. HOLD-Z leverages a combination of approaches, including three powerful models: *LSTM* architecture, *Zypher*, and *openchat_3.5*. The study highlights the effectiveness of prompt engineering and Quantized Low-Rank Adaptation (QLoRA) in boosting performance. Notably, HOLD-Z secured the 9th place in the prestigious *HOLD-Telugu DravidianLangTech@EACL-2024* shared task, showcasing its potential for tackling the complexities of hate and offensive comment classification.

1 Introduction

In today's world, nearly everyone possesses a smartphone and easy internet access, making social media an integral part of daily life. Particularly among the youth, there is a keen interest in exploring the latest technologies and an active engagement on social media platforms to connect with diverse individuals and share thoughts. While this connectivity brings numerous positive aspects, such as information exchange and community building, it also introduces challenges. Some individuals exploit social media platforms, asserting their right to freedom of speech, but use it to share private or personal information about others. Moreover, certain users engage in trolling and spread hate on these platforms, revealing the darker side of technology. This misuse presents a significant challenge, especially considering the increasing number of children using the internet and social media, necessitating measures to protect them from harmful and hateful content.

Detecting hate speech online has become a critical but challenging task due to the vast amount of data that requires significant computing power. Furthermore, social media utilizes specific algorithms that identify repeated words in messages, subsequently placing them on the trending list. Unfortunately, this process may lead to the unintentional promotion of controversial content. This rapid spread raises the possibility that hate speech will reach a larger audience and inflict, hurt or offense on those who come across it. One solution to mitigate this issue is the development of machine learning and deep learning-based models capable of effectively detecting hate speech content (Nozza, 2021; Fharook et al.). However, the rising popularity of social media platforms and their expanding user bases have led to the dissemination of content in various languages, often taking the form of script-mixed expressions. Unfortunately, a significant proportion of existing methods are primarily trained to handle monolingual text (Nozza, 2021), neglecting the unique challenges posed by multilingual and code-mixed contexts. There has been only marginal effort directed towards addressing hate speech in low-resource code-mixed text (Biradar et al., 2021; Saumya et al., 2022). Moreover, considering the widespread usage of Dravidian languages across India, it is noteworthy that these languages remain largely unexplored in the context of hate speech detection.

To promote research in this direction, the organisers of *DravidianLangTech-2024*[1] created a shared task for hate speech detection in Telugu-English code-mixed text (B et al., 2024). Our team has participated in the task. We developed three different models, the openchat LLM which achieved a 79% Macro F1 on the validation data, while the Zephyr LLM reached an 80% Macro F1. Additionally, we implemented an LSTM architecture, which yielded

[1] https://sites.google.com/view/dravidianlangtech-2024/?pli=1

Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 134–139
March 22, 2024 ©2024 Association for Computational Linguistics

a 76% Macro F1 on the validation dataset. However, when applied to the test data, Zephyr achieved a 67.39% Macro F1, and the LSTM model achieved a 65.04% Macro F1.

The remainder of the article is organized as follows: Section 2 furnishes details about the proposed architecture. Subsequently, Section 3 presents the findings from the experiments. Lastly, Section 4 provides the conclusion and outlines future research directions.

2 Methodology

This section provides a comprehensive overview of the proposed HOLD-Z framework. Initially, a brief introduction to the problem statement and dataset is presented. Subsequently, the approaches employed to address this challenge are discussed.

2.1 Task and Data

The proposed work considers data from the *HOLD-Telugu DravidianLangTech@EACL 2024* shared task (Priyadharshini et al., 2023). The shared task organizers released the data in train and test, comprising 4,000 and 500 comments in each stage, respectively. Assuming our training dataset as $D = [s_1, s_2, ..., s_n]$ of length *n*, where s_1 represents sentence 1, s_2 represents sentence 2 and similarly s_n represents sentence n (≤4000) in our dataset. According to the organizers, the dataset were collected from YouTube comments (B et al., 2024). The objective of the task involves sentence-level classification of each Telugu-English code-mixed social media comment into hate or no-hateful categories. The detailed specifications of the dataset are provided in Table 1.

	Hate	Non-hate	Total
Train	1,939	2,061	4,000
Test	250	250	500

Table 1: Data distribution

Code-mixing presents a unique challenge for our models. While translating directly to English might seem straightforward, it often misses the nuanced meaning. Take the Telugu-English sentence (example from training set) *Students tho adukovtam thappu*, which translates literally to *Playing games with students is wrong*. However, the intended meaning is far deeper: *Playing with students' lives is wrong*. This context-dependence makes accurate identification of hate and offensive comments in code-mixed text a complex task, pushing our models to truly understand the underlying intent.

2.2 Context focused

To understand context, *Model 1* utilizes context-aware embeddings and a multi-layered LSTM network. Embeddings capture contextual information, which is then fed into two bidirectional LSTMs followed by a standard LSTM for deeper context analysis and filtering. This forms the basis for the baseline score. The model architecture consists of an embedding layer, followed by two bidirectional LSTM layers and one standard LSTM layer, all connected to a single-neuron classifier with sigmoid activation for hate/non-hate prediction. Figure 1 showcases the complete pipeline.

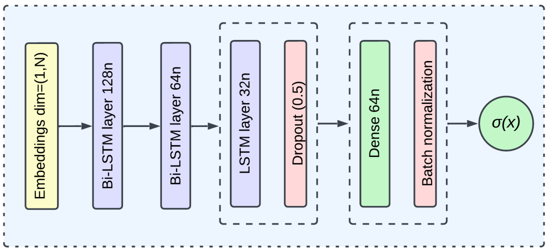

Figure 1: Model 1: Context focused LSTM Network.

We further explored alternative embedding approaches by replacing the initial layer (of *Model 1*) with pre-trained options like BERT(N=768) (Kenton and Toutanova, 2019), Hate-BERT (Tommaso-Caselli and JelenaMitrovic, 2021), mBERT(Kenton and Toutanova, 2019), and OpenAI's ada-embeddings-002 (N=1536) [2]. While the overall architecture remained unchanged, this experiment yielded notable gains in performance on the test data.

2.3 7B-LLMs cluster

Model 2 employs the capabilities of 7B LLMs, recognized for their proficiency in both Telugu and English. These models, with their advanced language processing abilities, can effectively capture the contextual intricacies of sentences, as outlined in the problem statement. Figure 2 presents an architectural overview of the *Model 2*. The implementation utilizes several prominent 7B LLMs, such as *"Llama-2-7b-chat-hf"*, *"Llama-2-13b-chat-hf"* (Touvron et al., 2023), *"Mistral-7B-Instruct-*

[2] https://arxiv.org/abs/2303.08774

v0.1" (Jiang et al., 2023), *"zephyr-7b-beta"* (Tunstall et al., 2023), and *"openchat_3.5"* (Wang et al., 2023).

2.3.1 Prompt engineering

Prompt engineering lies at the heart of successful LLM interaction. Figure 2 illustrates how each input (s_n) is crafted into a precise prompt. Let's dissect an example:

Input: "*Students tho adukovtam thappu*".
The processed prompt (for *Zephyr-7B-beta* LLM) looks something like this:

- System prompt: Defines the LLM's role and expected behavior within the interaction, guiding its response.

  <|*system*|> *You are an expert in sentiment analysis.*

- Hypothesis prompt: Presents a statement and requests the LLM to evaluate its truthfulness, promoting critical thinking.

  <|*hypothesis*|> *The sentence "Students tho adukovtam thappu" contains hateful or offensive content.*

- Assistant prompt: Provides an incomplete statement or scenario, inviting the LLM to complete it creatively, encouraging open-ended generation.

  <|*assistant*|>*The given hypothesis is..*

Each model leverages a specific prompt template for optimal performance. We demonstrate the *Zephyr-7B-beta* template for illustrative purposes. Through rigorous experimentation, we discovered that *openchat_3.5* achieves superior results with the *Zephyr* prompt. Notably, other LLMs utilize their own prompt templates.

2.3.2 Importance of Assistant prompt

The proposed LLM operates as a text completion model. Given an input sentence, it predicts the most likely next word (within its vocabulary) based on softmax probabilities. This predicted word is appended to the input, forming a new input for subsequent predictions. The process iterates until reaching the end-of-sequence (<eos>) token.

Leaving the assistant prompt (*(<|assistant|>The given hypothesis is..)*) incomplete leads the LLM to predict probabilities for all words in its vocabulary, with a bias towards terms like "True", "False", "right", "wrong", and so on. LLM selects the word with the highest probability as next word. However, we have now refined the output layer to solely consider two options: 0 (False hypothesis) or 1 (True hypothesis). This simplifies the LLM's learning process and facilitates a more definitive answer.

2.3.3 Fine Tuning with QLoRA

Given the constraints of catastrophic forgetting and computational limitations, we are unable to conduct the complete training of LLM's (7B's). Instead, we have chosen Quantized Low-Rank Adaptation (QLoRA). This approach involves quantizing the model during inference and subsequently applying LoRA. In LoRA, we freeze the model parameters and add an extra low-rank matrix next to the attention layer weights, instead of training all parameters. This significantly reduces training time and memory needs, while often leading to better performance compared to traditional fine-tuning techniques (Hu et al., 2021).In our case all models are inferred and trained in FP16 (Half-precision, float16). After extensive experiments we identified hyper parameters which worked for the proposed model are mentioned in Table 2. We also trained high-performing models in FP32 (float32), utilizing our substantial RAM and computing capabilities, with the support of 3×32G Nvidia V100 GPU's.

Hyper parameter	**Value**
Rank (LoRA config)	16
LoRA Alpha (LoRA config)	64
Dropout (LoRA config)	0.1, 0.2
Learning Rate	2×10^{-5}
Learning Rate Scheduler	Constant
adam_beta1	0.9
adam_beta2	0.999
adam_epsilon	1.000×10^{-8}
rms_norm_eps	1.000×10^{-5}

Table 2: Hyper parameters for Training 7B's

3 Results

In this section we give extensive study results conducted on different models and different approaches to the problem statement that involves context focused approach and 7 Billion-parameter models (7B's).

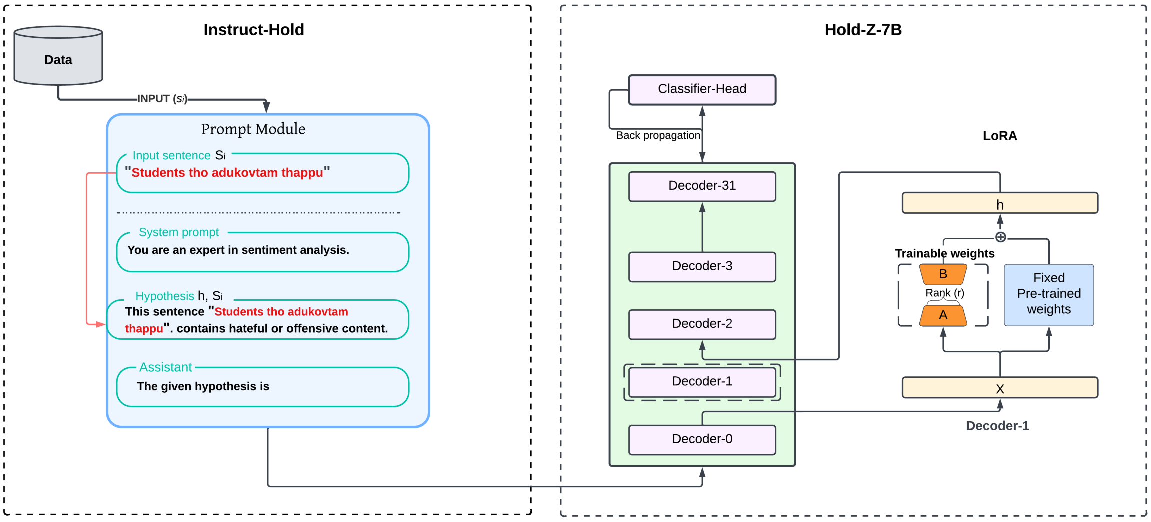

Figure 2: The overview of HOLD-Z framework

3.1 Context Focused approach

In Model 1, our exploration commenced with BERT variants serving as the baseline score. Subsequently, we delved into several cross-lingual pre-trained models to generate embeddings. The experimental findings are illustrated in Table 3. Notably, the Keras embedding layer outperformed all other models according to the results presented in Table 3.

Embedding model	Macro-F1
Keras	68.17
mBERT	60.13
XLM-Roberta	65.46
Telugu-BERT	63.94
Indic-BERT	58.11
OpenAI	64.83

Table 3: Macro-F1 scores with different embedding models

The performance of the Keras model can be attributed to the trainability of the Keras embedding layer. This feature enables the layer to autonomously comprehend and acquire optimal contextual representations for input sentences in code-mixed text. In contrast, the other models rely on pre-trained embeddings. This is the rationale behind our belief that Keras surpassed the performance of all other models.

3.2 7Bb's

In Model 2 (HOLD-Z) , we've conducted experiments with various models, exploring different hyperparameters, including *target_modules*. After examining LoRA configurations, we concluded that including all seven parameters in the *target_modules* along with optimal rank yielded better results. Further, we understand higher the value implies a greater number of trainable parameters and increased computational demands. To address this concern, we settled on the optimal community-consensus value of r=16, and to train all seven prameters in target_modules.
We observed minimal changes when altering dropouts beyond 0.3. Consequently, we focused on experimenting with dropout rates of 0.1 and 0.2, which ultimately led to the best outcomes as illustrated in Table 4 .

Model	D 0.1	D 0.2
Llama-2-7b-chat	43.52	64.95
Mistral-7B-Instruct-v0.1	72.39	72.98
Zephyr-7b-beta	73.98	73.79
openchat_3.5	72.80	74.62
Llama-2-13b-chat	**75.27**	71.94

Table 4: Macro-F1 of LLM's with Dropouts (D) in FP16 QLoRA

3.3 7B's on full precision

Training models with full precision takes a lot of time. Because of this, we set the configurations based on the best models to work around the computational limits. For instance, when we trained QLoRA using FP16, we achieved the best Macro F1 score of 75.27 with llama-2-13b-chat. We then use the same model configurations and train it in FP32 without quantization. Interestingly, the outcomes show a high similarity in model-to-model scores as illustrated in Figure 3.

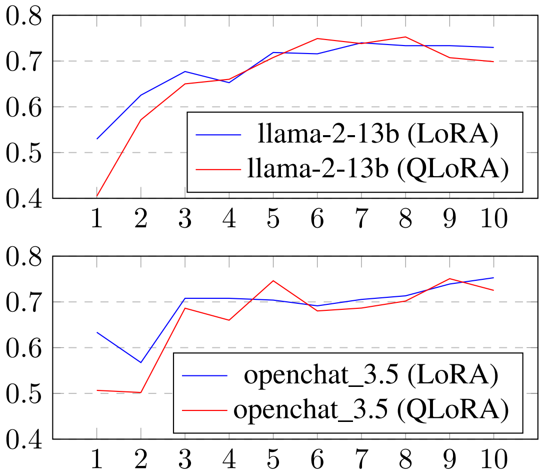

Figure 3: Macro-F1 scores with LoRA vs QLoRA for different LLM's and epochs.

The openchat_3.5 LLM outperformed numerous models and surpassed llama-70b-chat by 8 points in the lmsys-chatbot-arena [3], which is the reasoning behind considering openchat_3.5 in the proposed work. openchat_3.5 proved it self again by outperforming 13b model, and stood top of the board as illustrated in Table 5.

Model	**Macro-F1**
Llama-2-7b-chat	73.77 (D 0.2)
Mistral-7B-Instruct-v0.1	72.96 (D 0.2)
Zephyr-7b-beta	73.55 (D 0.2)
openchat_3.5	**75.28 (D 0.2)**
Llama-2-13b-chat	73.99 (D 0.1)

Table 5: Macro-F1 of LLM's with FP32 LoRA

4 Conclusion and Future work

In conclusion, our study introduced the HOLD-Z framework for Telugu-English code-mixed social media comments classification. Leveraging context-focused approaches and 7B LLMs, particularly openchat_3.5, proved its effectiveness. The exploration of prompt engineering and fine-tuning with QLoRA demonstrated promising results. The proposed work and model are added to Github[4] and HuggingFace[5] respectively. Future work involves refining model architectures, exploring additional embeddings, and addressing the evolving challenges of code-mixed text classification. The proposed work achieved 9th rank in the *HOLD-Telugu DravidianLangTech@EACL-2024* shared task signifies the potential for further advancements in this domain.

References

Premjth B, Bharathi Raja, Prasanna Kumar Kumaresan, Saranya Rajiakodi, Sai Prashanth Karnati, Sai Rishith Reddy Mangamuru, and Janakiram Chandu. 2024. "Findings of the Shared Task on Hate and Offensive Language Detection in Telugu Codemixed Text (HOLD-Telugu)". In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, Malta. European Chapter of the Association for Computational Linguistics.

Shankar Biradar, Sunil Saumya, and Arun Chauhan. 2021. mBERT based model for identification of offensive content in south Indian languages.

Shaik Fharook, Syed Sufyan Ahmed, Gurram Rithika, Sumith Sai Budde, Sunil Saumya, and Shankar Biradar. Are you a hero or a villain? A semantic role labelling approach for detecting harmful memes.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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*.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Debora Nozza. 2021. Exposing the limits of zero-shot cross-lingual hate speech detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 907–914.

[3] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

[4] https://github.com/Zuhashaik/HOLD-Z

[5] https://huggingface.co/zuhashaik/HOLD-Z/tree/main

Ruba Priyadharshini, Bharathi Raja Chakravarthi, S Malliga, Subalalitha Cn, SV Kogilavani, B Premjith, Abirami Murugappan, and Prasanna Kumar Kumaresan. 2023. Overview of shared-task on abusive comment detection in Tamil and Telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 80–87.

Sunil Saumya, Vanshita Jha, and Shankar Biradar. 2022. Sentiment and Homophobia Detection on YouTube using Ensemble Machine Learning Techniques. In *Working Notes of FIRE 2022-Forum for Information Retrieval Evaluation (Hybrid). CEUR*.

ValerioBasile TommasoCaselli and MichaelGranitzer JelenaMitrovic. 2021. HateBERT: Retraining BERT for Abusive Language Detection in English. *WOAH 2021*, page 17.

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*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*.