mathematics

MDPI

Article

Investigating the Predominance of Large Language Models in Low-Resource Bangla Language over Transformer Models for Hate Speech Detection: A Comparative Analysis

Fatema Tuj Johora Faria [1], Laith H. Baniata [2,*] and Sangwoo Kang [2,*]

[1] Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka 1208, Bangladesh; fatema.faria142@gmail.com
[2] School of Computing, Gachon University, Seongnam 13120, Republic of Korea
* Correspondence: laith@gachon.ac.kr (L.H.B.); swkang@gachon.ac.kr (S.K.)

Abstract: The rise in abusive language on social media is a significant threat to mental health and social cohesion. For Bengali speakers, the need for effective detection is critical. However, current methods fall short in addressing the massive volume of content. Improved techniques are urgently needed to combat online hate speech in Bengali. Traditional machine learning techniques, while useful, often require large, linguistically diverse datasets to train models effectively. This paper addresses the urgent need for improved hate speech detection methods in Bengali, aiming to fill the existing research gap. Contextual understanding is crucial in differentiating between harmful speech and benign expressions. Large language models (LLMs) have shown state-of-the-art performance in various natural language tasks due to their extensive training on vast amounts of data. We explore the application of LLMs, specifically GPT-3.5 Turbo and Gemini 1.5 Pro, for Bengali hate speech detection using Zero-Shot and Few-Shot Learning approaches. Unlike conventional methods, Zero-Shot Learning identifies hate speech without task-specific training data, making it highly adaptable to new datasets and languages. Few-Shot Learning, on the other hand, requires minimal labeled examples, allowing for efficient model training with limited resources. Our experimental results show that LLMs outperform traditional approaches. In this study, we evaluate GPT-3.5 Turbo and Gemini 1.5 Pro on multiple datasets. To further enhance our study, we consider the distribution of comments in different datasets and the challenge of class imbalance, which can affect model performance. The BD-SHS dataset consists of 35,197 comments in the training set, 7542 in the validation set, and 7542 in the test set. The Bengali Hate Speech Dataset v1.0 and v2.0 include comments distributed across various hate categories: personal hate (629), political hate (1771), religious hate (502), geopolitical hate (1179), and gender abusive hate (316). The Bengali Hate Dataset comprises 7500 non-hate and 7500 hate comments. GPT-3.5 Turbo achieved impressive results with 97.33%, 98.42%, and 98.53% accuracy. In contrast, Gemini 1.5 Pro showed lower performance across all datasets. Specifically, GPT-3.5 Turbo excelled with significantly higher accuracy compared to Gemini 1.5 Pro. These outcomes highlight a 6.28% increase in accuracy compared to traditional methods, which achieved 92.25%. Our research contributes to the growing body of literature on LLM applications in natural language processing, particularly in the context of low-resource languages.

Keywords: hate speech detection; Bengali language; low resource language; large language models; few-shot learning; zero-shot learning; natural language processing

MSC: 68T07

check for updates

Citation: Faria, F.T.J.; Baniata, L.H.; Kang, S. Investigating the Predominance of Large Language Models in Low-Resource Bangla Language over Transformer Models for Hate Speech Detection: A Comparative Analysis. *Mathematics* **2024**, *12*, 3687. https://doi.org/10.3390/math12233687

Academic Editors: Florin Leon, Mircea Hulea and Marius Gavrilescu

Received: 13 October 2024
Revised: 6 November 2024
Accepted: 16 November 2024
Published: 25 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Bengali, spoken by approximately 260 million people, stands as the sixth most spoken language globally. It holds the distinction of being the second most spoken language in India and serves as the national language of Bangladesh. With nearly 205 million native

speakers, Bengali ranks as the seventh most spoken native language worldwide, encompassing about 3.05% of the global population. Online social media platforms have evolved into crucial channels for communication, information sharing, opinion expression, and connection between individuals and businesses. However, they also contend with issues such as the proliferation of hateful or toxic content, bullying, and intimidation. Detecting such content manually on such large platforms is impractical, necessitating automated detection systems. Yet, deploying effective automated detection remains challenging due to the dynamic nature of hate speech. In Bangladesh, approximately 81.7 million people use the internet, with 30 million actively engaging on social media, primarily via mobile phones. Notably, 42 million Bengali-speaking Facebook users participate in commenting, posting, and sharing content, constituting nearly 1.9% of all Facebook users. The use of Bengali on other social media platforms is also seeing a notable increase [1].

Despite extensive research on detecting abusive text in English on social networks, the Bengali language remains significantly underrepresented, even as its online presence grows. While the internet has promoted free speech, it has unfortunately also facilitated an increase in hate speech and vulgar language, particularly targeting Bangladeshi women. Social media has become an essential part of modern life, providing a popular and convenient platform for individuals to communicate and publicly express their thoughts. These platforms, along with online streaming services, have democratized information and amplified freedom of speech, often under the veil of anonymity. However, these same platforms also enable the spread of misinformation and hate speech, presenting significant challenges for regulatory authorities and law enforcement. Addressing hate speech is crucial for protecting human rights and preventing marginalization based on race, gender, ethnicity, or other affiliations. The ease of communication has unfortunately also led to harassment and attacks on individuals based on their expressions of sexism, racism, political opinions, or other concerns. Consequently, incidents of blackmail, cyberterrorism, and online harassment are proliferating rapidly across various social media platforms [2,3].

The vast amount of user-generated content necessitates the application of natural language processing (NLP) and machine learning (ML) models to effectively address the issue of online hate speech. Rapid advancements in artificial intelligence (AI) and machine learning technologies have led to numerous studies achieving promising results in this domain. However, state-of-the-art AI models predominantly rely on supervised learning techniques, which are generally limited to simple binary predictions of hate speech. A critical challenge in AI-based hate speech detection is the highly contextual nature of the problem. Existing supervised learning methods often fail to fully capture this context, resulting in inaccurate predictions. This underscores the need for detection methods capable of understanding and utilizing the full context of hate speech. Despite ongoing research and the development of diverse datasets for training classifiers, most efforts concentrate on English, neglecting low-resource languages like Bengali. The task is further complicated by challenges such as informal language syntax, spelling errors, and non-standard acronyms. Addressing these issues requires robust and context-aware AI models to ensure accurate and comprehensive hate speech detection across different languages and contexts [4,5].

Hate speech is a complex and evolving concept, highly dependent on the prevailing societal norms and the specific context in which it occurs. The deployment of advanced Large Language Models (LLMs) for content moderation is gaining popularity as a method to identify harmful and toxic content online. These models are trained to detect various forms of hate speech, both explicit and implicit. Recent studies have extensively explored the capabilities of models like GPT-3 and GPT-3.5 in hate speech detection. Notably, OpenAI's internal testing has highlighted the potential of GPT-4 as an effective tool for content moderation. Similarly, the latest open-source model, Llama 2, has shown promising results in identifying instances of hate speech [6,7]. These advancements underscore the growing role of LLMs in combating harmful online content, reflecting ongoing efforts to enhance digital safety and community well-being.

A crucial step in content moderation is the filtering of abusive content. A common method for achieving this is training language models on human-annotated content for classification. However, this approach presents several challenges, including the substantial resources required in terms of labor and expertise to annotate hateful content [8].

Additionally, this task exposes annotators to a wide array of hateful content, which is almost always psychologically taxing. Many studies have explored the potential of LLMs in detecting abusive language, but none have examined the role of incorporating additional context as input to or output from such LLMs. This gap highlights the need for further research into how contextual information can enhance the effectiveness of LLMs in moderating abusive content [9–11].

This research addresses the critical need for effective hate speech detection in Bengali, a low-resource language with limited computational tools and datasets. Leveraging Large Language Models (LLMs), this study explores prompt-based strategies aimed at enhancing detection accuracy and tackling unique linguistic challenges within Bengali hate speech. The primary objective is to establish a robust methodology for identifying and mitigating hate speech, contributing to safer and more inclusive digital spaces.

For the first time, we introduce a variety of prompt and input instruction formats to evaluate two prominent LLMs—GPT-3.5 Turbo and Gemini 1.5 Pro—across three datasets: BD-SHS, Bengali Hate Speech Dataset v1.0 and v2.0, and the Bengali Hate Dataset. These datasets are distinct in providing explanations, including rationales and implied statements, that reflect annotator reasoning. Our approach centers on designing prompts that use only the hate speech content as input and query for classification labels, allowing for a streamlined assessment of LLM performance.

We conduct a detailed evaluation using both Zero-Shot and Few-Shot Learning techniques to measure each model's capacity to detect hate speech under varied linguistic scenarios. Additionally, this study systematically examines the impact of different verbalizers on model efficacy, comparing multiple verbalization strategies across the selected models to determine the optimal configurations for enhanced hate speech detection.

This research paper makes several key contributions to the field of hate speech detection. Our key contributions are summarized as follows:

- We propose to employ class-balanced weights in the loss function with pre-trained language models. This approach adjusts the contribution of each class to the overall loss, ensuring that minority classes are given appropriate importance during training.
- We demonstrate that LLMs surpass current benchmarks in hate speech detection accuracy, achieving significant improvements over traditional methods.
- Our Zero-Shot Learning proves effective in identifying hate speech without task-specific training data, demonstrating adaptability to new datasets and languages.
- Our Few-Shot Learning enables efficient model training with minimal labeled examples, allowing for scalable real-time detection in low-resource language contexts.
- Our detailed analysis of prompting strategies reveals insights into optimizing LLMs for hate speech detection tasks, enhancing their applicability in diverse online communities.
- We conduct an error analysis within the framework of pre-trained language models for hate speech detection.
- Our detailed hallucination analysis of Zero-Shot and Few-Shot Learning strategies provide insights into optimizing LLMs for hate speech detection tasks, improving their suitability in a variety of internet groups.

The remainder of this paper is organized as follows: Section 2 offers an extensive review of the related literature, establishing the foundation for our research. Section 3 explores background studies. Section 4 details the datasets used in the study. Section 5 describes the implementation details. Section 6 interprets the results. Section 7 examines the study's limitations, and Section 8 proposes directions for future research.

2. Literature Review

In this section, we provide a concise overview of earlier studies pertinent to our research on hate speech detection in the Bangla language. The overview is structured into four main categories based on the methodologies employed: traditional approaches, deep learning approaches, transformer-based approaches, and Large Language Model-based approaches. Below are summaries of the studies categorized in Tables 1 and 2.

2.1. Traditional Based Approaches

This paper [12] addresses the pressing issue of hate speech and anti-social behavior on social media in Bangladesh, focusing on Bengali language comments. The authors collected 2000 comments from Facebook and YouTube and developed a dataset for analysis. Utilizing a Gated Recurrent Unit (GRU) neural network and various machine learning classifiers like Logistic Regression, Random Forest, Multinomial Naive Bayes (MNB), and Support Vector Machine (SVM), the study aimed to distinguish between social and anti-social comments. The GRU model achieved 78.89% accuracy, while MNB attained 80.51% accuracy. This research highlights the scarcity of Bengali language datasets and emphasizes the importance of context-specific feature extraction. The study's findings contribute significantly to the field by providing a benchmark dataset and demonstrating the effectiveness of GRU and MNB models in detecting anti-social comments. However, this paper does not explore the use of LLMs, and it has limitations in terms of scalability and the ability to handle nuanced context in comments. Similarly, another paper [13] highlights the increasing significance of social media in everyday life and the rising problem of negative comments on these platforms. While substantial work has been conducted on abusive text detection in other languages, similar research in Bengali is limited. This study utilizes a dataset comprising 5000 comments collected from various social media platforms, such as Facebook and YouTube, with 2698 labeled as abusive and 2196 as non-abusive. Six machine learning algorithms—Logistic Regression, Multinomial Naive Bayes, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Gradient Boosting—are employed, finding that SVM achieves the highest accuracy at 85.7%. The research emphasizes the challenges of using multiclass classification for Bangla and opts for binary classification to categorize comments as abusive or not. Additionally, the study underscores the importance of data pre-processing and feature extraction using TFIDF Transformer and Vectorizer. However, this paper also does not explore the use of LLMs, limiting its potential to capture complex linguistic patterns and context. Furthermore, another paper [14] discusses the rapid growth of the internet and social media, highlighting how these platforms have enabled free expression but also facilitated the spread of hate speech targeting individuals based on ethnicity, religion, gender, and other characteristics. This increase in online hate speech has led to disputes and cyberbullying, prompting organizations to seek effective solutions. The study focuses on detecting hate speech in Bangla videos using machine learning classification methods. Due to the lack of available datasets, the authors create a dataset from scratch, involving collecting, transcribing, and pre-processing videos from YouTube. They experiment with various machine learning and deep learning models, finding that Logistic Regression and the GRU model demonstrate the best accuracy. The GRU model achieves an impressive 98.89% accuracy, while the Logistic Regression model also shows high precision, recall, and F1-Scores. Despite these successes, this paper does not explore the use of LLMs, which could have provided a more nuanced understanding of the transcribed video content.

Table 1. Summary of studies on hate speech detection using Traditional Approaches and Deep Learning Methods.

Types	Authors	Year	Models Employed	Performance Metrics	Key Findings
Traditional Approaches	Manash et al. [12]	2022	Gated Recurrent Unit (GRU), Logistic Regression, Random Forest, Multinomial Naive Bayes (MNB), Support Vector Machine (SVM)	GRU: 78.89% accuracy, MNB: 80.51% accuracy	Developed a dataset of 2000 Bengali comments; highlighted scarcity of Bengali datasets and importance of context-specific feature extraction; MNB and GRU models effective in detecting anti-social comments.
	Sherin et al. [13]	2022	Logistic Regression, Multinomial Naive Bayes, Random Forest, Support Vector Machine (SVM), Gradient Boosting	SVM: 85.7% accuracy	Dataset of 5000 comments; emphasized challenges in multiclass classification for Bangla; binary classification was used; highlighted importance of data pre-processing and TFIDF feature extraction.
	Istiaq et al. [14]	2021	Logistic Regression, Gated Recurrent Unit (GRU)	GRU: 98.89% accuracy	Created dataset from scratch with videos from YouTube; high accuracy with GRU model; Logistic Regression also showed high precision, recall, and F1-Scores; focused on detecting hate speech in Bangla videos.
Deep Learning Approaches	Rezaul et al. [15]	2020	Multichannel Convolutional LSTM (MConv-LSTM), incorporating BengFastText	MConv-LSTM: F1-Scores of 90.45%	Developed BengFastText, the largest Bengali word embedding model based on 250 million articles; created three extensive datasets; MC-LSTM with BengFastText outperformed baseline models.
	Nauros et al. [4]	2022	Bi-LSTM, Support Vector Machine (SVM)	Bi-LSTM: F1-Score of 91.0%	Introduced BD-SHS, a large manually labeled dataset with over 50,200 offensive comments; Bi-LSTM trained with informal embeddings achieved highest F1-Score; outperformed other pre-trained embeddings like BengFastText and MFT.
	Amit et al. [1]	2022	LSTM, GRU, Attention-based decoders	Attention-based model: 77% accuracy	Proposed an encoder–decoder-based model for classifying Bengali Facebook comments; collected 7425 comments across seven hate speech categories; attention-based model achieved highest accuracy; included Bangla Emot Module
	Alvi et al. [16]	2019	GRU, Random Forest	GRU: 70.10% accuracy	Compiled and annotated a dataset of 5126 comments into six classes; Random Forest achieved 52.20% accuracy, GRU model improved to 70.10%; emphasized importance of linguistic and quantitative feature extraction for Bengali.

Table 2. Summary of studies on hate speech detection using Transformer-Based Approaches and Large Language Models.

Types	Authors	Year	Models Employed	Performance Metrics	Key Findings
Transformer-Based Approaches	Jobair et al. [3]	2023	BERT, SVM, LSTM, BiLSTM	BERT: 80% accuracy on new dataset, 97% accuracy on existing dataset	Compiled a dataset of 8600 comments; BERT showed highest accuracy at 80% on new dataset and 97% on existing dataset of 30,000 records; BERT outperformed SVM, LSTM, and BiLSTM.
	Mithun et al. [5]	2022	m-BERT, XLM-RoBERTa, IndicBERT, MuRIL	m-BERT: F1-Score of 0.81	Developed an annotated dataset of 10K Bengali posts (5K actual, 5K Romanized); XLM-RoBERTa performed best in separate training; MuRIL outperformed in joint and few-shot training scenarios.
	Rezaul et al. [17]	2020	BERT variants (including XLM-RoBERTa), traditional ML models, DNN models (CNN, Bi-LSTM)	XLM-RoBERTa: F1-Score of 87%, MCC score of 0.82	Evaluated BERT variants; XLM-RoBERTa achieved highest F1-Score of 87%; ensemble approach improved overall accuracy by 1.8%; highlighted challenges in detecting political hate speech; traditional ML models showed varied performance due to feature selection.
Large Language Models	Keyan et al. [7]	2024	GPT-3.5-turbo, Chain-of-Thought prompts	Accuracy: 0.85, Precision: 0.8, Recall: 0.95, F1-Score: 0.87	Chain-of-Thought reasoning prompts significantly outperform other strategies, capturing intricate contextual details for accurate hate speech detection.
	Sarthak et al. [8]	2023	Flan-T5-large, text-davinci-003, GPT-3.5-turbo-0301	F1-Scores: Flan-T5-large: 0.59 (HateXplain), 0.63 (implicit hate), text-davinci-003: 0.45 (HateXplain), 0.36 (implicit hate)	Flan-T5-large outperforms other models with vanilla prompts. Incorporating target community information into prompts yields a 20–30% performance boost. Precise prompt engineering is critical for optimizing LLMs in hate speech detection.
	Flor et al. [9]	2023	mT0, FLAN-T5, multilingual XLM-RoBERTa	Macro-F1-Scores: FLAN-T5: 65.34 (English), 62.61 (Spanish), 57.29 (Italian)	Zero-Shot Learning with prompting can match or surpass fine-tuned models' performance, particularly with instruction fine-tuned models. Prompt and model selection significantly impact accuracy.

2.2. Deep Learning-Based Approaches

This research paper [15] addresses the exponential growth of social media, which, while empowering free expression, also facilitates online harassment and hate speech. It highlights the lack of computational resources for under-resourced languages like Bengali. In response, it develops BengFastText, the largest Bengali word embedding model, based on 250 million articles. It creates three extensive datasets for hate speech detection, document classification, and sentiment analysis. Experiments with a Multichannel Convolutional LSTM (MC-LSTM) network, incorporating BengFastText, demonstrate superior performance compared to baseline models, achieving high F1-Scores. Similarly, another research paper [4] introduces the impact of social media platforms and online streaming services on the proliferation of hate speech. To address the need for linguistically diverse datasets, it introduces BD-SHS, a large, manually labeled dataset including hate speech in various social contexts. BD-SHS contains over 50,200 offensive comments, making it significantly larger than previous Bangla hate speech datasets. The dataset is annotated using a hierarchical process that includes hate speech identification, target identification, and the categorization of hate speech types. For benchmarking, various models, including SVM and Bi-LSTM architectures, are experimented with. The Bi-LSTM model, trained with informal embeddings derived from 1.47 million social media comments, achieves the highest F1-Score of 91.0% in hate speech identification. This model consistently outperforms other pre-trained embeddings like BengFastText and multilingual fastText, which are trained on formal texts. Another study [1] highlights the growing prevalence of hate speech on social media, particularly in Bengali, despite extensive research in other languages. It addresses this gap by proposing an encoder–decoder-based machine learning model to classify Bengali comments on Facebook. It collects a dataset of 7425 comments across seven hate speech categories, using a combination of automated and manual methods due to limitations with the Facebook Graph API. The pre-processing involves tokenization, stemming, stopword removal, and extracting features using TF-IDF and word embedding. Three models—LSTM, GRU, and attention-based decoders—are evaluated, with the attention-based model achieving the highest accuracy at 77%. The study also incorporates a Bangla Emot Module to detect emotions from emojis and emoticons, enhancing the model's interpretability. Furthermore, another research paper [16] investigates the detection and classification of hateful speech in Bengali language social media, focusing on Facebook comments. The study employs both traditional machine learning algorithms and a GRU-based deep neural network model. It compiles and annotates a dataset of 5126 comments, categorizing them into six classes: hate speech, communal attack, inciteful, religious hatred, political comments, and religious comments. This dataset marks the first significant contribution to Bengali language hate speech detection in social media. Comparing various machine learning algorithms, it achieves 52.20% accuracy with Random Forest, which improved to 70.10% using the GRU model. The study highlights the importance of linguistic and quantitative feature extraction tailored to the Bengali social context and underscores the superior performance of the GRU model in understanding context and semantics, which is critical for accurate hate speech detection in Bengali. Collectively, these studies emphasize the need for more advanced computational models and resources tailored to the Bengali language, highlighting the absence of LLMs in addressing hate speech detection comprehensively.

2.3. Transformer Based Approaches

This research paper [3] emphasizes the critical need to address the proliferation of hate speech on social media in Bangladesh, particularly due to the lack of comprehensive Bangla datasets. It compiles a new dataset of 8600 user comments from Facebook and YouTube, categorized into sports, religion, politics, entertainment, and others. Various models are tested for hate speech detection, with BERT showing the highest accuracy at 80% on their dataset. When applied to an existing dataset of 30,000 records, BERT achieves an impressive accuracy of 97%, surpassing the performance of previously tested models like SVM, LSTM, and BiLSTM. However, there is a noticeable absence of Large

Language Models specifically tailored for Bangla hate speech detection. Furthermore, another research paper [5] highlights that social media is a hotspot for hateful and offensive content, impacting race, gender, and religion in an unprejudiced society. Despite significant research on hate speech detection in English, there is a notable gap in low-resource languages like Bengali, including its Romanized form used in social media interactions. To address this, it develops an annotated dataset of 10K Bengali posts (5K actual and 5K Romanized) and implements several baseline models for classification. Experiments using m-BERT, XLM-Roberta, and IndicBERT are conducted, exploring interlingual transfer mechanisms. XLM-Roberta performs best when training datasets separately, while MuRIL outperforms in joint and few-shot training scenarios by better interpreting semantic expressions. Nonetheless, the absence of advanced Large Language Models for Bangla hate speech detection remains a significant gap. Moreover, this research paper [17] highlights the rapid expansion of social media and micro-blogging platforms, which have empowered freedom of expression but also facilitated the spread of antisocial behaviors such as online harassment, cyberbullying, and hate speech. The study evaluates four variants of the BERT model on a test set, reporting XML-RoBERTa as the top-performing model with an F1-Score of 87%, outperforming other transformer models by 2% to 5%. Using WeightWatcher for ensemble prediction, they achieve the highest MCC score of 0.82, indicating a strong correlation between predictions and ground truths. Their ensemble approach improve overall accuracy by 1.8% across classes, effectively addressing misclassification rates. The effectiveness of BERT variants over traditional ML and DNN baselines is emphasized, particularly in minimizing classification errors across imbalanced datasets. The study provides class-specific classification reports, highlighting nuances in detecting personal versus political hate speech, with specific challenges noted in identifying political hate due to overlapping terms with personal hate expressions. It also explores feature selection's impact on ML baselines, noting significant improvements in models like GBT, which outperforms others with an MCC score of 0.571. However, SVM, LR, and NB models show degraded performance due to feature selection's impact on their assumptions of feature independence. In contrast, CNN and Bi-LSTM among DNN baselines perform reasonably well, although they fall short compared to transformer-based models like XML-RoBERTa. Despite these advances, the lack of Large Language Models specifically designed for Bangla hate speech detection persists as a major limitation.

2.4. Large Language Model Based Approaches

This research paper [7] addresses the critical issue of online hate speech detection, highlighting its contextual nature and the limitations of existing methods. It underscores the potential of LLMs for context-aware detection due to their extensive training on diverse datasets. However, it notes the lack of effective prompting strategies for utilizing LLMs in this domain. Conducting a large-scale study using five established hate speech datasets, the researchers discover that LLMs, especially with carefully crafted prompts, often surpass traditional models. They propose four diverse prompting strategies, with the Chain-of-Thought reasoning prompt significantly outperforming others by capturing intricate contextual details. The Chain-of-Thought prompt achieves an accuracy of 0.85, precision of 0.8, recall of 0.95, and an F1-Score of 0.87. This study emphasizes the importance of prompt engineering in optimizing LLMs for accurate hate speech detection, employing models such as GPT-3.5-turbo in their experiments. However, they only explore the English language for hate speech detection, leaving a gap for Bangla language applications. Furthermore, another research paper [8] provides a comprehensive analysis of various prompting strategies applied to LLMs for online hate speech detection. It demonstrates that Flan-T5-large outperforms other models with vanilla prompts, while text-davinci-003 shows superior results over GPT-3.5-turbo-0301. Incorporating target community information into prompts yields a 20–30% performance boost. Additionally, explanations and definitions as prompts enhance accuracy, though combining multiple strategies does not consistently yield further improvements. Detailed error analysis reveals frequent

misclassifications, particularly for non-hate/non-toxic categories, underscoring the need for precise prompt engineering to optimize LLMs in hate speech detection. For the HateXplain and implicit hate datasets, Flan-T5-large achieves F1-Scores of 0.59 and 0.63, respectively, outperforming gpt-3.5-turbo-0301 and text-davinci-003. However, this study also focuses solely on the English language, indicating a gap in the research for Bangla hate speech detection. Moreover, another research paper [9] identifies two key challenges in hate speech detection: the limited availability of labeled data and the high variability of hate speech across contexts and languages. Prompting offers a solution by enabling models to incorporate task-specific knowledge without labeled data. The study explores Zero-Shot Learning (ZSL) with prompting for hate speech detection in three languages using eight benchmark datasets. The findings reveal that prompt selection significantly impacts results, with prompting—especially with recent Large Language Models—often matching or surpassing fine-tuned models. This highlights the potential of prompting for under-resourced languages, showing that both the prompt and model choices are crucial for accurate hate speech detection. Nonetheless, this research too only explores the English language, leaving a gap for Bangla language applications.

3. Background Study

3.1. Transformer-Based Models

3.1.1. BERT-Based Transformer Models

Bidirectional Encoder Representations from transformers, commonly known as BERT [18], is built upon a sophisticated deep learning framework that establishes connections between input and output elements, adaptively determining their relationships. The distinctive feature of BERT is its bidirectional training capability, which allows the model to understand the context of a word by considering both its preceding and succeeding words. This bidirectional approach contrasts with traditional models that only consider one direction, either forward or backward. The BERT architecture is designed to enable the model to capture intricate linguistic patterns and contextual dependencies in text, making it exceptionally powerful for a variety of natural language processing tasks. BanglaBERT Base [19], a variant tailored for the Bengali language, follows the same robust architecture as the original BERT model. This alignment ensures that the BanglaBERT Base inherits the powerful contextual understanding capabilities of BERT, enabling it to effectively process and analyze Bengali text with high accuracy. mBERT [18], or Multilingual BERT, extends the BERT architecture to support multiple languages. It is pre-trained on a large corpus of text from 104 different languages, including Bengali, enabling it to understand and generate text across various languages. mBERT leverages the same bidirectional training approach as BERT, allowing it to capture the context of words from diverse linguistic backgrounds. This makes mBERT a versatile tool for multilingual natural language processing tasks. XLM-RoBERTa [20], or Cross-lingual Language Model-RoBERTa, is an extension of the RoBERTa model designed to handle multiple languages. It is pre-trained on a vast dataset that includes text from 100 languages, which helps it learn cross-lingual representations. XLM-RoBERTa uses the same bidirectional training approach as BERT, allowing it to capture the context of words in a variety of languages. This makes it a powerful tool for multilingual tasks, such as cross-lingual understanding and translation.

3.1.2. ELECTRA-Based Transformer Models

Efficiently Learning an Encoder that Classifies Token Replacements Accurately, known as ELECTRA [21], utilizes a unique pre-training task focused on identifying replaced tokens within an input sequence. This method involves two main components: a discriminator model and a generator model. The discriminator model is trained to recognize which tokens have been replaced in a corrupted sequence, while the generator model is simultaneously trained to predict the original tokens for the masked out ones. This setup is somewhat reminiscent of a generative adversarial network (GAN) training system, but without the adversarial component. In ELECTRA, the generator is not trained to deceive the

discriminator but rather to provide accurate token predictions. This collaborative process enhances the model's ability to understand and generate text. BanglaBERT [22] serves as the ELECTRA discriminator model, specifically designed for the Bengali language. This model effectively leverages the ELECTRA architecture to accurately classify and process Bengali text, improving the overall performance of text classification tasks in the Bengali language.

3.1.3. ALBERT-Based Transformer Models

A Lite BERT, known as ALBERT [23], has shown that exceptional language models do not always require larger architectures. ALBERT achieves efficiency and high performance by utilizing the same encoder segment architecture as the original transformer but introduces three crucial modifications: factorized embedding parameters, cross-layer parameter sharing, and employing Sentence Order Prediction (SOP) instead of Next Sentence Prediction (NSP). These modifications allow ALBERT to maintain a smaller model size while still delivering superior performance. Factorized embedding parameters reduce the number of parameters, cross-layer parameter sharing ensures consistency and reduces redundancy across layers, and SOP provides a more challenging pre-training objective than NSP, leading to better model understanding and contextual awareness. In the context of the Bengali language, sahajBERT (https://huggingface.co/neuropark/sahajBERT, accessed on 15 November 2024) is a collaborative pre-trained ALBERT model that leverages these innovations. It utilizes masked language modeling (MLM) and Sentence Order Prediction (SOP) objectives to effectively learn from and understand Bengali text. This approach enables sahajBERT to perform various natural language processing tasks with high accuracy and efficiency, making it a powerful tool for Bengali language applications (see Table 3).

Table 3. Architectural and training objective details of various pre-trained language models.

Architecture	Model	Layers	Attention Heads	Parameters	Objective Type During Training	Embedding Size
ELECTRA	BanglaBERT	12	12	110 M	MLM with Replaced Token Detection (RTD)	768
BERT	BanglaBERT Base	12	12	110 M	Masked Language Model (MLM)	768
	mBERT	12	12	110 M	Multilingual Masked Language Model (MLM)	768
	XLM-RoBERTa	24	16	125 M	Masked Language Model (MLM)	768
ALBERT	sahajBERT	24	16	18 M	Multilingual Masked Language Model (MLM)	128

3.2. Large Language Models

3.2.1. GPT 3.5 Turbo

Generative Pre-trained Transformer 3.5 [24], or GPT-3.5, developed by OpenAI, San Francisco, CA, USA, is a sophisticated language model renowned for its ability to understand and generate human-like text. Built on the transformer architecture, it leverages self-attention mechanisms to process and generate text efficiently. Key features of GPT-3.5 include its remarkable capability for Few-Shot Learning, where it can perform tasks with minimal task-specific data. It also excels in Zero-Shot Learning, requiring no examples, and one-shot learning, needing just a single example to generate responses. GPT-3.5 Turbo represents a significant advancement within the GPT series, particularly in comprehending and executing instructions accurately. This makes it ideal for tasks requiring specific formatting or outputs, such as creative content development. Developers can fine-tune the model to tailor its behavior to specific needs, enhancing performance across various applications. For instance, it can be adjusted to maintain consistent language use or to simplify prompts for desired responses. With an extensive context window capable of handling up to 16,385 tokens and improved precision in formatting, GPT-3.5 Turbo effectively addresses encoding challenges for non-English languages. It also offers rapid response times, restricted to generating outputs up to 4096 tokens in length, making it a versatile and cost-effective solution for demanding text generation tasks.

3.2.2. Gemini 1.5 Pro

Gemini 1.5 [25] represents a significant leap forward, incorporating innovations in research and engineering across nearly every aspect of foundational model development and infrastructure. One of its standout features is the Mixture-of-Experts (MoE) architecture, which enhances efficiency in training and deployment. Unlike traditional Transformers that function as a single large neural network, MoE models are divided into smaller "expert" neural networks. These experts selectively activate based on the input type, greatly enhancing the model's efficiency. The first model in the Gemini 1.5 series is Gemini 1.5 Pro. This mid-size multimodal model is optimized for a wide range of tasks and performs at a level comparable to Gemini 1.0 Ultra, the largest model to date. A key innovation in Gemini 1.5 Pro is its breakthrough capability in long-context understanding. With a standard context window of 128,000 tokens, Gemini 1.5 Pro is already impressive. Additionally, a select group of developers and enterprise customers can experiment with a context window of up to 1 million tokens through AI Studio and Vertex AI in a private preview. This extended context window allows the model to process vast amounts of information in a single prompt, enhancing the consistency, relevance, and usefulness of its outputs. Gemini 1.5 Pro's context window capacity significantly exceeds the original 32,000 tokens of Gemini 1.0. Now capable of handling up to 1 million tokens in production, Gemini 1.5 Pro can process extensive information in a single go, making it a powerful tool for large-scale data processing and analysis.

3.2.3. Comparison of GPT-3.5 Turbo and Gemini 1.5 Pro in Hate Speech Detection

The comparative analysis of GPT-3.5 Turbo and Gemini 1.5 Pro in hate speech detection reveals a complex landscape of artificial intelligence capabilities in content moderation. While GPT-3.5 Turbo demonstrates robust performance in identifying explicit hate speech and common discriminatory patterns, likely due to its extensive training on diverse internet content, Gemini 1.5 Pro shows enhanced capability in detecting more subtle forms of bias and contextual nuances, particularly in cases where cultural or situational context is crucial for accurate classification. Both models exhibit varying false positive rates, with GPT-3.5 Turbo occasionally flagging legitimate social commentary as hate speech, while Gemini 1.5 Pro tends to be more conservative in its classifications but potentially misses some borderline cases. The key differentiator appears to be Gemini 1.5 Pro's improved ability to understand conversational context and implicit bias, though this comes at the cost of increased computational requirements and slower processing times compared to GPT-3.5 Turbo's more streamlined approach. In terms of technical implementation, GPT-3.5 Turbo employs a more traditional transformer-based architecture optimized for efficiency and quick response times, making it particularly suitable for real-time content moderation in high-traffic environments. Its training methodology appears to prioritize precision in detecting widely recognized forms of hate speech, though this sometimes results in over-sensitivity to certain keywords and phrases without fully grasping the broader context. In contrast, Gemini 1.5 Pro's architecture incorporates more sophisticated contextual analysis mechanisms, allowing it to better understand the intent behind potentially harmful content and distinguish between genuine hate speech and legitimate discussion of sensitive topics. The performance metrics reveal interesting patterns in both models' capabilities. Practical implications for deployment suggest that the choice between these models should be guided by specific use case requirements. For high-volume social media platforms requiring real-time moderation, GPT-3.5 Turbo's efficiency and higher recall rate might be preferable, despite the higher false positive rate. For applications where accuracy and nuanced understanding are paramount, such as academic research or legal content analysis, Gemini 1.5 Pro's superior contextual understanding and lower false positive rate make it the more suitable choice. The ideal solution might involve a hybrid approach, using GPT-3.5 Turbo for initial screening and Gemini 1.5 Pro for more detailed analysis of flagged content. Looking toward future developments, both models' limitations highlight areas for improvement in hate speech detection systems. These include the need for better

understanding of evolving language patterns, improved detection of coded language and dog whistles, and more sophisticated analysis of multimodal content, where hate speech might be conveyed through a combination of text, images, and context. Additionally, there is a growing need for these systems to better handle intersectional aspects of hate speech, where multiple forms of discrimination might intersect in complex ways. Resource considerations also play a crucial role in model selection. GPT-3.5 Turbo's more efficient architecture translates to lower computational costs and energy consumption, making it more accessible for smaller organizations and applications. Gemini 1.5 Pro's superior contextual understanding comes with higher computational requirements and associated costs, potentially limiting its deployment in resource-constrained environments. This trade-off between performance and resource utilization remains a key consideration for organizations implementing hate speech detection systems.

3.3. Evaluation Metrics

3.3.1. Accuracy

Accuracy [26] measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. It is the most intuitive performance measure but can be misleading if the dataset is imbalanced, such as when hate speech instances are much fewer compared to non-hate speech instances. The formula for accuracy is

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- TP = True Positives (correctly identified hate speech)
- TN = True Negatives (correctly identified non-hate speech)
- FP = False Positives (non-hate speech incorrectly identified as hate speech)
- FN = False Negatives (hate speech incorrectly identified as non-hate speech)

3.3.2. Precision

Precision [27], also known as Positive Predictive Value, measures the proportion of positive identifications (hate speech) that are actually correct. This metric is crucial in contexts where the cost of false positives is high, such as flagging non-hate speech content as hate speech. The formula for precision is

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

3.3.3. Recall

Recall [27], also known as sensitivity or True Positive Rate, measures the proportion of actual positives (hate speech instances) that were correctly identified. This metric is crucial when the cost of false negatives is high, such as failing to identify actual hate speech. The formula for recall is

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

3.3.4. F1-Score

The F1-Score [26] is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, especially useful when you need a balance between the two. The formula for the F1-Score is

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

4. Dataset Description

4.1. Dataset 1: BD-SHS

The BD-SHS [4] dataset is a comprehensive collection of Bangla hate speech (HS) comments sourced from various social media platforms. It consists of 50,281 comments meticulously annotated using a three-level hierarchical scheme by three annotators per comment. The majority decision among annotators is adopted, resulting in a Fleiss Kappa score of 0.658, indicating moderate inter-annotator agreement. From this dataset, we conduct Level 1 hate speech (HS) identification, classifying comments as either hate speech (HS) or non-hate speech (NH) based on the criteria outlined in Table 4. Figure 1 illustrating the overall distribution of comments and Figure 2 showcasing detailed examples of hate speech detection categories.

Table 4. Detailed descriptions of hate speech (HS) and non-hate speech (NH) comment categories.

Category Name	Description
HS Comments	• Attacks based on ethnicity, nationality, religion, sexual orientation, age, gender, disability, or disease. • Implicit support for hate speech acts without direct dehumanization. • Expressions advocating or supporting violence against individuals or communities. • Dehumanization through comparisons with animals, criminals, or historically vilified figures. • Expressions of disgust towards individuals or groups.
NH Comments	• Lack of dehumanization or attacks based on the specified criteria. • Use of swear words not directed towards humans is not considered HS.

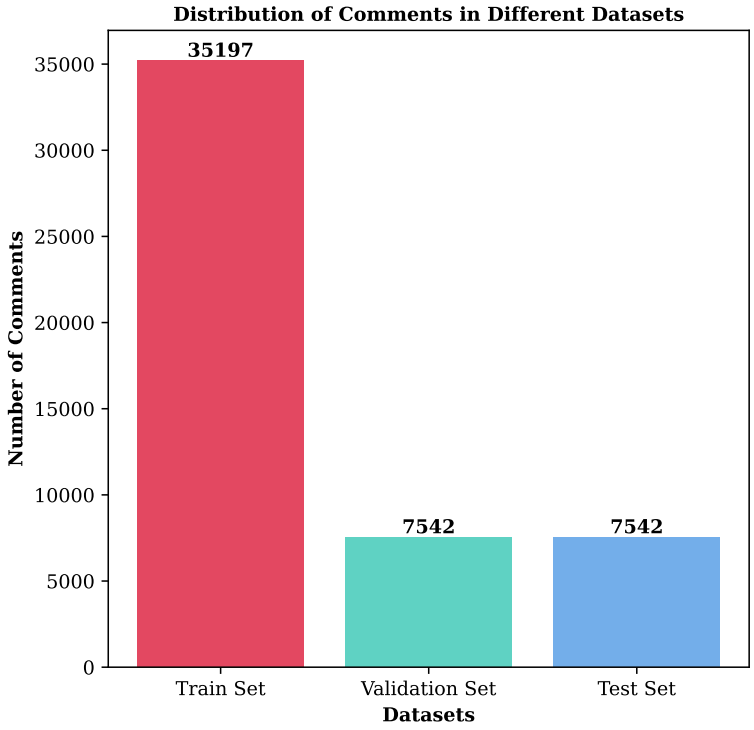


Figure 1. Visual representation of comment distribution across datasets in Dataset 1.

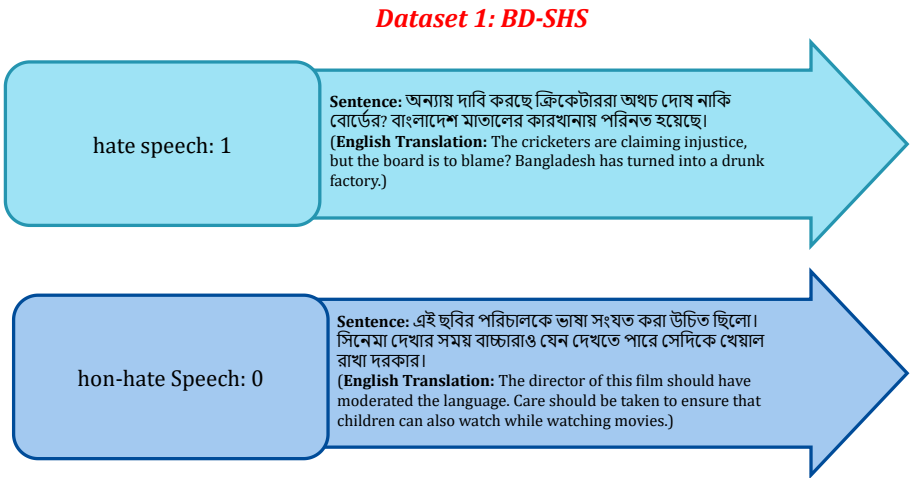


Figure 2. Detailed visual examples of hate speech detection categories in Dataset 1.

4.2. Dataset 2: Bengali Hate Speech Dataset V1.0 and V2.0

The dataset [15,17] is an extensive compilation of Bengali articles from a variety of sources, including social media (Twitter, Facebook, and LinkedIn), books, TV channels, news items from major newspapers, blogs, and sports websites. Using a bootstrapping approach, two linguists and three native Bengali speakers annotated hate speech from this dataset. At first, particular texts with common insults and signs of hate speech were found. The inclusion of 175 normalized abusive phrases that are frequently employed in hate speech in Bengali was the basis for a semi-automatic annotation process applied to a set of 10,000 statements, texts, or articles. If an annotation contained one or more of these specified terms, it was labeled as "hate". Notwithstanding difficulties in separating hate speech from unpleasant language and regional differences in hate speech categories, the procedure was centered on objective standards. Political, personal, gender-based abuse, geopolitical, and religious hate are among the categories of hate speech that have been recognized. The dataset contains 3418 remarks (or roughly 3.5% of the annotated texts) that have been classified as hate speech. Three experts verified and edited the annotations, assuring their robustness and minimizing bias. An additional 3000 labeled samples that fell into the categories of political, personal, geopolitical, religious, and gender abusive hate were added to the dataset. Semantic overlap made it difficult to differentiate between hate that is directed towards a person and hate that is directed towards a gender. Personal hate was defined as remarks that were antagonistic to specific people and primarily addressed to women in Bengali language. A bootstrap technique was used to collect data, with an emphasis on texts that contained particular sorts of slurs and phrases that were aimed towards individuals or groups. Texts were collected from newspapers, YouTube comments, and Facebook. The annotation procedure involved three annotators: an NLP researcher, a linguist, and a native Bengali speaker. Annotators were given objective materials to work with, and labels were decided upon by majority vote in order to reduce bias. In order to guarantee annotation quality and impartial criteria for decision-making, inter-annotator agreement was assessed using Cohen's Kappa statistic. Figure 3 illustrates the overall distribution of comments, and Figure 4 showcases detailed examples of the hate speech detection categories.

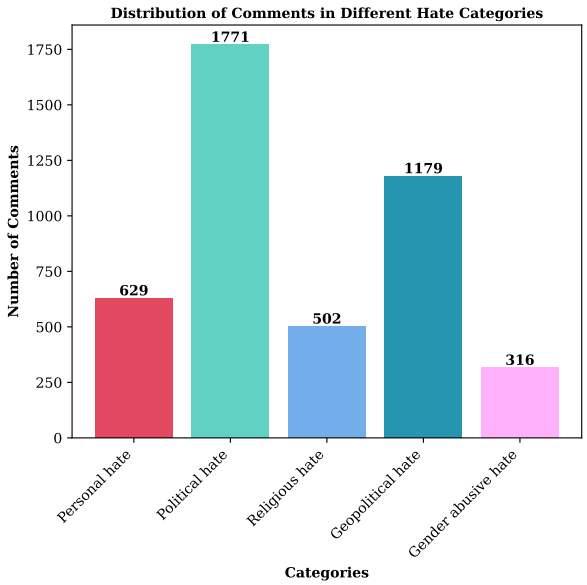


Figure 3. Visual representation of comment distribution across datasets in Dataset 2.

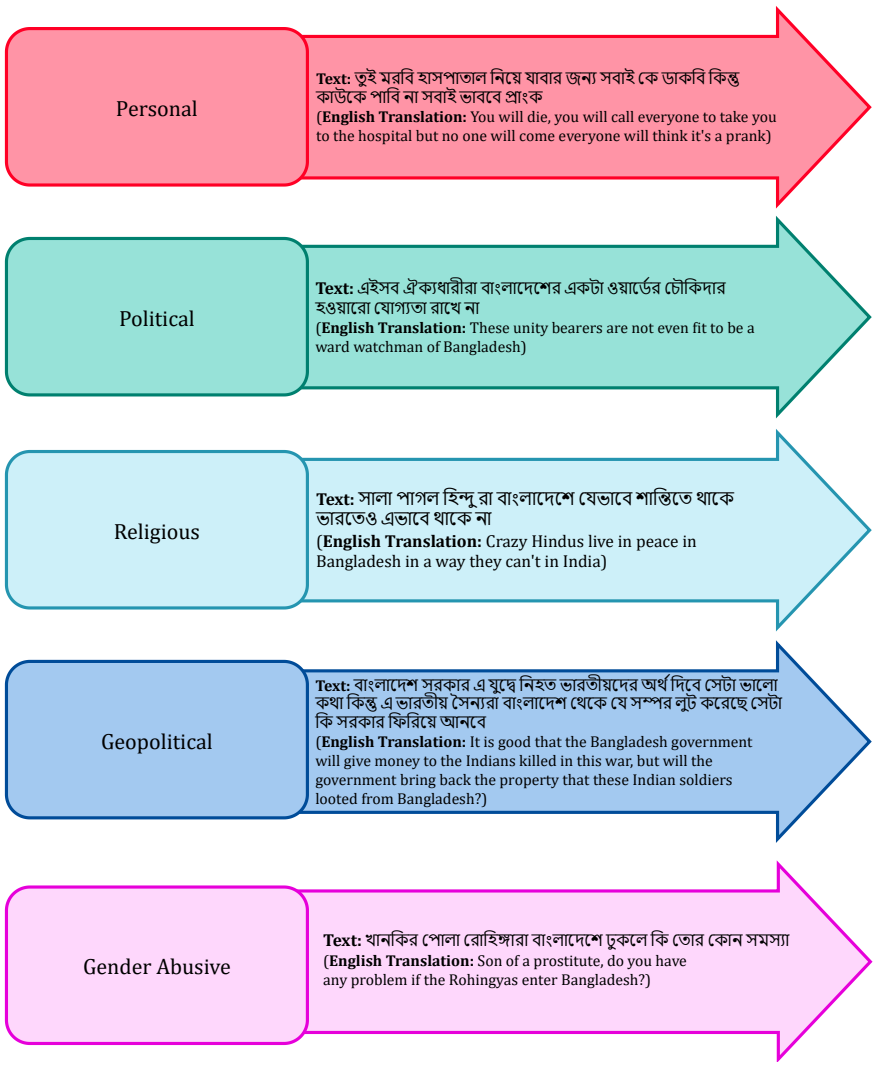


Figure 4. Detailed visual examples of hate speech detection categories in Dataset 2.

4.3. Dataset 3: Bengali Hate Dataset

The dataset [2] comprises 15,000 Bengali posts collected from social media platforms such as Facebook and YouTube between January 2021 and April 2022. Initially, 110,000 posts were gathered, from which 8500 posts containing Bengali profane words were filtered to focus on offensive content. To ensure class balance, an additional 8500 non-offensive posts were selected from the original dataset. After manual labeling and pre-processing to remove noise such as unidentified characters, symbols, and emojis, 15,000 posts were retained for analysis. The dataset annotation process involved 27 independent native Bengali labelers who meticulously categorized each sentence as either hate speech ('1') or non-hate ('0'), adhering to the predefined guidelines. Figure 5 illustrates the overall distribution of comments, and Figure 6 showcases detailed examples of hate speech detection categories.

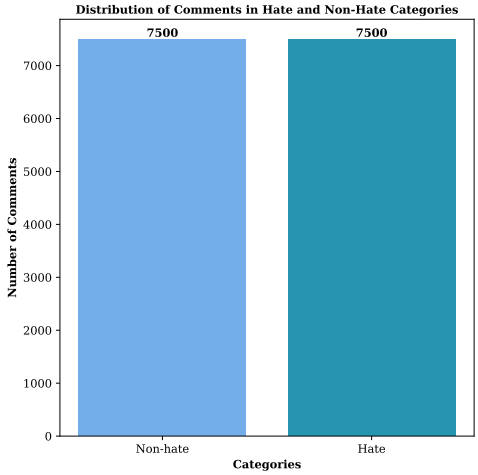


Figure 5. Visual representation of comment distribution across datasets in Dataset 3.

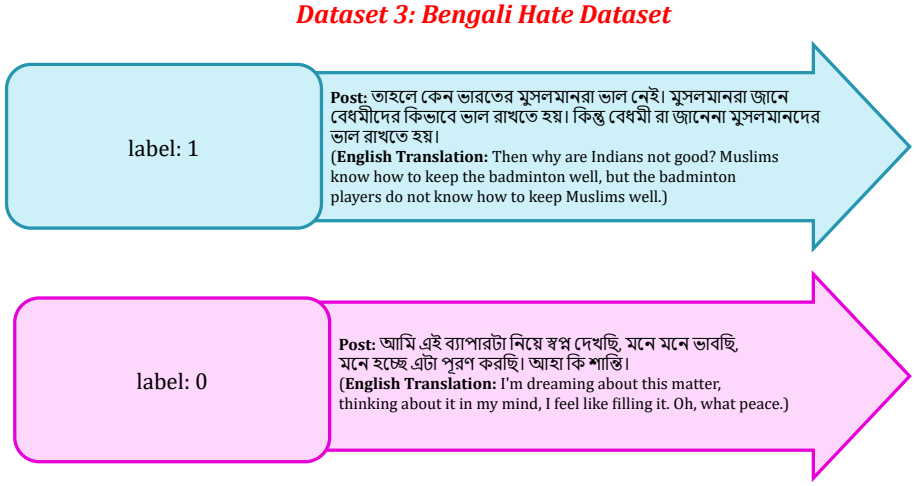


Figure 6. Detailed visual examples of hate speech detection categories in Dataset 3.

5. Implementation Details

In this section, we detail our implementation of Bangla hate speech detection using advanced natural language processing techniques. Our approach harnesses the power of both pre-trained language models (PLMs) and Large Language Models (LLMs) to achieve the precise classification of hate speech and non-hate speech content in Bengali text. We begin by leveraging pre-trained language models (PLMs) to handle Bangla hate speech detection. This experiment involves tailored text pre-processing steps specific to Bengali, addressing class imbalance through effective strategies, fine-tuning procedures to adapt PLMs to the hate speech detection task, hyperparameter optimization for optimal

performance, and evaluation using appropriate metrics. We also apply rigorous error analysis techniques and post-processing methods to refine prediction accuracy. In our second experiment, we explore the effectiveness of Large Language Models (LLMs) in Bangla hate speech detection. This involves adapting LLMs to the Bengali language, incorporating sophisticated pre-processing steps, handling class imbalance challenges, fine-tuning LLMs on hate speech detection datasets, optimizing hyperparameters, and evaluating performance using standard metrics. Additionally, we investigate the potential of zero-shot and Few-Shot Learning paradigms to further enhance model robustness across diverse datasets. Figure 7 depicts the workflow of Bangla hate speech detection using PLMs. The code and supporting files for this study are publicly available for reference and use on GitHub at https://github.com/fatemafaria142/Bangla-Hate-Speech-Detection (accessed on 1 September 2024).

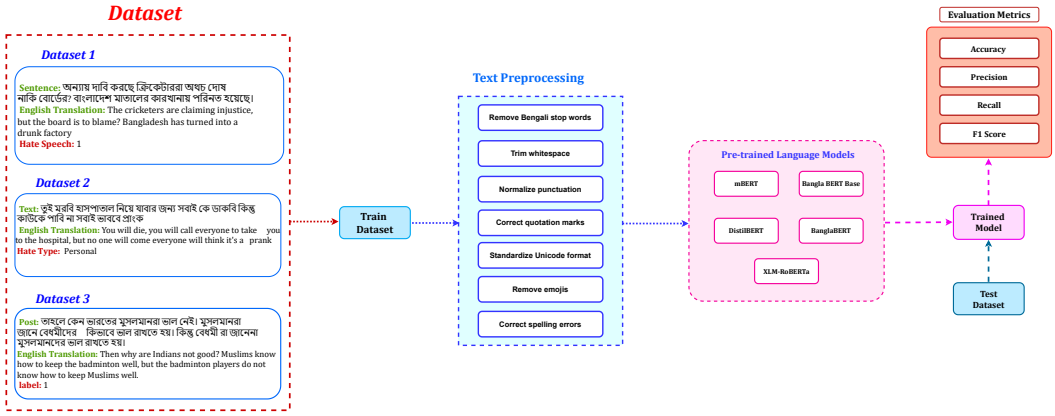


Figure 7. The diagram showcases the suggested methodology for Bangla hate speech detection using PLMs.

5.1. Experiment 1: Bangla Hate Speech Detection Using PLMs

5.1.1. Data Pre-processing

We use Dataset 1, Dataset 2, and Dataset 3, which undergo pre-processing to ensure compatibility with pre-trained language models (PLMs). A series of normalization steps are applied, specifically designed for Bengali text. These steps include handling whitespace by removing leading and trailing spaces and replacing multiple consecutive spaces with a single space, normalizing commas and other punctuation marks, correcting the placement of quotation marks, converting text to a consistent Unicode format, removing or replacing emojis depending on their relevance, normalizing numerical characters, identifying and removing English words unless contextually significant, removing common Bengali stop words, correcting common spelling mistakes, removing or replacing special characters and symbols that do not contribute to the text's meaning, and eliminating any other irrelevant information or noise present in the text. For each dataset, we divide the data into 80%, 10%, and 10% splits for training, testing, and validation, respectively, to ensure robust model evaluation and to maintain a balanced approach across all stages of the model development process.

5.1.2. Addressing Class Imbalance in Bangla Hate Speech Detection

Class imbalance, which can affect model performance, is a significant challenge in Bangla hate speech detection. To address this issue, we employ class-balanced weights in the loss function. This approach adjusts the contribution of each class to the overall loss, ensuring that minority classes are given appropriate importance during training. The balanced weight is calculated based on the inverse frequency of each class in the dataset. We calculate class weights based on the inverse frequency of each class in the dataset. This function computes weights that are inversely proportional to class frequencies, helping to mitigate the effects of class imbalance.

5.1.3. Fine-Tuning Procedure

Fine-tuning is a critical step in adapting PLMs for the specific task of Bangla hate speech detection. During this phase, the initialized PLMs are trained on a Bangla hate speech detection dataset using transfer learning techniques. This process involves updating the model parameters through gradient descent optimization algorithms to minimize the loss function. By fine-tuning, we leverage the pre-trained knowledge of the models while adapting them to the nuances of Bangla hate speech. To achieve this, we employ the AdamW optimizer, an extension of the Adam optimizer that includes weight decay. This helps prevent overfitting by regularizing the model parameters. The optimization process involves computing gradients of the loss function with respect to the model parameters and updating these parameters to reduce the loss. For the loss function, we use CrossEntropyLoss. To maintain consistency across experiments with PLMs, we set a fixed random seed value of 42, which is widely used in machine learning for reproducibility. This seed controls both the initial weight adjustments applied during fine-tuning and the shuffling of data in each training epoch.

5.1.4. Optimization of Training Settings for Enhanced Model Performance

Hyperparameters are critical settings in machine learning that govern the training process and significantly impact model performance. These include parameters such as the learning rate, batch size, and number of epochs. Properly adjusting these hyperparameters during fine-tuning is essential to optimize performance and prevent overfitting. In our experiments, we test batch sizes of 8, 16, and 32, and vary the numbers of epochs to 10, 15, 20, and 25. Additionally, we explore learning rates ranging from 0.01 to 0.001. By carefully tuning these parameters, we ensure that the fine-tuning process effectively adapts the pre-trained models to the specific task of Bangla hate speech detection, achieving optimal results.

5.1.5. Evaluation of Performance Metrics

The performance of fine-tuned PLMs is evaluated on a held-out test set using predefined evaluation criteria. Metrics such as accuracy, precision, recall, and F1-Score are utilized to objectively assess model performance. Section 6.1 present comprehensive performance metrics, providing a clear understanding of the effectiveness of fine-tuned models for Bangla hate speech detection tasks.

5.1.6. Error Analysis for Insights into Model Performance

To gain deeper insights into our model's performance, we conduct a thorough error analysis. This step is crucial for identifying and understanding the specific instances where our models struggle, allowing us to pinpoint common patterns and challenges. By closely examining the misclassified instances, we can uncover underlying issues that may not be immediately apparent through overall performance metrics alone. Section 6.4 provides a structured overview of the types of misclassifications, the number of instances for each type, the common patterns identified, the specific challenges faced by the models, and the potential improvements that can be implemented based on the analysis.

5.1.7. Post-Processing for Enhancing Prediction Accuracy

In the final step of refining the model's predictions, post-processing techniques focus on enhancing accuracy and reliability by filtering out low-confidence predictions. After generating predictions, each is assessed based on its confidence score or probability estimate. A confidence threshold is set to determine the minimum acceptable level, adaptable to specific application needs and balancing precision and recall. Predictions meeting or surpassing this threshold are retained as high-confidence, reliable outputs. Those failing to meet it are filtered out or marked for further scrutiny, indicating potential uncertainty or ambiguity requiring additional review or correction within the model's output.

5.1.8. Experimental Setup

The experiment is carried out in three different environments, which include two instances using Jupyter Notebook environments and the third using Kaggle. One Jupyter Notebook setup employs an NVIDIA GeForce RTX 3050 GPU (compute capability 8.6), an Intel Core i5 9400f CPU, and 16 GB of RAM. The other Jupyter Notebook environment boasts a more powerful NVIDIA GeForce RTX 3060 Ti GPU (compute capability 8.6), an Intel Core i5 13,600K CPU, and 32 GB of memory. Both Jupyter Notebook environments use Python version 3.8.18 with PyTorch version 1.10.0. Kaggle, on the other hand, provides access to NVIDIA Tesla P100 GPUs (compute capability 6.0), Intel Xeon CPUs, and 12.72 GB of RAM. The Kaggle setup utilizes Python 3.10.13 with PyTorch version 2.0.1.

5.1.9. Algorithm for Bangla Hate Speech Detection Using PLMs

Step 1: Data Pre-processing

1. **Input**: Let $D = \{D_1, D_2, D_3\}$ represent the datasets.
2. For each text sample $t \in D$:
   - Remove leading and trailing spaces, and replace multiple consecutive spaces with a single space:

$$t \leftarrow \text{trim}(t) \quad \text{and} \quad t \leftarrow \text{replace}(t, ' ', ' ')$$

   - Standardize punctuation characters, resulting in a new string t':

$$t' = \text{normalize_punctuation}(t)$$

   - Convert all text to a uniform Unicode format:

$$t \leftarrow \text{convert_to_unicode}(t)$$

   - Process emojis in t: retain relevant emojis and remove irrelevant ones:

$$t \leftarrow \text{remove_irrelevant_emojis}(t)$$

   - Remove English words and Bengali stop words:

$$t \leftarrow \text{remove}(t, \text{English_words} \cup \text{Bengali_stop_words})$$

   - Remove special characters and other irrelevant symbols.

Step 2: Addressing Class Imbalance

Calculate class-balanced weights w_i for each class i based on the inverse frequency:

$$w_i = \frac{1}{f_i}$$

where f_i is the frequency of class i in the dataset. Define the weighted loss function L for a prediction y with ground truth label y_{true}:

$$L = -\sum_{i=1}^{C} w_i \cdot y_{\text{true},i} \cdot \log(y_i)$$

where C represents the number of classes.

Step 3: Fine-tuning Procedure

The model parameters θ of a PLM $M(\theta)$ are updated through optimization. Given dataset D and loss function L, the AdamW optimizer is applied as follows:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \cdot \nabla_\theta L(\theta)$$

where η denotes the learning rate, and weight decay regularization is utilized to reduce overfitting.

Step 4: Optimization of Training Settings

Define and optimize a set of hyperparameters:

- Learning rate $\eta \in \{0.01, 0.001\}$
- Batch size $b \in \{8, 16, 32\}$
- Number of epochs $E \in \{10, 15, 20, 25\}$

Conduct multiple experiments to find the best configuration based on model performance.

Step 5: Evaluation Metrics

Use the following metrics to evaluate performance:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

Step 6: Error Analysis

Let $M(x)$ denote the model prediction for instance $x \in D_{\text{test}}$, and $y_{\text{true}}(x)$ be the ground truth label. Define the misclassification set as

$$\text{Error_set} = \{x \in D_{\text{test}} \mid M(x) \neq y_{\text{true}}(x)\}$$

Each instance $x \in$ Error_set is examined to categorize error types and identify patterns for model improvement.

Step 7: Post-processing for Enhanced Prediction Accuracy

After generating predictions, let $p(x)$ represent the confidence score for prediction $M(x)$. Apply a threshold τ to retain only high-confidence predictions:

$$M'(x) = \begin{cases} M(x) & \text{if } p(x) \geq \tau \\ \text{null} & \text{otherwise} \end{cases}$$

5.2. Experiment 2: Bangla Hate Speech Detection Using LLMs

5.2.1. Data Selection

To analyze the performance of Zero-Shot and Few-Shot prompting, we carefully select 400 data points from each dataset ("Dataset 1", "Dataset 2", and "Dataset 3") based on specific criteria to balance category representation and ensure statistical relevance. To confirm that our selected samples accurately represent the broader dataset, we perform statistical checks on the sample's distribution. Using a chi-square test of independence, we validate that the distribution of label categories within the sample aligns with that of the entire dataset, indicating that the sample effectively captures the diversity of the dataset's labels. Additionally, we conduct a two-sample t-test comparing the selected and unselected data points within each dataset, confirming that there are no significant differences in comment length or semantic variance. This sampling is focused on data instances where comments contain more than five words to ensure content depth for accurate model evaluation. Each label category, specifically hate speech (HS) and non-hate speech (NH), includes 100 instances per dataset, forming a consistent basis for inter-dataset comparison. In "Dataset 2", which comprises multilabel categories such as personal hate, political hate, religious hate, geopolitical hate, and gender abusive hate, we employ

stratified sampling to achieve a balanced representation of each category within the selected samples. This approach ensures that the model's performance can be fairly assessed across various types of hate speech without over-representing any specific category, which is crucial for generalization. These samples are drawn from the training subsets of each dataset, maintaining consistency with the input data used in model training. To manage the cost implications associated with using OpenAI's GPT-3.5 Turbo API and Gemini 1.5 Pro API, we limit our experiments to this subset of 400 data points per dataset.

5.2.2. Prompting Template

In our study on Bangla hate speech detection, we employ two distinct prompting strategies: zero-shot and few-shot prompting, both designed to enhance model performance. Zero-shot prompting aims to classify text without prior exposure to labels, showcasing the model's ability to generalize from existing knowledge. We emphasize the importance of clarity and specificity in prompts because variations in wording can significantly affect the model's understanding of hate speech versus non-hate speech. Thus, we refine our prompt formulations to ensure precise delineation between these categories, which we believe impacts classification accuracy.

For few-shot prompting, we provide a limited number of labeled examples in 5-shot, 10-shot, and 15-shot configurations to illustrate the incremental benefits of additional examples on contextual understanding and classification performance. These configurations are chosen to explore how varying the number of training examples influences model performance. Previous studies indicate that performance typically improves with the number of examples up to a certain threshold. Given the challenges of acquiring labeled data for hate speech classification, particularly in low-resource languages like Bangla, these configurations represent a pragmatic approach. They allow us to evaluate model performance without requiring extensive labeled datasets, which are often unfeasible. By providing specific examples, we demonstrate the model's ability to generalize from limited labeled data, a critical consideration in real-world applications where labeled instances may be scarce.

(a) Zero-shot prompting for Bangla hate speech detection

Zero-shot prompting for Bangla hate speech detection involves guiding a model to classify text as either hate speech or non-hate speech without prior training on such labels. In this context, hate speech includes offensive or derogatory language directed at individuals or groups, while non-hate speech conveys neutral or positive content. The model's objective is to accurately classify the text based solely on its content.

Figure 8 illustrates the application of zero-shot prompting for Bangla hate speech detection using the "Dataset 1" dataset. The dataset consists of premise–text pairs labeled as either hate speech or non-hate speech. The figure compares the performance of two models: GPT-3.5 Turbo and Gemini 1.5 Pro. Both models classify the text based solely on its content without prior training on the specific labels in the dataset.

Figure 9 focuses on the multilabel prompting approach for the "Dataset 2" dataset, guiding the model to predict one or more specific hate speech categories associated with a given text. The categories include personal hate, political hate, religious hate, geopolitical hate, and gender abusive hate.

Figure 10 illustrates the application of zero-shot prompting in Bangla hate speech detection using "Dataset 3". Similar to Tables 5–7 demonstrates how the model can classify text accurately as hate speech ('1') or non-hate ('0') without any training data. It highlights the model's ability to generalize from pre-existing knowledge to classify hate speech in Bengali texts.

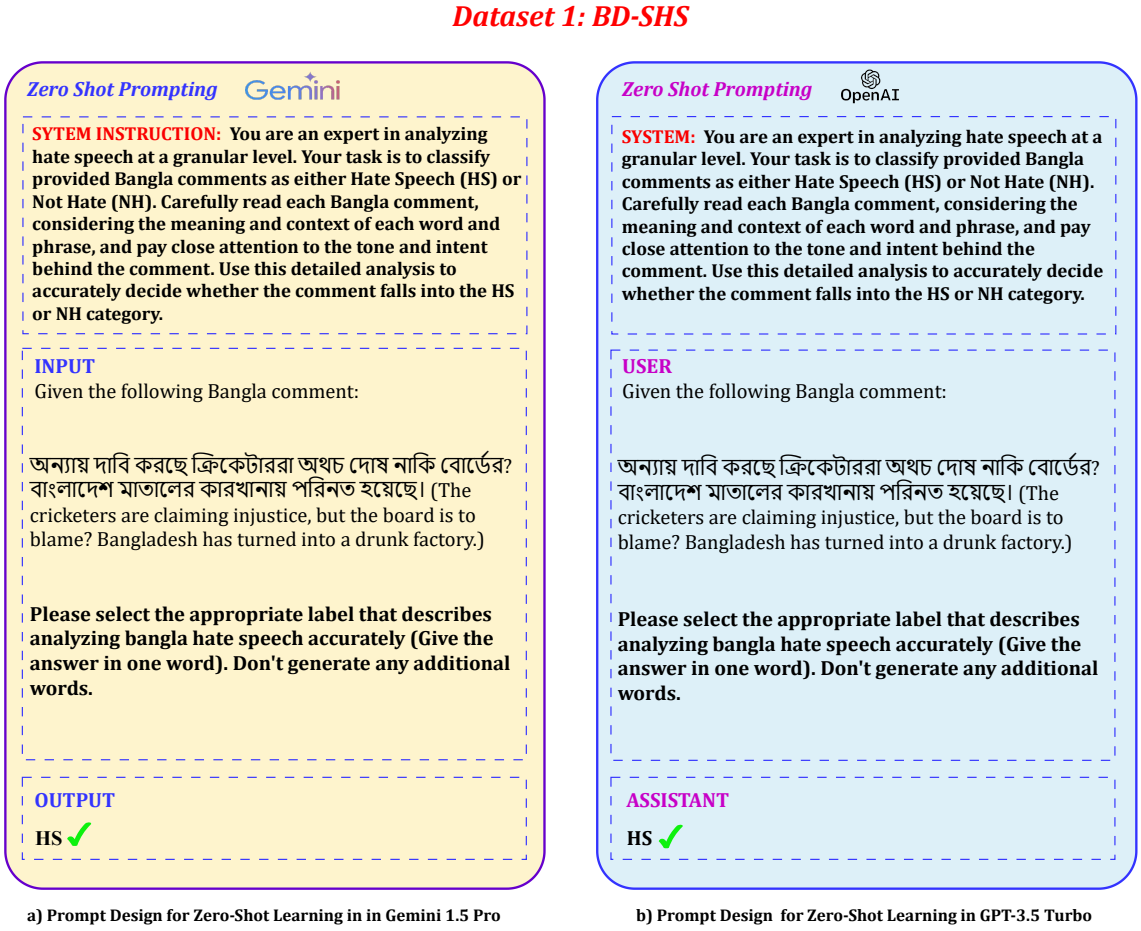


Figure 8. Illustration of prompt design for Zero-Shot Learning with Gemini 1.5 Pro and GPT-3.5 Turbo in Dataset 1.

(b) Few-shot prompting for Bangla Hate Speech Detection:
Few-shot prompting in Bangla hate speech detection involves providing the model with a limited number of example texts labeled as hate speech or non-hate speech to guide its predictions regarding the content expressed in subsequent texts. The model aims to determine the hate speech label associated with each text based on the provided examples, without extensive training on hate speech-labeled data. For instance, in a 5-shot scenario, the model receives five labeled examples for training before making predictions on new data, while in a 10-shot scenario, it receives ten examples, and so forth. In a 15-shot scenario, the model receives fifteen labeled examples, enhancing its ability to understand and classify the content expressed in the text accurately. The model utilizes the provided examples to inform its predictions, improving its classification performance as the number of examples increases.
Figure 11 showcases the application of few-shot prompting in Bangla hate speech detection using "Dataset 1". Section 6.1 present the labeled text examples used for training the model with a limited number of example texts labeled as hate speech (HS) or non-hate speech (NH). The model, without extensive training on hate speech-labeled data, accurately predicts the hate speech label associated with each text based on the provided examples.
Figure 12 demonstrates the few-shot capabilities of Gemini Pro in Bangla hate speech detection using "Dataset 2". It illustrates how Gemini Pro effectively makes accurate predictions with a small number of training instances. Section 6.1 showcases the

model's ability in a multilabel few-shot prompting scenario, where it is trained on a limited number of example texts. Each example is labeled with one or more hate speech categories, including personal hate, political hate, religious hate, geopolitical hate, and gender abusive hate. Gemini Pro uses this training to accurately predict and classify hate speech categories in subsequent texts, showcasing its robustness in handling diverse forms of hate speech in Bengali language.

Figure 13 illustrates the application of few-shot prompting in Bangla hate speech detection using "Dataset 3". This figure demonstrates how the model learns to classify text accurately as hate speech ('1') or non-hate ('0') with minimal training data. Tables 8–10 showcase the model's ability in a multilabel few-shot prompting scenario. It highlights the model's ability to generalize from a small number of labeled examples to classify hate speech in Bengali texts.

Dataset 2: Bengali Hate Speech Dataset v1.0 & v2.0

Zero Shot Prompting Gemini

SYTEM INSTRUCTION: You are an expert in analyzing hate speech at a granular level. Your task is to classify Bangla comments into the following categories: Personal Hate (hate speech targeting an individual based on personal attributes or actions), Political Hate (hate speech targeting political figures, parties, or ideologies), Religious Hate (hate speech targeting religious beliefs, practices, or followers), Geopolitical Hate (hate speech targeting countries, ethnic groups, or geopolitical entities), and Gender Abusive Hate (hate speech targeting individuals based on their gender or sexual orientation). Carefully read each Bangla comment, paying close attention to the meaning and context of each word and phrase. Understand the tone and intent behind the comment. Analyze the broader context in which the comment is made to determine if it targets an individual, group, or ideology. Based on your analysis, identify the type of hate speech and decide whether the comment falls into Personal Hate, Political Hate, Religious Hate, Geopolitical Hate, or Gender Abusive Hate.

INPUT

Given the following Bangla comment:

ভারতীয় সীমান্তরক্ষী বাহিনী বিএসএফ চীন সীমান্তে দুইজন চীনা নাগরিককে গুলি করে কি হত্যা করতে পারবে নিশ্চই না চীনা নাগরিককে গুলি করে হত্যার সঙ্গে সঙ্গেই যুদ্ধ শুরু হত পাক ভারত সীমান্তেও ভারতীয় সীমান্তরক্ষী বাহিনী এই দুইজন পাকিস্তানী নাগরিককে গুলি করে হত্যা করতে পারবে না বাংলাদেশ ভারতের বন্ধু রাষ্ট্র বাংলাদেশ ছোট রাষ্ট্র পাকিস্তান চীনের মতো শক্তি বাংলাদেশের নাই মানলাম যে দুজন বাংলাদেশি বিএসএফের গুলিতে নিহত হয়েছে তারা চোরাকারবারি মাদক ব্যবসায়ী হয়তোবা তারা গরু আনতে গিয়েছিল ভারতের সীমান্তে (Can the Indian Border Guard Force BSF shoot and kill two Chinese nationals on the Chinese border? Surely, the war would have started as soon as the Chinese nationals were shot and killed. Indian Border Guard Force cannot shoot and kill these two Pakistani nationals on the Pak-India border. Bangladesh is a friend of India. Bangladesh is a small country. Bangladesh is not as powerful as Pakistan and China. I believe that the two Bangladeshis who were killed by BSF firing were drug smugglers or they went to the Indian border to fetch cattle)

Please select the appropriate label that describes analyzing bangla hate speech accurately (Give the answer in one word). Don't generate any additional words.

OUTPUT

Geopolitical ✓

a) Prompt Design for Zero-Shot Learning in in Gemini 1.5 Pro

Zero Shot Prompting OpenAI

SYSTEM: You are an expert in analyzing hate speech at a granular level. Your task is to classify Bangla comments into the following categories: Personal Hate (hate speech targeting an individual based on personal attributes or actions), Political Hate (hate speech targeting political figures, parties, or ideologies), Religious Hate (hate speech targeting religious beliefs, practices, or followers), Geopolitical Hate (hate speech targeting countries, ethnic groups, or geopolitical entities), and Gender Abusive Hate (hate speech targeting individuals based on their gender or sexual orientation). Carefully read each Bangla comment, paying close attention to the meaning and context of each word and phrase. Understand the tone and intent behind the comment. Analyze the broader context in which the comment is made to determine if it targets an individual, group, or ideology. Based on your analysis, identify the type of hate speech and decide whether the comment falls into Personal Hate, Political Hate, Religious Hate, Geopolitical Hate, or Gender Abusive Hate.

USER

Given the following Bangla comment:

ভারতীয় সীমান্তরক্ষী বাহিনী বিএসএফ চীন সীমান্তে দুইজন চীনা নাগরিককে গুলি করে কি হত্যা করতে পারবে নিশ্চই না চীনা নাগরিককে গুলি করে হত্যার সঙ্গে সঙ্গেই যুদ্ধ শুরু হত পাক ভারত সীমান্তেও ভারতীয় সীমান্তরক্ষী বাহিনী এই দুইজন পাকিস্তানী নাগরিককে গুলি করে হত্যা করতে পারবে না বাংলাদেশ ভারতের বন্ধু রাষ্ট্র বাংলাদেশ ছোট রাষ্ট্র পাকিস্তান চীনের মতো শক্তি বাংলাদেশের নাই মানলাম যে দুজন বাংলাদেশি বিএসএফের গুলিতে নিহত হয়েছে তারা চোরাকারবারি মাদক ব্যবসায়ী হয়তোবা তারা গরু আনতে গিয়েছিল ভারতের সীমান্তে (Can the Indian Border Guard Force BSF shoot and kill two Chinese nationals on the Chinese border? Surely, the war would have started as soon as the Chinese nationals were shot and killed. Indian Border Guard Force cannot shoot and kill these two Pakistani nationals on the Pak-India border. Bangladesh is a friend of India. Bangladesh is a small country. Bangladesh is not as powerful as Pakistan and China. I believe that the two Bangladeshis who were killed by BSF firing were drug smugglers or they went to the Indian border to fetch cattle)

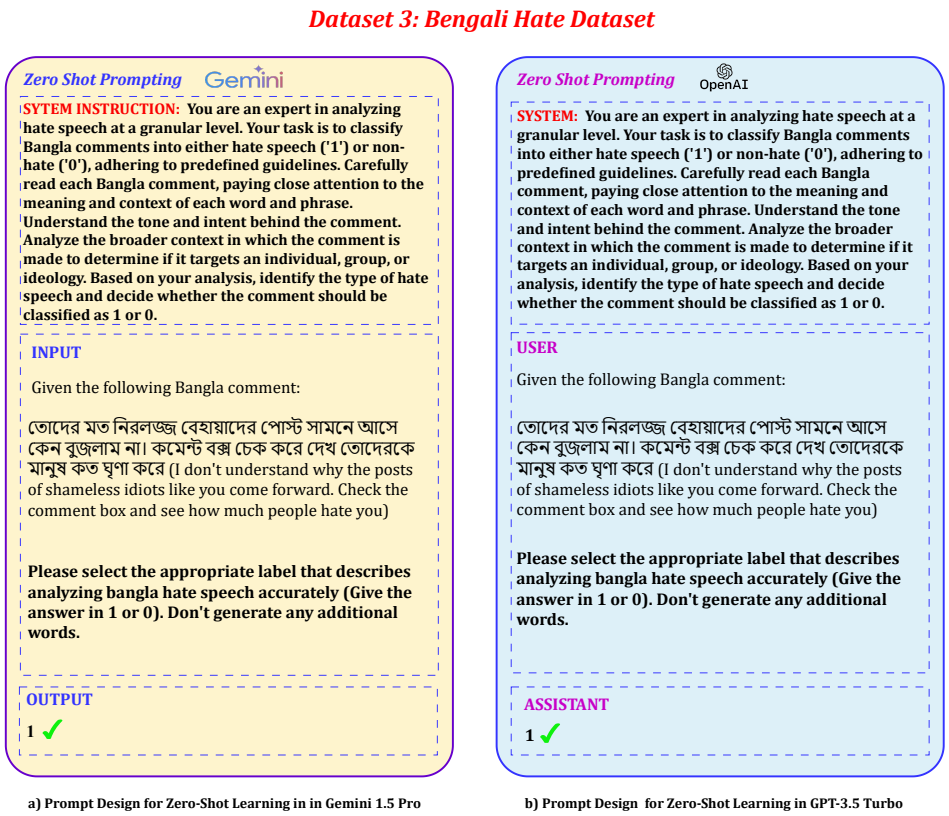
Please select the appropriate label that describes analyzing bangla hate speech accurately (Give the answer in one word). Don't generate any additional words.

ASSISTANT

Geopolitical ✓

b) Prompt Design for Zero-Shot Learning in GPT-3.5 Turbo

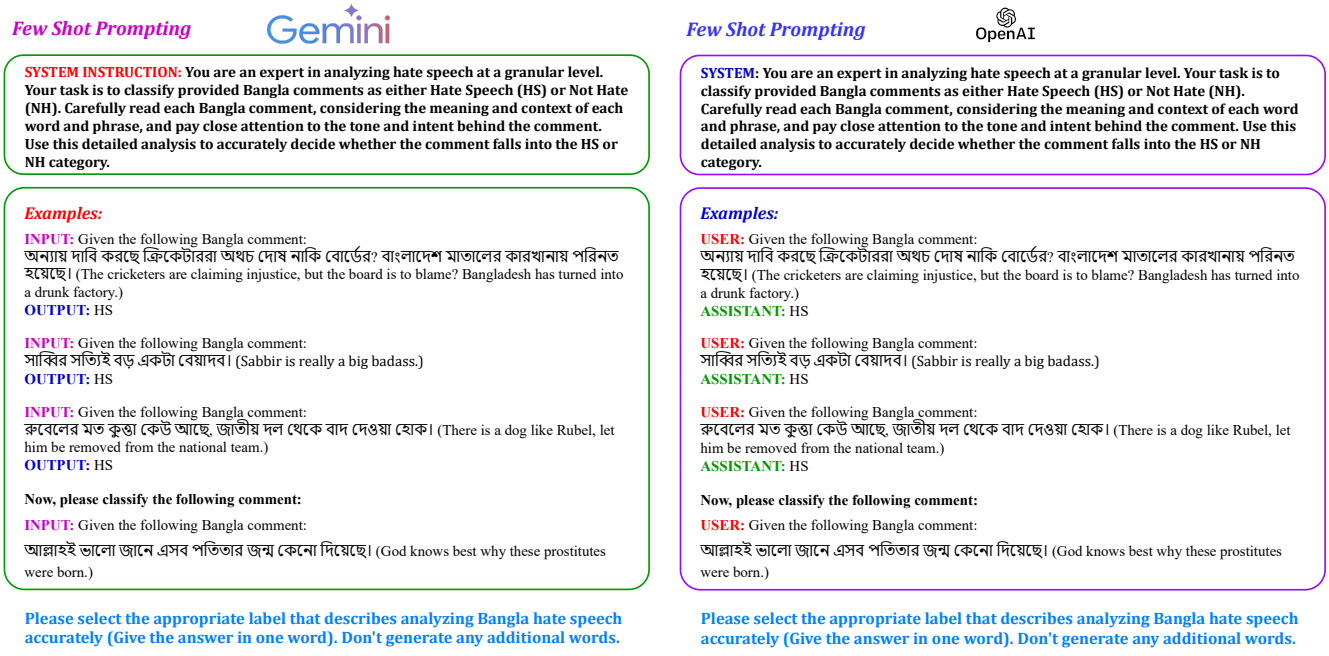
Figure 9. Illustration of prompt design for Zero-Shot Learning with Gemini 1.5 Pro and GPT-3.5 Turbo in Dataset 2.



a) Prompt Design for Zero-Shot Learning in in Gemini 1.5 Pro

b) Prompt Design for Zero-Shot Learning in GPT-3.5 Turbo

Figure 10. Illustration of prompt design for Zero-Shot Learning with Gemini 1.5 Pro and GPT-3.5 Turbo in Dataset 3.



a) Prompt Design for Few-Shot Learning in in Gemini 1.5 Pro

b) Prompt Design for Few-Shot Learning in GPT-3.5 Turbo

Figure 11. Visual representation of prompt design for Few-Shot Learning using Gemini 1.5 Pro and GPT-3.5 Turbo in Dataset 1.

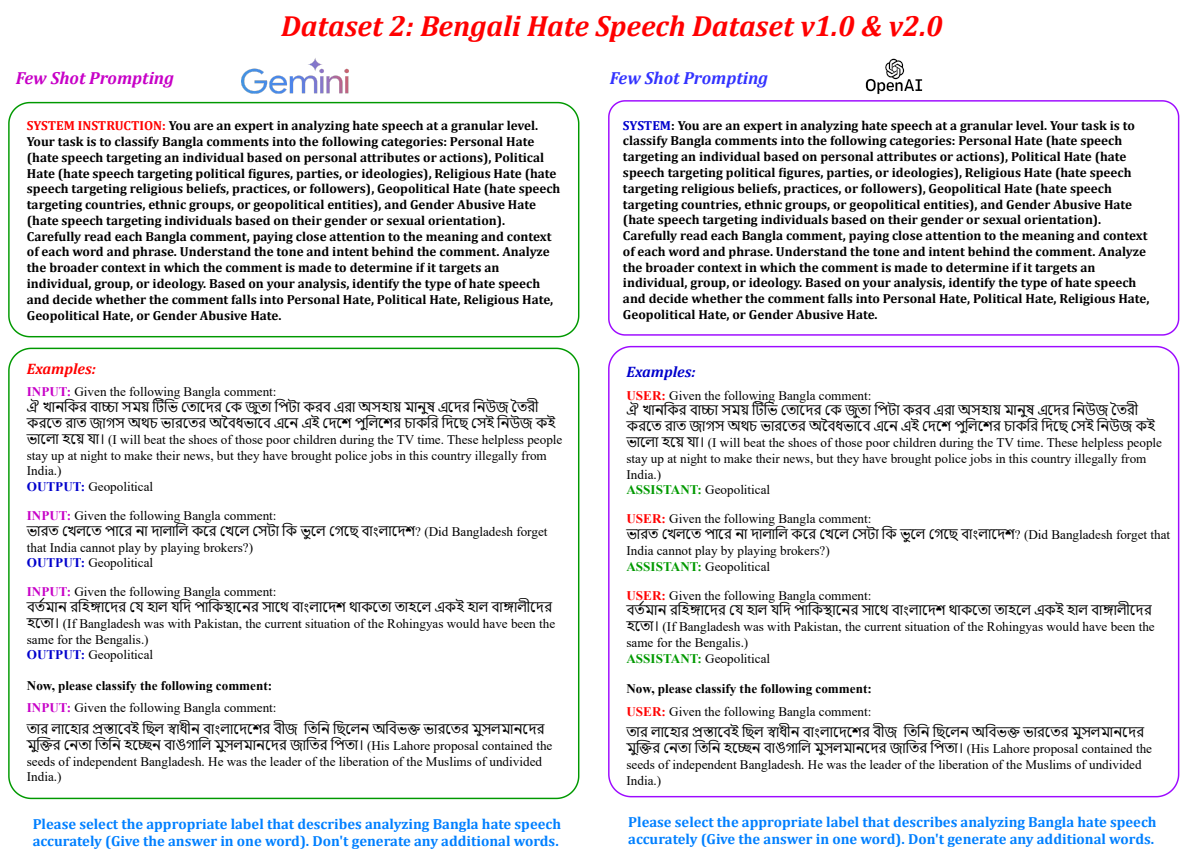


Figure 12. Visual representation of prompt design for Few-Shot Learning using Gemini 1.5 Pro and GPT-3.5 Turbo in Dataset 2.

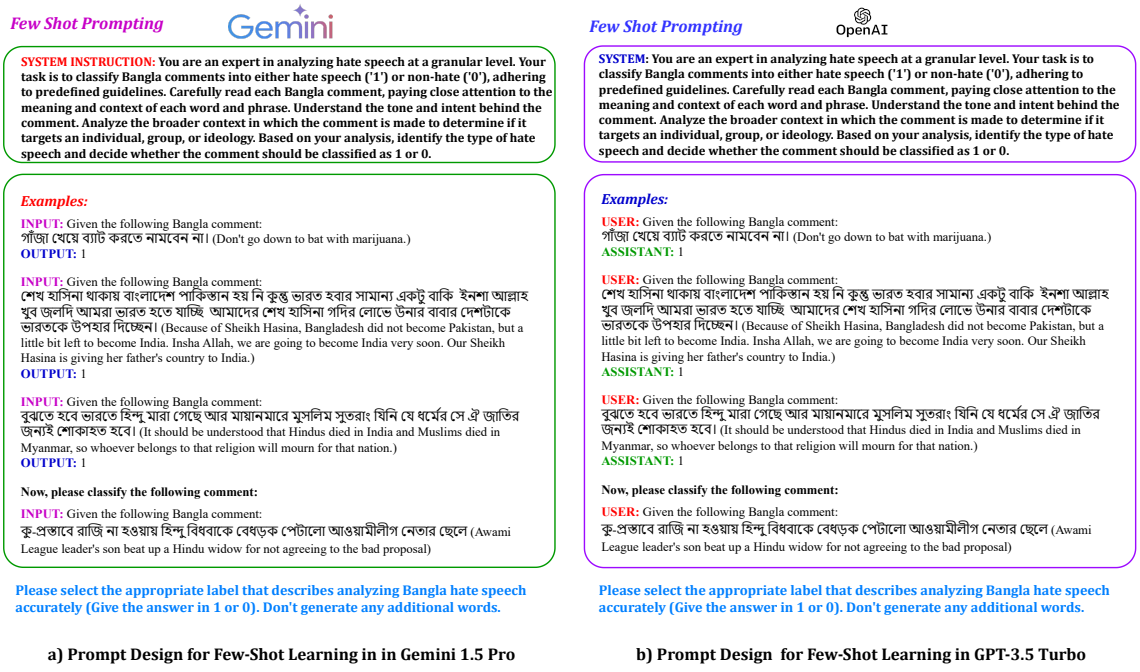


Figure 13. Visual representation of prompt design for Few-Shot Learning using Gemini 1.5 Pro and GPT-3.5 Turbo in Dataset 3.

Table 5. Performance of pre-trained language models on the BD-SHS dataset.

Model	Accuracy	Precision	Recall	F1-Score
BanglaBERT	0.9225	0.9223	0.9227	0.9219
Bangla BERT Base	0.9129	0.9130	0.9124	0.9127
mBERT	0.9128	0.9130	0.9224	0.9219
XLM-RoBERTa	0.9122	0.9136	0.9128	0.9027
sahajBERT	0.9067	0.9088	0.9014	0.9039

Table 6. Performance of pre-trained language models on the Bengali Hate Speech Dataset v1.0 and v2.0.

Model	Accuracy	Precision	Recall	F1-Score
BanglaBERT	0.8921	0.8814	0.8921	0.8920
Bangla BERT Base	0.8853	0.8903	0.8853	0.8849
mBERT	0.8793	0.8805	0.8793	0.8792
XLM-RoBERTa	0.8723	0.8732	0.8723	0.8723
sahajBERT	0.8793	0.8821	0.8793	0.8791

Table 7. Performance of pre-trained language models on the Bengali Hate Dataset.

Model	Accuracy	Precision	Recall	F1-Score
BanglaBERT	0.9042	0.9087	0.9025	0.9063
Bangla BERT Base	0.9134	0.9176	0.9112	0.9154
mBERT	0.9021	0.9143	0.9084	0.9126
XLM-RoBERTa	0.8552	0.7768	0.8184	0.7892
sahajBERT	0.8563	0.7807	0.8481	0.8014

Table 8. Performance comparison of GPT 3.5 Turbo and Gemini 1.5 Pro on three datasets in a 5-shot learning scenario.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Dataset 1	GPT 3.5 Turbo	0.9379	0.9385	0.9373	0.9379
	Gemini 1.5 Pro	0.9129	0.9130	0.9124	0.9127
Dataset 2	GPT 3.5 Turbo	0.9378	0.9382	0.9374	0.9378
	Gemini 1.5 Pro	0.9365	0.9371	0.9379	0.9364
Dataset 3	GPT 3.5 Turbo	0.9465	0.9463	0.9467	0.9465
	Gemini 1.5 Pro	0.9229	0.9230	0.9224	0.9227

Table 9. Performance comparison of GPT 3.5 Turbo and Gemini 1.5 Pro on three datasets in a 10-shot learning scenario.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Dataset 1	GPT 3.5 Turbo	0.9453	0.9448	0.9457	0.9452
	Gemini 1.5 Pro	0.9375	0.9372	0.9378	0.9376
Dataset 2	GPT 3.5 Turbo	0.9567	0.9563	0.9569	0.9566
	Gemini 1.5 Pro	0.9667	0.9663	0.9669	0.9666
Dataset 3	GPT 3.5 Turbo	0.9567	0.9563	0.9569	0.9566
	Gemini 1.5 Pro	0.9320	0.9318	0.9324	0.9319

Table 10. Performance comparison of GPT 3.5 Turbo and Gemini 1.5 Pro on three datasets in a 15-shot learning scenario.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Dataset 1	GPT 3.5 Turbo	0.9733	0.9731	0.9735	0.9733
	Gemini 1.5 Pro	0.9711	0.9702	0.9715	0.9713
Dataset 2	GPT 3.5 Turbo	0.9842	0.9840	0.9844	0.9842
	Gemini 1.5 Pro	0.9723	0.9727	0.9726	0.9723
Dataset 3	GPT 3.5 Turbo	0.9853	0.9851	0.9855	0.9853
	Gemini 1.5 Pro	0.9747	0.9743	0.9746	0.9748

5.2.3. Control Parameters for Large Language Models

In configuring control parameters for fine-tuning our Bangla hate speech detection model, each parameter in Table 11 is carefully selected to balance between model accuracy, relevance, and the nuanced detection needs of Bangla language hate speech. The temperature parameter, set to 1.0, modulates the model's response randomness; at this setting, the model achieves a balance between determinism and diversity, allowing for varied responses without sacrificing reliability. This is essential in hate speech detection, where consistent yet contextually appropriate responses are critical. Top P, also set to 1.0, enables the model to select from tokens until reaching the cumulative probability of 100%. This approach ensures that responses are both diverse and relevant by allowing the model to consider a wider range of token possibilities, enhancing its adaptability to different hate speech patterns and contexts in Bangla. Maximum Tokens is capped at 256 to ensure concise, contextually relevant outputs. This limit is designed to avoid overly verbose responses that could introduce irrelevant content, thereby streamlining the detection of hate speech without missing essential context. Frequency Penalty is set to 0.0 to avoid bias towards commonly used words, which is particularly beneficial in detecting hate speech that may rely on frequently repeated terms or slang within certain communities. By not penalizing for frequent words, the model remains unbiased toward specific vocabulary patterns while focusing on the overall context. Finally, Presence Penalty is also set to 0.0, which allows the model to generate responses without penalizing certain tokens or sequences. This setting ensures that all terms, regardless of past occurrences or discouraged patterns, are considered fairly, promoting a balanced and unbiased approach to hate speech detection in Bangla.

Table 11. Control parameters for Bangla hate speech detection model fine-tuning.

Parameter	Description	Value
Temperature	Controls randomness; lower values increase determinism, higher values increase diversity	1.0
Top P	Selects from most probable tokens; 1.0 considers tokens until cumulative probability reaches 100%, balancing diversity and relevance	1.0
Maximum Tokens	Limits number of generated tokens per response, ensuring concise and relevant outputs.	256
Frequency Penalty	Penalizes model for generating frequently used tokens; 0.0 avoids bias towards common words in hate speech detection.	0.0
Presence Penalty	Penalizes model based on presence of discouraged tokens or sequences; 0.0 ensures unbiased consideration of all text aspects in hate speech detection.	0.0

5.2.4. Cost Analysis of of Large Language Models

In this research, we utilize three datasets, necessitating a differentiated cost calculation for each based on the pricing structure of the language model APIs. For GPT-3.5 Turbo, the input cost is USD1.50 per million tokens, while the output cost is USD 2.00 per million tokens. In comparison, Gemini 1.5 Pro offers a lower input price of USD 1.25 per million tokens, but its output cost is significantly higher at USD 5.00 per million tokens. To accurately calculate token usage for each dataset, we employ the Tiktoken library, which enables us to break down text data into tokens compatible with the OpenAI API.

5.2.5. Performance Evaluation of LLMs

To objectively assess the performance of GPT-3.5 Turbo and Gemini 1.5 Pro in detecting Bangla hate speech, we use metrics such as accuracy, precision, recall, and F1-Score. These metrics provide a comprehensive understanding of each model's strengths and weaknesses. Section 6.1 summarize the performance metrics for both models, offering insights into their proficiency in handling the specific task of Bangla hate speech detection.

5.2.6. Error Analysis for LLMs

Error analysis is crucial in identifying and understanding the types and sources of errors made by LLMs in Bangla hate speech detection. By analyzing these errors, we can gain insights into the model's weaknesses and improve its performance. Section 6.4 illustrates the common error types encountered in this task, providing a detailed breakdown of the misclassifications.

5.3. Experiment 3: Bangla Hate Speech Detection Using Traditional Machine Learning Models

We utilize three distinct datasets for our analysis: BD-SHS, Bengali Hate Speech Dataset v1.0 and v2.0, and Bengali Hate Dataset. To establish a comprehensive benchmarking framework, we employ traditional machine learning models, specifically, Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes (NB), to evaluate their performance on each dataset. The data are systematically partitioned into training and testing sets using an 80/20 split, ensuring a robust evaluation of the models' predictive capabilities. Prior to model training, we undertake several preprocessing steps to prepare the text data. This process begins with tokenization, which involves breaking down the text into individual words or tokens, facilitating subsequent analysis. We then remove stop words—common words that add little meaning to the analysis—thereby refining the dataset to focus on more informative terms. To convert the textual data into a numerical format suitable for machine learning algorithms, we employ the Term Frequency-Inverse Document Frequency (TF-IDF) technique. This method not only captures the importance of words in the context of the dataset but also mitigates the impact of frequently occurring terms that may not be significant in distinguishing between classes. Furthermore, we apply lemmatization to transform words into their base or root forms, which helps in reducing inflected words to a common base form, thereby enhancing the models' ability to generalize and improve overall performance.

6. Result Analysis

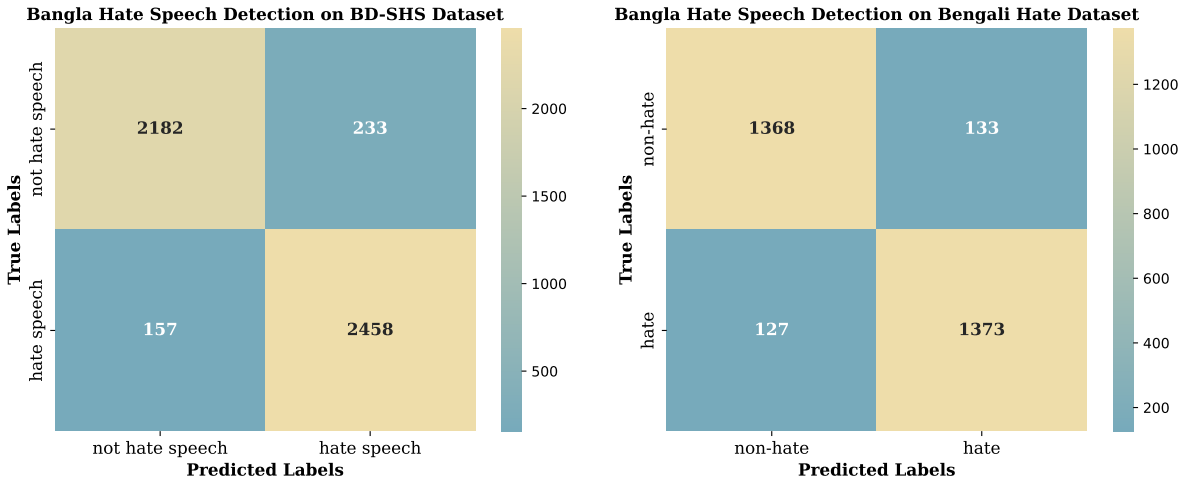
6.1. Quantitative Analysis

Table 5 demonstrates that BanglaBERT demonstrates the highest performance across all metrics, with an accuracy of 92.25%, precision of 92.23%, recall of 92.27%, and an F1-Score of 92.19%. This indicates its strong performance in hate speech detection. Bangla BERT Base performs slightly worse, with accuracy, precision, recall, and F1-Score around 91.29%, 91.30%, 91.24%, and 91.27%, respectively, showing it is a strong model, though not as effective as BanglaBERT. mBERT has an accuracy of 91.28% and precision of 91.30%, but it excels in recall (92.24%) and F1-Score (92.19%), making it comparable to BanglaBERT. XLM-RoBERTa shows an accuracy of 91.22% and precision of 91.36%, but its F1-Score drops to 90.27%, indicating a slight trade-off between precision and recall. sahajBERT has the

lowest performance among the evaluated models, with an accuracy of 90.67%, precision of 90.88%, recall of 90.14%, and an F1-Score of 90.39%. Despite performing well, sahajBERT is less effective compared to the other models listed. This analysis highlights BanglaBERT as the top-performing model for the BD-SHS dataset, followed closely by mBERT due to its high recall and F1-Score.

Table 6 shows that BanglaBERT emerges as the top performer in Bengali hate speech detection, excelling across several key metrics. It achieves the highest accuracy at 89.21%, indicating that it correctly classifies 89.21% of the samples. Its recall is equally impressive at 89.21%, demonstrating its effectiveness in identifying actual positive samples. BanglaBERT also leads with an F1-Score of 89.20%, reflecting a balanced performance between precision and recall. However, its precision is slightly lower at 88.05%, which means that while it identifies most positive samples, a small proportion of its positive predictions is incorrect. In close competition, Bangla BERT Base exhibits an accuracy of 88.53% and shines with the highest precision among the models at 89.03%. This indicates that it has a high ratio of true positive predictions to total predicted positives. Its recall and F1-Score are 88.53% and 88.49%, respectively, showcasing its reliability and balanced performance, though marginally behind BanglaBERT in recall and F1-Score. Both mBERT and sahajBERT present similar results, each attaining an accuracy of 87.93%. Their precision and F1-Scores are closely matched, with mBERT achieving a precision of 88.14% and an F1-Score of 87.92%, while sahajBERT scores 88.21% in precision and 87.91% in F1-Score. These results suggest that both models are competent, with minor variations in their ability to balance precision and recall. XLM-RoBERTa, while still competitive, ranks lowest among the evaluated models. It achieves an accuracy of 87.23%, precision of 87.32%, recall of 87.23%, and an F1-Score of 87.23%. Despite being at the lower end of the performance spectrum in this comparison, XLM-RoBERTa still offers a robust performance, underscoring the overall competitive nature of these models in handling Bengali hate speech detection.

Table 7 provides a comprehensive evaluation of several models on the Bengali Hate Dataset, revealing that Bangla BERT Base achieves the highest accuracy at 91.34%, indicating it correctly classifies approximately 91.34% of the instances. Following closely, BanglaBERT and mBERT perform well with accuracies of 90.42% and 90.21%, respectively, while sahajBERT and XLM-RoBERTa have lower accuracies of 85.63% and 85.52%. In terms of precision, mBERT stands out with the highest value of 91.43%, suggesting a high rate of correctly identified positive instances, followed by Bangla BERT Base and BanglaBERT with similar high precision values of 91.76% and 90.87%. SahajBERT and XLM-RoBERTa, however, have lower precision values of 78.07% and 77.68%, indicating more false positives. Bangla BERT Base again leads with a recall of 91.12%, closely followed by BanglaBERT at 90.25% and mBERT at 90.84%, whereas sahajBERT and XLM-RoBERTa have lower recall values of 84.81% and 81.84%, respectively, indicating they miss more positive instances. The highest F1-Score is achieved by Bangla BERT Base at 91.54%, reflecting a strong balance between precision and recall, with BanglaBERT and mBERT also performing well with F1-Scores of 90.63% and 91.26%, respectively. Conversely, sahajBERT and XLM-RoBERTa have lower F1-Scores of 80.14% and 78.92%, reflecting their lower precision and recall. Overall, Bangla BERT Base demonstrates the best performance across all metrics, making it the most effective model for the Bengali Hate Dataset, while BanglaBERT and mBERT also show strong performance, particularly in precision and recall, making them reliable choices for hate speech detection. In contrast, sahajBERT and XLM-RoBERTa show comparatively lower performance across all metrics, suggesting they are more prone to false positives and false negatives, respectively. Furthermore, Figures 14 and 15 present the confusion metrics for all the pre-trained language models (PLMs).



(**a**) Dataset 1 (BanglaBERT) (**b**) Dataset 3 (Bangla BERT Base)

Figure 14. Visualization of confusion matrices showing the performance of BanglaBERT and Bangla BERT Base in hate speech detection.

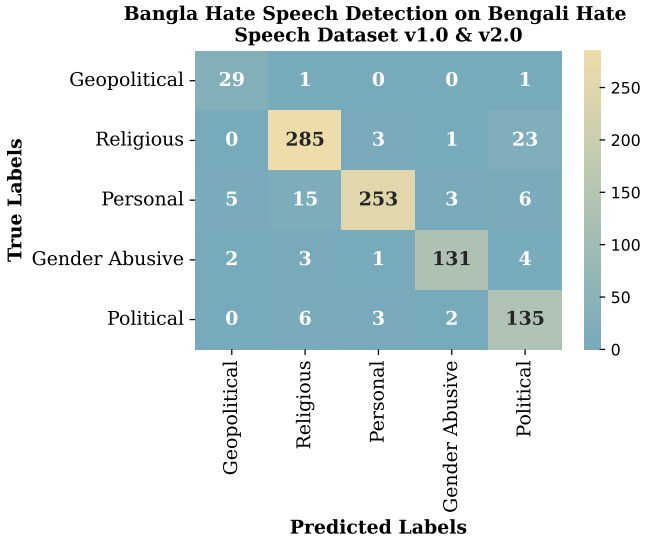


Figure 15. Visualization of confusion matrices showing the performance of BanglaBERT in hate speech detection for Dataset 2.

To further evaluate the performance differences among the pre-trained language models on the BD-SHS dataset, we conduct a one-way ANOVA analysis. This statistical test is applied to assess whether there are significant differences in accuracy, precision, recall, and F1-Score across the models presented in Tables 5–7. The null hypothesis (H0) states that there are no significant differences in the means of the performance metrics among the models, while the alternative hypothesis (H1) proposes that at least one model exhibits a significantly different performance. The results of the ANOVA indicate a statistically significant effect of the model type on performance metrics ($p < 0.05$). Consequently, post hoc tests, such as Tukey's HSD, are performed to identify specific model comparisons contributing to the significant differences observed.

Table 12 presents a comparative analysis of the performance metrics of two language models, GPT 3.5 Turbo and Gemini 1.5 Pro, across three distinct datasets in a Zero-Shot Learning setting. In Dataset 1, GPT 3.5 Turbo achieves an accuracy of 86.61%, with precision, recall, and F1-Score values closely aligned at 86.69%, 86.71%, and 86.65%, respectively, while Gemini 1.5 Pro achieves 82.20% accuracy, with precision, recall, and F1-Score values around

82.18%, 82.24%, and 82.19%, respectively. Moving to Dataset 2, GPT 3.5 Turbo demonstrates an accuracy of 80.29%, with precision, recall, and F1-Score values of approximately 80.31%, 80.24%, and 80.27%, respectively, whereas Gemini 1.5 Pro shows a slightly higher accuracy of 81.30%, maintaining consistent precision, recall, and F1-Score values of 81.30%. In Dataset 3, GPT 3.5 Turbo achieves an accuracy of 83.31%, with precision, recall, and F1-Score values all hovering around 83.30% and 83.31%, respectively, while Gemini 1.5 Pro demonstrates superior performance with an accuracy of 87.76%, achieving precision, recall, and F1-Score values of 87.82%, 87.69%, and 87.75%, respectively. Overall, both models show competitive performance metrics across datasets, with GPT 3.5 Turbo maintaining stable performance and Gemini 1.5 Pro exhibiting noticeable improvements, particularly in Dataset 3. However, it is important to note that these zero-shot results generally indicate worse performance compared to models fine-tuned on specific tasks, such as pre-trained language models, due to the lack of task-specific training and adaptation.

Table 12. Performance comparison of GPT 3.5 Turbo and Gemini 1.5 Pro on three datasets in a Zero-Shot Learning scenario.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Dataset 1	GPT 3.5 Turbo	0.8661	0.8669	0.8671	0.8665
	Gemini 1.5 Pro	0.8220	0.8218	0.8224	0.8219
Dataset 2	GPT 3.5 Turbo	0.8029	0.8031	0.8024	0.8027
	Gemini 1.5 Pro	0.8130	0.8130	0.8130	0.8130
Dataset 3	GPT 3.5 Turbo	0.8331	0.8330	0.8331	0.8331
	Gemini 1.5 Pro	0.8776	0.8782	0.8769	0.8775

Table 8 showcases the performance of two Large Language Models, GPT-3.5 Turbo and Gemini-1.5 Pro, across three datasets in a 5-shot learning scenario. GPT-3.5 Turbo consistently outperforms Gemini-1.5 Pro on all datasets, with the most pronounced difference in Dataset 1 (approximately 2.5 percentage points across all metrics) and the least in Dataset 2 (less than 0.2 percentage points). For Dataset 1, GPT-3.5 Turbo achieves 93.79% accuracy, 93.85% precision, 93.73% recall, and 93.79% F1-Score, while Gemini-1.5 Pro scores 91.29%, 91.30%, 91.24%, and 91.27%, respectively. On Dataset 2, GPT-3.5 Turbo's metrics remain strong and consistent with those of Dataset 1, whereas Gemini-1.5 Pro improves significantly to 93.65% accuracy, 93.71% precision, 93.79% recall, and 93.64% F1-Score. For Dataset 3, GPT-3.5 Turbo demonstrates its best performance with metrics around 94.65%, compared to Gemini-1.5 Pro's 92.29% accuracy, 92.30% precision, 92.24% recall, and 92.27% F1-Score. Overall, GPT-3.5 Turbo shows higher consistency and robustness across datasets, while Gemini-1.5 Pro exhibits more variation, indicating potential sensitivity to dataset characteristics. Notably, the 5-shot learning approach consistently outperforms both zero-shot and pre-trained language models due to its ability to leverage a small amount of task-specific training data, allowing for improved adaptation to the task at hand.

Table 9 provides a detailed performance comparison of two advanced Large Language Models, GPT-3.5 Turbo and Gemini 1.5 Pro, across three different datasets in a 10-shot learning scenario using four key evaluation metrics: accuracy, precision, recall, and F1-Score. For Dataset 1, GPT-3.5 Turbo demonstrates strong performance with an accuracy of 94.53%, precision of 94.48%, recall of 94.57%, and F1-Score of 94.52%, outperforming Gemini 1.5 Pro which has an accuracy of 93.75%, precision of 93.72%, recall of 93.78%, and F1-Score of 93.76%. On Dataset 2, GPT-3.5 Turbo maintains high performance with an accuracy of 95.67%, precision of 95.63%, recall of 95.69%, and F1-Score of 95.66% but is surpassed by Gemini 1.5 Pro, which achieves an accuracy of 96.67%, precision of 96.63%, recall of 96.69%, and F1-Score of 96.66%. For Dataset 3, GPT-3.5 Turbo again shows strong performance with an accuracy of 95.67%, precision of 95.63%, recall of 95.69%, and F1-Score of 95.66%, whereas Gemini 1.5 Pro performs less well with an accuracy of 93.20%, precision of 93.18%, recall of 93.24%, and F1-Score of 93.19%. Overall, GPT-3.5 Turbo generally outperforms

Gemini 1.5 Pro on Datasets 1 and 3, while Gemini 1.5 Pro shows superior performance on Dataset 2, with trends in precision, recall, and F1-Score following the accuracy trends. The 10-shot learning approach consistently demonstrates better performance compared to 5-shot learning, zero-shot, and pre-trained language models. This improvement is attributed to the increased amount of task-specific training data, allowing the models to better adapt and generalize to the evaluation tasks, resulting in higher accuracy and more balanced precision–recall trade-offs.

In the comparative analysis presented in Table 10, GPT 3.5 Turbo and Gemini 1.5 Pro are evaluated across three distinct datasets in a 15-shot learning scenario. Across Dataset 1, GPT 3.5 Turbo slightly outperforms Gemini 1.5 Pro with higher accuracy (97.33% compared to 97.11%), precision (97.31% compared to 97.02%), recall (97.35% compared to 97.15%), and F1-Score (97.33% compared to 97.13%). Moving to Dataset 2 and Dataset 3, GPT 3.5 Turbo consistently demonstrates superior performance with noticeably higher accuracy, precision, recall, and F1-Scores compared to Gemini 1.5 Pro. Specifically, in Dataset 2, GPT 3.5 Turbo achieves an accuracy and F1-Score of 98.42%, while Gemini 1.5 Pro scores 97.23% and 97.23%, respectively. In Dataset 3, GPT 3.5 Turbo maintains high metrics with 98.53% accuracy and 98.53% F1-Score, whereas Gemini 1.5 Pro achieves 97.47% and 97.48%. This comprehensive analysis highlights GPT 3.5 Turbo's consistent superiority over Gemini 1.5 Pro across diverse datasets in the 15-shot learning scenario. The 15-shot learning approach demonstrates superior performance compared to the 5-shot and 10-shot learning methods, as well as the zero-shot and pre-trained language models. This improvement can be attributed to the increased availability of task-specific training data, allowing the models to refine their understanding and optimization for the evaluation tasks, resulting in higher accuracy and precision–recall balance. Additionally, the 15-shot learning scenario benefits from a larger sample of task-specific examples during training, facilitating deeper model adaptation and more accurate predictions.

Table 13 provides a comprehensive performance comparison of four traditional machine learning models—Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes (NB)—across three distinct datasets. In Dataset 1, Naive Bayes achieves the highest accuracy at 74.35%, demonstrating its efficacy in handling the data characteristics, while Logistic Regression closely follows with an accuracy of 74.01%. Random Forest and SVM perform slightly worse, with accuracies of 73.23% and 72.12%, respectively. This indicates that, while all models show comparable performance, Naive Bayes and LR are particularly adept in this scenario. In Dataset 2, Logistic Regression leads with an accuracy of 75.12%, showcasing its strength in this dataset. SVM, despite being the second-best performer overall, records an accuracy of 73.34%, which is significantly lower than the performance of LR. Random Forest and Naive Bayes trail with accuracies of 74.05% and 72.56%, respectively, further highlighting the competitive edge of LR in this context. Dataset 3 reveals a similar trend, where Naive Bayes again excels, achieving the highest accuracy of 75.92%. Random Forest follows with an accuracy of 73.56%, while SVM lags behind at 71.23%. This dataset reinforces Naive Bayes' robust performance across varying data characteristics. Overall, the results underscore that Naive Bayes consistently provides strong performance across all datasets, particularly excelling in Dataset 3, while Logistic Regression stands out in Dataset 2.

However, it is essential to note that the reported metrics lack confidence intervals, which limits our ability to fully assess the uncertainty and variability of these performance estimates. Confidence intervals provide a critical range within which the true performance of the models is likely to fall, allowing for a more nuanced understanding of their reliability. For instance, a model reported with an accuracy of 74.35% may have a confidence interval of 72% to 76%, indicating that while the model performs well, there is still some uncertainty about its effectiveness across different samples. Conversely, if another model has an accuracy of 75.12% with a confidence interval of 70% to 80%, this suggests a broader range of uncertainty, which could affect the interpretability of the results.

Table 13. Performance comparison of traditional machine learning models on three datasets.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
Dataset 1	LR	0.7401	0.7384	0.7425	0.7404
	SVM	0.7212	0.7190	0.7240	0.7216
	RF	0.7323	0.7301	0.7350	0.7326
	NB	0.7435	0.7400	0.7450	0.7425
Dataset 2	LR	0.7512	0.7485	0.7520	0.7502
	SVM	0.7334	0.7310	0.7360	0.7335
	RF	0.7405	0.7380	0.7420	0.7402
	NB	0.7256	0.7230	0.7280	0.7255
Dataset 3	LR	0.7478	0.7450	0.7500	0.7474
	SVM	0.7123	0.7100	0.7150	0.7125
	RF	0.7356	0.7330	0.7380	0.7355
	NB	0.7592	0.7570	0.7610	0.7582

Moreover, confidence intervals are particularly valuable when comparing models; overlapping intervals can suggest that the differences in performance are not statistically significant, while non-overlapping intervals can indicate clear distinctions between model effectiveness. Therefore, the absence of confidence intervals means we cannot confidently claim which model performs better or how reliable these performance metrics are in real-world applications (see Figure 16).

The comparative analysis indicates that while traditional models can yield valuable insights, their effectiveness is significantly overshadowed by the performance of pre-trained language models (PLMs) and Large Language Models (LLMs) in our experiment. Naive Bayes and Logistic Regression, although promising for accurate predictions, are notably less effective compared to the advanced capabilities exhibited by PLMs and LLMs, highlighting a clear advantage for these modern approaches in this study.

Figure 10 illustrates an error analysis of Bangla language social media posts evaluated by PLMs, emphasizing the critical role of error analysis in evaluating model performance. Firstly, it sheds light on the model's proficiency in interpreting nuanced language, evident in its misclassification of sentiments related to war, equality, and peace in the first post. This highlights the necessity for the model to better comprehend complex socio-political discussions for accurate sentiment analysis and contextual understanding in sensitive topics. Secondly, error analysis identifies specific challenges faced by the model, such as its difficulty in distinguishing neutral or rhetorical statements from positive sentiments as observed in the misclassifications of the second and third posts. Understanding these challenges is pivotal for refining model training strategies to enhance its performance in real-world applications where precise sentiment classification is crucial. Moreover, correct classifications, exemplified by the fourth and sixth posts, validate the model's capability to accurately interpret sentiments concerning human rights issues and neutral content, respectively. These instances underscore areas where the model excels and provides reliable predictions, bolstering confidence in its performance. Conversely, the misclassification of the fifth post, which discusses intricate political and religious themes, exposes significant hurdles in the model's comprehension of such culturally specific references. This insight underscores the need for targeted improvements to broaden the model's understanding of diverse content, thereby enhancing its overall reliability in Bangla language processing tasks. Ultimately, conducting thorough error analysis yields actionable insights for enhancing PLMs' proficiency in sentiment analysis and contextual understanding of Bangla social media discourse. By addressing the identified challenges and leveraging strengths, these efforts aim to bolster the models' accuracy and effectiveness in handling complex linguistic nuances and socio-cultural contexts.

Post Content	English Translation	Actual Label	Predicted Label	Error Type	Possible Reason for Error
ভারত বা পাকিস্তান সবার বাচার অধিকার আছে তবে তাহারই বাচার অধিকার নাই যারা শান্তিপ্রিয় মানুষকে কস্ট দেয় আমি কোনো জাতি বিরোধী নই আমার কাছে সকল দেশের ও সকল ধর্মের মানুষ সমান যদি সে শান্তি প্রিয় হয় আর যুদ্ধ লাগলে কেউ সফল হবে না কারন তখন ভারত ও পাকিস্তান লাসের দেশ হয়ে যাবে মরবে হিন্দু মুসলিম সবাই মরবে সাধারন মানুষ	In India or Pakistan, everyone has the right to live, but there is no right to live for those who cost peaceful people. I am not anti-any nation. To me, all people of all countries and religions are equal. If war becomes preferable, no one will succeed because then India and Pakistan will become countries of corpses. Hindus, Muslims, everyone will die, ordinary people will die.	1	0 ✗	Misclassification	The model misinterprets the nuanced language discussing peace and war, struggling with the complex sentence structure and contextual meaning.
অনেক কিছু লিখার ছিল। কিন্তু ছোট ভাই আর বন্ধুরা দেখে ফেলবে তাই আর কিছু লিখলাম না এটাওত লিখে দিলাম।লিখলাম মানেত দেখলাম।কী দেখলাম?	I have written a lot. But the younger brother and friends will see it, so I didn't write anything more, I wrote this too. What did I see?	0	1 ✗	Misclassification	The model fails to understand the intention behind the message, struggling with the context that involves familial relationships and social dynamics.
কাজটা কি ভাল হলো? আমার এই ছোট জীবন থেকে আড়াই মিনিট কেড়ে নিয়ে আপনার কি লাভ হলো?	Was the work good? Taking half a minute from my short life, what did you gain?	0	1 ✗	Misclassification	The model misunderstands the rhetorical question about the value of time and personal effort, struggling with the nuanced language and context.
বাংলাদেশে যদি হিন্দুদের উপর কোন সহিংসতা ঘটে তাহলে ভারত উদ্বেগ জানায় শুধু ভারত নয় বাংলাদেশেও অনেক মানবাধিকার কর্মি এবং বিশিষ্ট জনেরা চিন্তিত থাকে কিন্তু ভারতে এবং অন্যান্য দেশে যদি কোন মুসলমানের উপর অত্যাচার করা হয় তখন কোন মানবাধিকার কর্মি এবং কোন বিশিষ্ট জন খিজে পাওয়া যায়না তখন তারা কিছু দেখতে পায়না শুনতেও পায়না	If violence occurs against Hindus in Bangladesh, it not only bothers India but also concerns many human rights activists and prominent people in Bangladesh. But if any Muslim is oppressed in India or other countries, no human rights activist or prominent person finds anything or hears anything.	1	1 ✓	Correct Classification	None
ভারত একটা হিন্দু রাষ্ট্র হয়ে মুসলিমদের সাহায্য করেছে ঠাই দিয়েছে আবার শত্রুর বিরুদ্ধে যুদ্ধ করেছে আর বাংলাদেশ মুসলিম হয়ে মুসলিমদের সাহায্য করতে পারেন না	India has become a Hindu state, helping Muslims by putting stars, and has fought against enemies, and Bangladesh, being Muslim, cannot help Muslims.	1	0 ✗	Misclassification	The model struggles with the complex political and religious references, leading to misinterpretation of the sentiment and context.
আমার এই অন্ধ শিশু শিল্পীর জন্য সবাই দোয়া করবেন এবং আমার চ্যানেলটি সাবস্ক্রাইব করুন	Everyone pray for my blind child artist and subscribe my channel	0	0 ✓	Correct Classification	None

Figure 16. Error analysis of Large Language Models on Bangla hate speech detection.

6.2. Hallucination Analysis

In the context of Bangla hate speech detection, the challenge of hallucinations [28,29] in LLMs becomes particularly significant. Given the sensitive nature of hate speech, it is crucial for models to provide accurate and reliable outputs. Hallucinations in this domain can lead to the misidentification of hate speech, either by falsely flagging benign content or by missing harmful content. This can have serious implications for content moderation, public discourse, and community safety. Bangla, being a low-resource language, presents additional complexities. The lack of extensive, high-quality datasets for training and evaluation makes it more challenging to ensure the accuracy and robustness of LLMs

in detecting hate speech. Furthermore, the cultural and linguistic nuances of Bangla require careful consideration to avoid misinterpretations that could result in hallucinations. Hallucinations in LLMs, where the models generate factually incorrect or misleading information, present a significant challenge, especially for critical applications such as hate speech detection. While this phenomenon has been extensively studied in more widely spoken languages, the Bangla language remains underexplored. Given the complexity and rich linguistic features of Bangla, our study aims to bridge this gap by conducting a detailed case study on hallucinations in LLMs for Bangla hate speech detection. The two LLMs we experimented with are GPT-3.5 Turbo and Gemini 1.5 Pro. We conducted a detailed analysis of factual and linguistic errors in GPT-3.5 Turbo and Gemini 1.5 Pro. Our approach involved generating hate speech detection outputs in Bangla and systematically evaluating these outputs for various types of hallucinations.

1. **Factuality:** We assessed the factual accuracy of the generated texts by identifying instances where GPT-3.5 Turbo and Gemini 1.5 Pro produced false or misleading information. This involved cross-referencing the outputs with verified facts to measure the extent of hallucinations. In the context of hate speech detection, factuality checks ensured that the models correctly identified and categorized hate speech instances without fabricating or misrepresenting information.
2. **Correctness:** We evaluated the correctness of the hate speech detection outputs by comparing them with expert annotations and verified datasets. This step ensured that the models' outputs were aligned with established standards. Correctness evaluation was crucial to verify that the models accurately detected and flagged hate speech in Bangla, maintaining high precision and recall.
3. **Linguistic Errors:** We analyzed the texts for grammatical, syntactic, and semantic inaccuracies. This step helped us understand how linguistic errors impacted the overall quality of the generated content. Identifying linguistic errors allowed us to refine the models' outputs, ensuring they were not only accurate but also clear and comprehensible, which is vital for sensitive applications like hate speech detection.
4. **Reasons for Incorrect Facts:** We explored the potential causes of hallucinations by examining factors such as training data limitations, model architecture, and contextual understanding capabilities. Understanding the root causes of incorrect facts helped us develop strategies to mitigate these hallucinations, improving the reliability of hate speech detection outputs.
5. **Factuality versus Readability:** We compared the factual accuracy of the texts with their readability to determine if improvements in one aspect affected the other. Balancing factuality and readability ensured that the hate speech detection outputs were both accurate and easy to understand, facilitating better use and interpretation by users.

6.3. Comparison with Existing Approaches

Table 14 highlights the distinctions between existing approaches and our proposed method in addressing key challenges. Unlike previous works, which primarily rely on conventional techniques or limited datasets, our approach leverages advanced models and enriched datasets for superior performance. This analysis underscores the innovative aspects and improved outcomes of our research.

6.4. Error Analysis

Our approach demonstrates superior performance in Bangla hate speech detection compared to existing methods. By harnessing the power of Large Language Models (LLMs), we achieved exceptional results using minimal labeled data. This highlights the effectiveness of LLMs in tackling intricate tasks such as hate speech detection in Bengali, outperforming traditional methodologies in terms of both accuracy and scalability. Table 14 provides a detailed comparison with existing approaches.

Table 14. Detailed comparison of approaches from selected research papers in Bangla hate speech detection.

Paper	Dataset	Approach	Performance Metrics	Comments
This paper	BD-SHS, Bengali Hate Speech Dataset v1.0, Bengali Hate Speech Dataset v2.0, Bengali Hate Dataset	In the context of 15-shot learning, both GPT 3.5 Turbo and Gemini 1.5 Pro are evaluated.	97.33% in Dataset 1, 98.42% in Dataset 2, and 98.53% in Dataset 3.	GPT 3.5 Turbo excels particularly in Dataset 1, Dataset 2 and Dataset 3, demonstrating significantly higher accuracy compared to Gemini 1.5 Pro.
Saroar et al. [2]	Offensive posts filtered: 8.5 k. Non-offensive posts identified: 8.5 k. Final manually labeled dataset: 15 k posts (balanced with 7.5 k offensive and 7.5 k non-offensive posts).	The existing BanglaBERT model, pre-trained on 18.6 GB of Bengali text (1 million steps over 3 billion tokens), is retrained with 1.5 million offensive posts for 15 epochs (almost 2 million steps) in batches of 64 samples using MLM and the Adam optimizer with a learning rate of 5×10^{-5}.	Bangla Hate BERT: Accuracy—94.3%, F1-Score—94.1%	The dataset is balanced with equally offensive and non-offensive posts and high-quality labels from manual annotation. A limitation is the need for a large corpus for traditional models. However, LLMs can generalize from large-scale pre-existing datasets, reducing the need for extensive domain-specific annotated data.
Rezaul et al. [15]	The dataset has 100,000 annotated hate speech statements, covering political, personal, gender-based, geopolitical, and religious hate, created with a bootstrapping and semi-automatic annotation approach.	The MC-LSTM integrates BengFastText embeddings for hate speech detection, capturing contextual and semantic information from Bengali texts. Additionally, traditional ML models (SVM, KNN, LR, NB, DT, RF, and GBT) and embedding models (Word2Vec and GloVe) are trained for a comprehensive performance comparison.	Achieves up to 90.45% F1-Score.	The authors' traditional model training approach does not address the need for a large corpus. LLMs mitigate this by generalizing from large pre-existing datasets, showing that LLMs offer a more efficient and adaptive alternative to traditional methods.
Nauros et al. [4]	BD-SHS, the largest Bangla hate speech dataset, consists of 50,281 comments manually labeled in different social contexts. In total, 24,156 comments are tagged as hate speech (HS).	Various ML models, including SVM and Bi-LSTM, are used to identify and categorize hate speech, combined with word embeddings like pre-trained formal (BFT, MFT) and informal (IFT) embeddings.	Weighted F1-Score of 91.00%	LLMs can be leveraged to mitigate the need for extensive labeled data, which is often time-consuming to gather, by utilizing Few-Shot Learning techniques and transfer learning to achieve robust performance with minimal annotated examples.
Jobair et al. [3]	The new dataset consists of 8600 user comments from Facebook and YouTube, categorized into sports, religion, politics, entertainment, and others.	Conducts a comprehensive study using five distinct models to analyze abusive language in Bengali. The models tested include CNN, LSTM, Bi-LSTM, GRU, and BERT. Additionally, we run these models on an existing dataset of 30,000 records to compare performance across different datasets.	The BERT model outperforms others with a 97% accuracy and an F1-Score of 96%.	LLMs can effectively minimize the reliance on extensive labeled datasets. Leveraging techniques like Few-Shot Learning, Zero-Shot Learning, and transfer learning, LLMs achieve robust performance even with minimal annotated examples, circumventing the time-consuming process of gathering extensive labeled data.

7. Limitations

The study primarily focused on zero-shot and few-shot prompt techniques, which provide initial steps towards improving hate speech detection. However, future research could significantly benefit from advancing to reasoning-based prompts. These prompts would enhance the model's understanding and contextual reasoning abilities, thereby improving its accuracy in identifying hate speech in various linguistic contexts. The research did not incorporate Explainable AI techniques such as LIME (Local Interpretable Model-agnostic Explanations) or SHAP (SHapley Additive exPlanations). These methods are crucial for providing insights into why the model makes specific predictions. By integrating LIME and SHAP, researchers can enhance the interpretability and transparency of hate speech detection systems. This step is essential for building trust in AI-driven solutions and understanding the decision-making processes behind hate speech identification. During the study, instances of hallucination were observed in the model outputs. Hallucination refers to the generation of erroneous or misleading content by the model. This issue underscores the importance of robust techniques, such as advanced prompts or sophisticated data augmentation strategies, to mitigate such occurrences. Improving the reliability of hate speech detection models is critical for their ethical use in real-world applications. In this research paper, we did not explore any multimodal (combining image and text pair) data from social media posts for Bangla hate speech detection, but multimodal data are important for gaining a better understanding.

8. Future Research Directions

In our research, we will integrate Large Language Models such as Claude 3 and GPT-4, specifically tailored for detecting hate speech in Bengali. These advanced models will significantly enhance our ability to comprehend and categorize hate speech across various dimensions such as ethnicity, religion, gender, and more. By focusing on multilabel classification, we aim to achieve a comprehensive and nuanced understanding of hate speech expressions in Bengali, leveraging the capabilities of GPT-4 and Claude 3 to identify diverse forms and contextual variations effectively. This approach will promise superior performance in hate speech detection and support the development of robust solutions that align with the complex socio-cultural dynamics of Bengali-speaking communities. In our future work, we will prioritize integrating Explainable AI techniques like LIME and SHAP. These methods are essential for providing transparency into the complex decision-making processes of hate speech detection models, which often operate as "black boxes" due to their intricate algorithms and abstract nature. LIME generates detailed, local explanations for individual predictions, revealing which features are most influential in identifying hate speech across diverse contexts, including multilabel classifications. SHAP complements this by using game theory to assign credits to each feature's contribution, offering a global view of feature importance and validating model decisions. By leveraging these techniques, our aim is to enhance both the effectiveness and interpretability of hate speech detection models in Bengali, ensuring they not only perform robustly but also instill trust and facilitate ethical deployment in real-world applications. In Bangladesh, the Bengali dialect is the most widely spoken variation of the Bengali language, encompassing regions such as Khulna, Barisal, Dhaka, Mymensingh, Sylhet, and Chittagong. We plan to develop specialized datasets focusing on Bengali. These datasets will capture diverse expressions and nuanced contextual aspects of hate speech specific to each region. This initiative is crucial for enhancing the accuracy and relevance of our multilabel hate speech detection models, ensuring they effectively address the unique linguistic and cultural dynamics across different parts of Bangladesh. In the future, we will focus on developing methods for detecting and annotating hate speech in Banglish. This will involve addressing the unique challenges posed by its mixed linguistic nature and cultural context, aiming to enhance our understanding and capability in this domain. Our research agenda includes exploring multimodal techniques combining textual and visual information for enhanced multilabel hate speech detection. Integrating large vision models like Claude 3.5 Sonnet with advanced

text-based models such as GPT-4 will provide a holistic approach to identifying and categorizing hate speech across various media types and contexts. In the future, we will employ Chain-of-Thought (CoT) prompting to enhance the reasoning capabilities of LLMs for Bangla hate speech detection. This approach will involve breaking down complex tasks related to hate speech detection into smaller, manageable steps. By guiding the LLM through a logical sequence of intermediate reasoning, CoT prompting aims to improve the accuracy and transparency of the model's responses.

9. Conclusions

This study has addressed the critical need for enhanced hate speech detection methods in the Bengali language, a domain significantly underexplored in current research. By leveraging LLMs such as GPT-3.5 Turbo and Gemini 1.5 Pro, our approach demonstrated substantial improvements over traditional machine learning techniques. The use of Zero-Shot and Few-Shot Learning approaches proved particularly effective, enabling accurate detection with minimal reliance on extensive labeled datasets. We experimented with three different datasets for Bangla hate speech and applied different prompts for each dataset, tailoring our approach for both GPT-3.5 Turbo and Gemini 1.5 Pro. The in-depth analysis of prompting strategies provided valuable insights into optimizing LLMs for hate speech detection tasks. Our experimental results show that LLMs outperformed traditional methods, achieving greater accuracy in understanding and classifying nuanced hate speech. Specifically, Few-Shot Learning with GPT-3.5 Turbo yielded better results than Gemini 1.5 Pro, while Zero-Shot Learning performed better with Gemini 1.5 Pro than GPT-3.5 Turbo. Additionally, we encountered hallucination issues in the Zero-Shot Learning approach, which affected the reliability of the results. By showcasing the potential of LLMs in low-resource languages, the research advances the field of natural language processing. For real-time hate speech detection in Bengali-speaking communities, our models can provide a scalable and reliable solution by capturing contextual nuances efficiently and minimizing reliance on large annotated datasets. The study concludes that LLMs can greatly improve hate speech detection with appropriate prompting techniques, resulting in safer and more accepting online communities for Bengali users. The impact of this research could be increased by developing these models further and investigating how they could be applied to additional low-resource languages in future work.

Author Contributions: F.T.J.F., L.H.B. and S.K. conceived and designed the methodology and experiments; F.T.J.F. performed the experiments; L.H.B. analyzed the results; L.H.B. and S.K. analyzed the data; F.T.J.F. wrote the manuscript. L.H.B. and S.K. reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Science and ICT under Grant NRF-2022R1A2C1005316.

Data Availability Statement: No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Das, A.K.; Al Asif, A.; Paul, A.; Hossain, M.N. Bangla hate speech detection on social media using attention-based recurrent neural network. *J. Intell. Syst.* **2021**, *30*, 578–591. [CrossRef]
2. Jahan, M.S.; Haque, M.; Arhab, N.; Oussalah, M. BanglaHateBERT: BERT for Abusive Language Detection in Bengali. In Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis, Marseille, France, 24 June 2022; European Language Resources Association: Paris, France, 2022; pp. 8–15.
3. Jobair, M.; Das, D.; Islam, N.B.; Dhar, M. *Bengali Hate Speech Detection with BERT and Deep Learning Models*; Springer Nature Singapore: Singapore, 2023. [CrossRef]
4. Romim, N.; Ahmed, M.; Islam, M.S.; Sharma, A.S.; Talukder, H.; Amin, M.R. BD-SHS: A Benchmark Dataset for Learning to Detect Online Bangla Hate Speech in Different Social Contexts. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, Marseille, France, 20–25 June 2022; European Language Resources Associatio: Paris, France, 2022; pp. 5153–5162.

5. Das, M.; Banerjee, S.; Saha, P.; Mukherjee, A. Hate Speech and Offensive Language Detection in Bengali. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing, Online, 20–23 November 2022; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; Volume 1, pp. 286–296.
6. Kumarage, T.; Bhattacharjee, A.; Garl, J. Harnessing artificial intelligence to combat online hate: Exploring the challenges and opportunities of large language models in hate speech detection. *arXiv* **2024**, arXiv:2403.08035.
7. Guo, K.; Hu, A.; Mu, J.; Shi, Z.; Zhao, Z.; Vishwamitra, N.; Hu, H. An investigation of large language models for real-world hate speech detection. In Proceedings of the 2023 International Conference on Machine Learning and Applications (ICMLA), Jacksonville, FL, USA, 15–17 December 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1568–1573.
8. Roy, S.; Harshvardhan, A.; Mukherjee, A.; Saha, P. Probing LLMs for hate speech detection: Strengths and vulnerabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*; Association for Computational Linguistics: Singapore, 2023; pp. 6116–6128.
9. Plaza-del-arco, F.M.; Nozza, D.; Hovy, D. Respectful or Toxic? Using Zero-Shot Learning with Language Models to Detect Hate Speech. In Proceedings of the 7th Workshop on Online Abuse and Harms (WOAH) , Toronto, ON, Canada, 13 July 2023; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 60–68 .
10. Kabir, M.; Islam, M.S.; Laskar, M.T.R.; Nayeem, M.T.; Bari, M.S.; Hoque, E. Benllmeval: A comprehensive evaluation into the potentials and pitfalls of large language models on bengali nlp. *arXiv* **2023** arXiv:2309.13173.
11. Rohanian, O.; Nouriborji, M.; Kouchaki, S.; Nooralahzadeh, F.; Clifton, L.; Clifton, D.A. Exploring the Effectiveness of Instruction Tuning in Biomedical Language Processing. *arXiv* **2023**, arXiv:2401.00579. [CrossRef] [PubMed]
12. Sarker, M.; Hossain, M.F.; Liza, F.R.; Sakib, S.N.; Al Farooq, A. A Machine Learning Approach to Classify Anti-social Bengali Comments on Social Media. In Proceedings of the 2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), Gazipur, Bengal, 24–26 February 2022; pp. 1–6. [CrossRef]
13. Sultana, S.; Redoy, M.O.F.; Al Nahian, J.; Masum, A.K.M.; Abujar, S. Detection of Abusive Bengali Comments for Mixed Social Media Data Using Machine Learning. *preprint* 2022. [CrossRef]
14. Junaid, M.I.H.; Hossain, F.; Rahman, R.M. Bangla Hate Speech Detection in Videos Using Machine Learning. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021. [CrossRef]
15. Karim, M.R.; Chakravarthi, B.R.; McCrae, J.P.; Cochez, M. Classification benchmarks for under-resourced bengali language based on multichannel convolutional-lstm network. In Proceedings of the 2020 IEEE 7th international conference on Data Science and Advanced Analytics (DSAA), Sydney, Australia, 6–9 October 2020; IEEE: Piscataway, NJ, USA, 2020.
16. Ishmam, A.M.; Sharmin, S. Hateful Speech Detection in Public Facebook Pages for the Bengali Language. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 555–560. [CrossRef]
17. Karim, M.R.; Dey, S.K.; Islam, T.; Sarker, S.; Menon, M.H.; Hossain, K.; Decker, S. DeepHateExplainer: Explainable Hate Speech Detection in Under-resourced Bengali Language. In Proceedings of the 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), Porto, Portugal, 6–9 October 2020.
18. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
19. Sarker, S. BanglaBERT: Bengali Mask Language Model for Bengali Language Understanding. 2020. Available online: https://github.com/sagorbrur/bangla-bert (accessed on 16 June 2024).
20. Alexis, C. Unsupervised cross-lingual representation learning at scale. *arXiv* **2019**, arXiv:1911.02116.
21. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
22. Bhattacharjee, A.; Hasan, T.; Ahmad, W.U.; Samin, K.; Islam, M.S.; Iqbal, A.; Rahman, M.S.; Shahriyar, R. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. *arXiv* **2021**, arXiv:2101.00204.
23. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
24. Pahwa, B.; Pahwa, B. BpHigh at SemEval-2023 Task 7: Can Fine-tuned Cross-encoders Outperform GPT-3.5 in NLI Tasks on Clinical Trial Data? In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), Toronto, ON, Canada, 13–14 July 2023; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 1936–1944.
25. Ye, J.; Chen, X.; Xu, N.; Zu, C.; Shao, Z.; Liu, S. ; Huang, X. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv* **2023**, arXiv:2303.10420.
26. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In *AI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4304, pp. 1015–1021. [CrossRef]
27. Goutte, C.; Gaussier, E. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In *Advances in Information Retrieval. ECIR 2005. Lecture Notes in Computer Science*; Losada, D.E., Fernández-Luna, J.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3408. [CrossRef]

28. Mubarak, H.; Al-Khalifa, H.; Alkhalefah, K.S. Halwasa: Quantify and Analyze Hallucinations in Large Language Models: Arabic as a Case Study. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italy, 20–25 May 2024; pp. 8008–8015.
29. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Liu, T. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv* **2023**, arXiv:2311.05232. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.