Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 1, pp. 107-119
DOI : 10.47974/JIOS-1856

Hate content detection through prompt engineering : A pioneering attempt

Shillpi Mishrra *
Anup Kumar Barman †
Apurbalal Senapati §
Department of Computer Science and Engineering
Central Institute of Technology
Kokrajhar 100190
Assam
India

Abstract

The detection of hate speech is a difficult task and an active study subject at the moment. Effective hate speech identification is more important than ever in a time when a sizable section of the populace uses social media to express their emotions, feelings, rage, and anxiety as well as to look for protection. Conventional approaches to detecting hate content typically rely on hate-related keywords or utilize a labelled dataset to train a learning model. However, hate speech often stands out more due to its context. Conversely, training a learning model requires a substantial amount of labelled data. This paper explores a pioneering approach to hate content detection in the Bengali language through the innovative use of prompt engineering. We develop and apply dynamic prompts that improve the model's comprehension and recognition of complex hate speech in a range of circumstances by utilizing the power of sophisticated language models. Using the publicly accessible HASOC 2023 dataset, we tested ChatGPT in our experiment and compared its performance to the state-of-the-art.

Subject Classification: *68T50 Natural language processing.*

Keywords: *Prompt engineering, Hate speech, Bengali language, Large language model, Social media.*

* *E-mail:* **shllpimishrra91@gmail.com (Corresponding Author)**
† *E-mail:* **ak.barman@cit.ac.in**
§ *E-mail:* **a.senapati@cit.ac.in**

1. Introduction

Hate speech is described by the Cambridge Dictionary as "public speech that expresses hate or encourages violence towards an individual or group on the basis of race, religion, sex, or sexual orientation." [2]. The determination of actual hate speech depends on the context of the text. Sometimes, it is subjective and depends on various external factors. Therefore, detecting hate content automatically is challenging. A significant section of the population uses numerous social media platforms these days. According to the Digital Overview Report 2024, social media users on average spent 2 hours and 23 minutes daily[1]. On social media, individuals are liberated to convey their feelings, emotions, frustrations, and even to express protest. As a result, individuals or groups can disseminate hate speech both deliberately and inadvertently.

Progress in hate speech detection has accelerated significantly in recent years. Initially, people employed a keyword-matching approach [1], but later shifted towards a learning-based approach [3, 4]. Even low-resource languages are endeavouring to integrate pre-trained models [5]. However, it remains challenging due to the lack of tagged data. This circumstance inspired us to employ prompt engineering for hate speech detection, eliminating the need for tagged data. This paper begins with a general overview introduction to prompt engineering, followed by an exploration of various prompt techniques. Subsequently, it outlines our experimental methodology, provides a description of the data utilized, and concludes with the results and findings.

2. Early Work

Recent papers highlight advancements in hate speech detection using large language models (LLMs). These models have significantly improved NLP tasks but are computationally expensive and require extensive labeled data. To address this, zero-shot and few-shot learning techniques have emerged as efficient alternatives. One significant difficulty is identifying hate speech in less-resource, mix-code languages, where LLMs have demonstrated promise. For instance, researchers analysed misogyny in Hinglish by collecting 100 YouTube comments and applying weak labels to reduce annotation effort. They tested zero-shot, one-shot, and few-shot prompting methods, comparing results with human annotations. Few-shot prompting with GPT-3 and zero-shot classification using the BART model yielded the best results [6]. In the paper [7], it focuses on hate

speech detection datasets across various domains, emphasizing the advantages of zero and few-shot learning over supervised methods. It evaluates generative models like BLOOM, Llama-2, and T5, showcasing their ability to learn from minimal data and generate text. By analyzing performance on datasets in English and Spanish, the study highlights their cross-lingual and cross-domain capabilities. These findings demonstrate how generative models address data constraints while maintaining effectiveness. Nevertheless, domain-specific modification and applications in low-resource languages are not explored in the paper. Additionally, it doesn't look closely at fine-tuning algorithms for detecting hate speech in real time. With little labeled data, the study [8] examines how well zero-shot learning detects hate speech in three different languages. It uses eight benchmark datasets to assess several large language models and verbalizers. Results show that prompting can compete with or outperform fine-tuned models, particularly with bigger language models, underscoring the need for fast selection. This approach is valuable for low-resource languages. The study emphasizes the roles of prompts and models in achieving higher accuracyin hate speech detection. It also calls for further research into better verbalizers for instruction-tuned models, more diverse datasets, and the evaluation of closed-source models like GPT to address reproducibility challenges.

3. Prompt Engineering and Large Language Model (LLM)

In the realm of artificial intelligence, LLMs have opened a new frontier in the field of language processing. Ideally the LLM model contains all the available text from the web and it can serve human-like communication. Practically the models have been trained on vast volumes of text data from the web and it acts on user query. Figure 1 depicted the generic overview of prompt engineering technique. Prompt engineering was once restricted to the trial-and-error stages and has since developed in

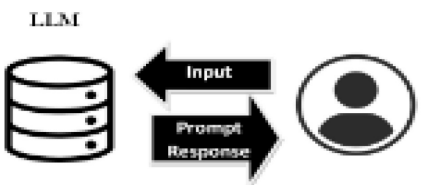

Figure 1

Schematic diagram of prompt engineering

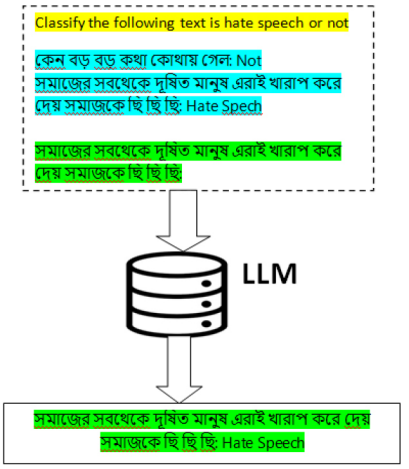

Figure 2

An overview of hate speech detection. The prompt consists of a task description marked in yellow, training examples marked in the sky, and test instance marked in green

tandem with LLMs [9]. Later on, it changed as LLMs developed and gained popularity thanks to the efforts of numerous researchers. We investigate the most recent state-of-the-art prompt approaches and how they are applied in different LLMs, with a primary focus on prompt methodologies.

Here we are dealing with a specific problem i.e. detection of hate speech using prompt engineering. Figure 2 showcases a successful example of hate speech identified within the input text.

4. Principles of Prompt Engineering

In order to elicit desirable answers, prompt engineering entails efficiently interacting with a big language model. Big language models (LLMs) have proven to be exceptionally effective in a number of areas, including reasoning, code production, and answering questions. Nevertheless, creating the best prompts or instructions is crucial to utilizing these powers. However, this procedure can occasionally be complex and confusing for end users, which is why the research community is concentrating on developing prompt design guidelines. In response to this requirement, Basharat et al.[10] have produced the following list of 29 thorough principles, which are divided into five groups. Figure 3 defines

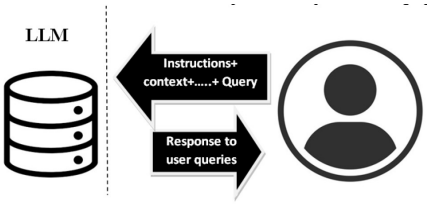

Figure 3

Architecture of prompt engineering

the architecture of the prompt engineering where if user send queries, instructions and contexts to the LLM model, then model will response that queries through prompts.

5. Prompt Engineering Techniques

Various prompting techniques are available, evolving from simple to complex methods. The latest survey paper reports 29 different prompting techniques. Since our experiment is conducted with the three basic prompt techniques, this section focuses exclusively on those techniques.

5.1 *Zero-Shot Prompting*

Zero-shot prompting refers to providing a task or prompt to a language model (LLM) without prior training or specific examples. It relies solely on the model's pre-trained parameters and the information in the prompt to generate a response. Wei et al. [11] highlight that instruction tuning, a form of fine-tuning, can enhance zero-shot prompting. Christiano et al. [12] introduced reinforcement learning to align model outputs more closely with human preferences. This adaptable technique requires no prior data training and is illustrated in Figure 4, where prompts are directly sent to the model for responses.

5.2 *Few-Shot Prompting*

Zero-shot prompting shows potential but struggles with complex tasks. Few-shot prompting improves performance by providing example input-output pairs to guide the model. This context helps the model understand the task better, as noted by Min et al. [13]. However, few-shot prompting is less effective for challenging reasoning problems. Figure 5 illustrates few-shot prompting, where users provide intermediate reasoning steps in prompts, and the model responds accordingly.

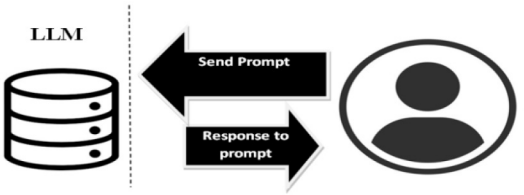

Figure 4

Diagram of Zero-shot prompting

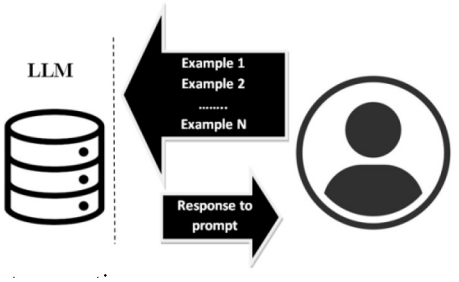

Figure 5

Diagram of Few-shot prompting

5.3 *Chain-of-Thought (CoT) Prompting*

Wei et al. [14] introduced Chain-of-Thought (CoT) prompting to tackle complex reasoning tasks. This method involves breaking reasoning into a series of intermediate natural language steps, enhancing the clarity of the thought process. They demonstrated this approach using a scenario in Figure 6, comparing standard prompting to CoT prompting. The results

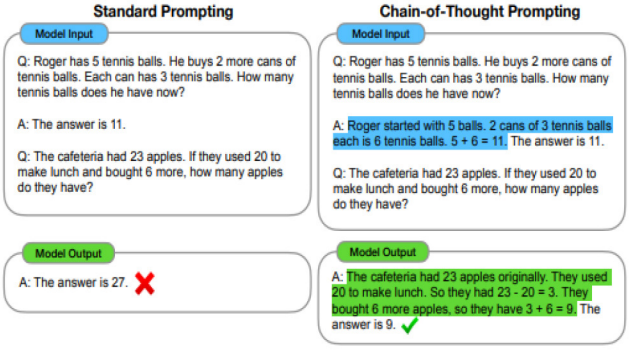

Figure 6

Reasoning processes of Chain-of-thought prompting are highlighted.

showed significant improvements in large language models' performance on tasks requiring reasoning, computation, and judgment. By mimicking human thinking in input prompts, CoT facilitates intricate problem solving. The goal is to increase the reasoning skills of language models through structured input design.

6. Data set and Implementation

In this experiment, we compare the effectiveness of three prompt engineering methods: Chain-of-Thought (CoT), Few-Shot, and Zero-Shot. The experiment was implemented using the OpenAI Python library (GPT 3.5 turbo model), leveraging the GetPass library to securely retrieve the API key.

6.1 *Dataset*

Our experiment utilized the HASOC 2023 dataset [15]. The original dataset encompassed text in Assamese, Bengali, and Bodo languages. The data consists of binary classifications: hate speech (labeled 1) and non-hate speech (labeled 0) (see Figure 7 for details). However, for this experiment, we focused solely on the Bengali dataset. The Bengali dataset consisted of 1,601 data points, with 641 labeled as hate speech and 960 labeled as non-hate speech. Figure 7 illustrates a sample of the tagged dataset.

The prompt instructions we offer the LLMs that serve as our categorization model are depicted in Figure 8. In order to make it easier to analyze and extract the LLMs' response, these instructions are set up for the LLMs to react with either "Hate Speech" or "Not hate speech." To

Text	Hate/Not
ভোটটা নেওয়ার সময় বাড়ি বাড়ি যাবেন আর চাকরিটা চামচাদের দেবেন।	Hate
কালীর হাত আছে পা আছে কারণ তার অস্তিত্ব আছে। যার কোনো অস্তিত্ব নেই কেবল কল্পনার দ্বারা সৃষ্ট একটি প্রতিষ্ঠান তারা কী বুঝবে তার অস্তিত্বের প্রমাণ	Not Hate
আপনার কথার ভাবধারা খুব ভালো। কিন্তু প্রকৃত বাস্তব হোল " রদ্দুর রায়"। এটাই সত্যি।আর একটা বিষয় হলো -- আপনি কতটুকু বাস্তবের সম্মুখীন হচ্ছেন এটা বড় বিষয়। আমরা সাধারণ মানুষ,আর ভদ্র লোক আমাদের দেখেছেন বা রোজ দেখছেন এবং শুনার দুঃখ হচ্ছে -- এটাই সবচেয়ে বড় বিষয়। এটা continue হওয়া উচিত। যদি সমাজ-পতিদের টনক নড়ে । ধন্যবাদ ---- ।	Not Hate
এই শুওরের বাচ্চার ফাঁসি দেওয়া দরকার	Hate

Figure 7

Sample Bengali tagged dataset

mark the start and finish of an instruction, specific tokens are specified by each model, which has its own input or instruction format.

7. Results and Discussion

Using the HASOC dataset, our study assesses the effectiveness of zero-shot, few shot, and chain-of-thought prompting approaches for the identification of hate speech. Two phases of experiments were conducted, applying all three techniques (zero-shot, few-shot, and chain-of-thoughts classification) to the entire dataset. Results of the classification models are presented in Table 1, highlighting the different levels of performance. Few-shot, zero-shot, and chain-of-thought approaches demonstrate distinct strengths in guiding GPT-based models. The study emphasizes the comparative impact of different prompts on hate speech detection. In Zero-shot prompting, we find the F1-score is 0.79 for Bengali hate speech

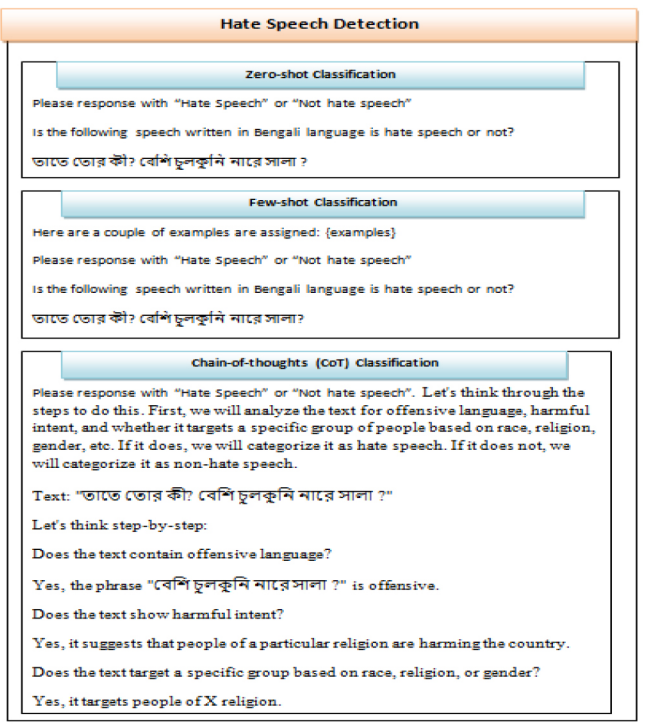

Figure 8

Our study's prompts were designed to prompt the LLMs for every classification job

Table 1

Different prompt techniques performance to detect hate speech

Prompt Techniques	Precision (t)	Recall(t)	F1-Score(t)
Zero-shot prompting	0.781	0.8	0.79
Few-shot prompting	0.775	0.79	0.782
Chain-of-Thoughts prompting	0.792	0.815	0.802

detection using the LLM model. Out of the three prompt techniques, the few-shot prompt performs the worst, especially when it comes to F1-score (0.782). With precision of 0.792, recall of 0.815, and F1 score of 0.802, Chain-of-thought prompt technique performs better than the other prompts on every parameter. This shows that the model's comprehension and identification of hate speech are much improved when a structured line of reasoning is included in the prompt. Furthermore, the model's learned knowledge base may be better utilized to make contextual decisions if the hate speech detection problem were divided into many parts. However, the HASOC 2023 benchmark reports a best F1 score of 0.77 for Bengali hate speech detection using a pre-trained Transformer model [15]. The greatest accuracy recorded was 79 percent, which is not promising. A macro-average is used to compute precision, recall, and F1-score.

In the second experiment, we applied the three prompting technique to each sentence in the same dataset and measured the performance of each sentence that is displayed in Table 2. It is worth noting that the F1-score only fluctuated between0.747 and 0.79. It makes sense to draw the conclusion that small changes to the language of prompts have no appreciable impact on performance.

Our experiment focuses on binary classification, extendable to multi-class classification for detecting hate speech across various categories like religion, sexuality, race, politics, and other categories. Multi-class detection requires refined prompts and real-world context beyond sentence-level understanding. A zero-shot prompt was designed for multi-class classification, but existing prompts often yield incorrect results. This highlights the need for improved prompt techniques to enhance performance. An example of a zero-shot prompt for multi-class classification is provided.

Table 2

Performance of each sentence using prompting techniques to detect hate speech

Prompts (with example)	Precision (t)	Recall(t)	F1-Score (t)
1. **Zero-shot Prompt:** Is the following speech written in bengali language is hate speech or not:ভাষণ বন্ধ করুন। আপনাদের মানুষ খুব ভালো করে চেনে।	0.771	0.759	0.781
2. **Few-shot Prompt:** Here are a couple of examples are assigned: examples কেন বড় বড় কথা কোথায় গেল: Not Hate Speech সমাজের সবথেকে দূষিত মানুষ এরাই খারাপ করে দেয় সমাজকে ছি ছি ছি: Hate Speech Please response with "Hate speech" or "Not hate speech" Is the following speech written in bengali language is hate speech or not: ইয়ে মাথায় ওঠেনি। এগুলো সবই পরিকল্পনা করে বিদ্বেষ ছাড়ানো	0.753	0.743	0.747
3. **Zero-shot Prompt:** Is the following speech written in bengali language is hate speech or not: সুবিধাভোগী চটি চাটা বুদ্ধিজীবি	0.783	0.768	0.758

Example 1: Prompt: "Classify the following statement into one of the following categories: 'Religion', 'Race', 'Sexuality', 'General Hate Speech', or 'Not Hate Speech': '**বাবার এক্সট্রা সন্তান**' .

Output: "General Hate Speech"

Example 2: Prompt: "Classify the following statement into one of the following cat egories: 'Religion', 'Race', 'Sexuality', 'General Hate Speech', or 'Not Hate Speech': '**যখন বাংলাদেশে হিন্দুহত্যা হচ্ছিলো তখন সব বুদ্ধিজীবী কোথায় ছিল** ' . '. Output: "Religion"

Our next step will be a comprehensive error analysis to identify the strengths and weaknesses of our system. While a full analysis is ongoing, we have begun exploring potential areas for improvement based on observations in Figure 9.

In Figure 9, the study highlighted errors in hate speech classification using prompting techniques in a Bengali dataset. It observed instances where chain-of-thought prompting gave incorrect results, while zero-shot

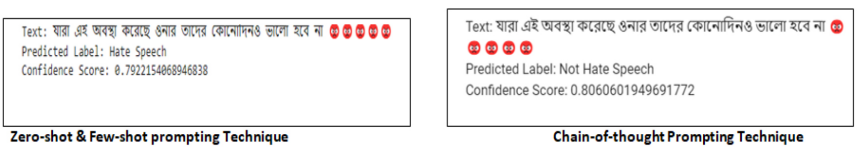

Figure 9

Hate speech detection using Zero-shot, Few-shot and chain-of-thoughts prompting technique in GPT 3.5 turbo

and few-shot prompting techniques provided accurate classifications. For example, the Bengali text "যারা এই অবস্থা করেছে ওনার তাদের কোনোদিনও ভালো হবে না" was misclassified as not hate speech using chain-of thought but correctly identified as hate speech with zero-shot and few-shot techniques. When translated to English, the outcomes remained consistent, revealing the limitation of CoT prompting. The analysis showed that subtle biases or indirect hateful expressions pose challenges for existing prompting methods. We have conducted an error-analysis where we have identified such instances and it is found that it is needed. It needs some dedicated specialised prompt to solve such instances.

8. Conclusion

This work represents one of the pioneering attempts at applying prompt engineering for hate speech detection. We evaluated our approach using a publicly available dataset. Although the results and analysis highlighted areas for improvement, they also demonstrate that prompt engineering shows significant potential as a competitive method for hate speech detection, comparable to dedicated systems. In order to identify the weaknesses of the system, an error analysis was conducted. This analysis revealed two primary areas for improvement: (i) in multi-class hate speech detection, the system required more refined prompts and, in many cases, real-world and con temporary knowledge rather than relying solely on sentence-level context; and (ii) it also needed more specialized and fine-tuned prompts to achieve higher accuracy.

References

[1] P. Mandal, A. Senapati, and A. Nag, "Hate-Speech Detection in News Articles: In the Context of West Bengal Assembly Election 2021," Pattern Recognition and Data Analysis with Applications, *Lecture Notes in Electrical Engineering*, Springer, vol. 888, pp. 247-256 (2022).

[2] Oxford English Dictionary, s.v. "hate speech (n.)," Mar. (2024). [Online]. Available: https://doi.org/10.1093/OED/2283136643.

[3] K. Ghosh and A. Senapati, "Hate speech detection in low-resourced Indian languages: An analysis of transformer-based monolingual

and multilingual models with cross-lingual experiments," Cambridge University Press, pp. 1–22 (2024).

[4] K. Ghosh, A. Senapati, M. Narzary, and M. Brahma, "Hate Speech Detection in Low-Resource Bodo and Assamese Texts with ML-DL and BERT Models," Scalable Computing: Practice and Experience, vol. 24, pp. 941–955 (2023).

[5] T. Mandl, M. J. R. S. Tjandra, V. L. S. V. D. G., D. A. H. P. Y., and M. S. L. R., "Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages," Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation, pp. 14–17 (2019).

[6] S. Yadav, A. Singh, R. S. K. S., and P. Tiwari, "Leveraging Weakly Annotated Data for Hate Speech Detection in Code-Mixed Hinglish: A Feasibility-Driven Transfer Learning Approach with Large Language Models," arXiv preprint arXiv:2403.02121 (2024).

[7] J. Antonio, P. Garcia, S. C. Soto, and L. S. Gomez, "Leveraging Zero and Few-Shot Learning for Enhanced Model Generality in Hate Speech Detection in Spanish and English," Mathematics, MDPI, vol. 11, p. 5004 (2023).

[8] F. Arco, J. Santos, P. Diaz, and L. Rodriguez, "Respectful or Toxic? Using Zero-Shot Learning with Language Models to Detect Hate Speech," The 7th Workshop on Online Abuse and Harms (WOAH), pp. 60–68 (2023).

[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, and S. M. Mishra, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877-1901 (2020).

[10] S. Bsharat, M. Khalil, S. Mahmood, and J. Anwar, "Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4," arXiv preprint arXiv:2312.16171 (2023).

[11] J. Wei, P. Devlin, and W. Lee, "Finetuned Language Models Are Zero-Shot Learners," arXiv preprint arXiv:2109.01652 (2021).

[12] P. Christiano, G. Dhariwal, and J. K. R. Y. Sharma, "Deep reinforcement learning from human preferences," *Advances in Neural Information Processing Systems*, vol. 30 (2023).

[13] S. Min, S. Shankar, N. Gupta, and M. Singh, "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" Proceedings of EMNLP 2022, pp. 11048–11064 (2022).

[14] J. Wei, C. Ouyang, L. Xu, and T. S. Zhang, "Chain of Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems,* vol. 35, pp. 24824-24837 (2022).

[15] K. Ghosh, M. Chakraborty, M. Choudhury, and P. Dutta, "Annihilate Hates (Task 4, HASOC 2023): Hate Speech Detection in Assamese, Bengali, and Bodo Languages," FIRE (Working Notes), pp. 368-382 (2023).

Received August, 2024