Detecting Hate Speech in Bangla: A Hybrid Model Using Machine Learning and Lexicon-Based Strategies

Vinayak Vijay[1,†], Avishikta Bhattacharjee[2,†], Kirti Kumari[1,*,†] and Upkar Kumar Kedia[1,†]

[1]Indian Institute of Information Technology Ranchi
[2]Kalinga Institute of Industrial Technology Bhubaneswar

Abstract

The proliferation of online content in regional and code-mixed languages has led to a significant increase in abusive and hate speech, necessitating the development of robust detection systems. This paper presents a comprehensive study on hate speech detection in Hinglish (a mix of Hindi and English) and Bangla language, focusing on the unique challenges these languages pose due to code-mixing, transliteration challenges, and rich morphological variations. Our approach includes pre-processing pipelines tailored to handle codes-mixing data and transliteration challenges. We employ techniques such as TF-IDF word embeddings and a lexicon-based hierarchical approach to capture the nuances of hate speech in these languages. The lexicon-based approach allows us to effectively identify hate speech terms and their variations, even in the presence of morphological variations and code-mixing. The models were trained and evaluated on curated datasets, showcasing their effectiveness in identifying hate speech with high precision.

Keywords

Hate Speech Detection, Deep learning model, Hybrid lexicon-based model, Tf-idf word embeddings

1. Introduction

The rapid proliferation of digital communication platforms has fundamentally transformed human interaction, fostering the widespread use of diverse languages and dialects online. While this linguistic diversity enriches global discourse, it also poses significant challenges—particularly in the detection and mitigation of hate speech. Such harmful content threatens social harmony and can incite real-world violence, making its identification a critical area of concern. In linguistically diverse regions like South Asia, where hybrid languages such as Hinglish (a blend of Hindi and English) and Bangla are widely spoken, hate speech detection becomes even more complex due to the nuances of code-mixing, transliteration, and cultural context.

This research addresses these challenges by developing a specialized hate speech detection framework tailored to the linguistic and cultural intricacies of Hinglish and Bangla. Leveraging advanced Natural Language Processing (NLP) techniques and machine learning algorithms, we analyze a rich corpus of user-generated content collected from various social media platforms. Our model achieves a macro F1-score of 72 for general Hate/Offensive content detection and 45 for the specific detection of Hate Speech, underscoring both the effectiveness and the difficulty of the task in these multilingual settings.

The results reveal distinct patterns and expressions of hate speech across the two languages, highlighting the necessity of language-specific modeling for accurate classification. Beyond the technical contributions, this study provides valuable insights into hate speech dynamics in code-mixed and underrepresented languages, offering a scalable framework for the development of robust content moderation tools. These tools are vital for maintaining respectful discourse online and fostering safer, more inclusive digital communities.

Forum for Information Retrieval Evaluation, December 12-15, 2024, India
*Corresponding author.
†All the authors contributed equally.
vinayakvijay2003@gmail.com (V. Vijay); avishikta.bhattacharjee@gmail.com (A. Bhattacharjee); kirti@iiitranchi.ac.in (K. Kumari); upkar.2023dr101@iiitranchi.ac.in (U. K. Kedia)
https://github.com/Vinayak164000 (V. Vijay)

The remaining sections of this work are organized as follows. Section 2 offers a quick introduction to relevant literature. Section 3 outlines the proposed approach and structure for dealing with Bangla text. Section 4 summarizes the experiments and findings. The paper concludes in Section 5.

2. Related Work

Hate speech (HS) is a form of expression that disseminates negativity, often inciting violence or discrimination based on innate characteristics such as race, ethnicity, or gender. Identifying and addressing HS, especially online, has become a critical issue, particularly in regional languages like Bengali and code-mixed languages such as Hinglish. One of the main hurdles in Bengali HS detection is the lack of labeled datasets, which makes model training difficult. Studies such as the one by Mithun et al. [1] introduced the HS-BAN dataset, a benchmark collection of social media comments for hate speech detection in Bangla. This dataset fills a critical gap by providing annotated data for training and testing machine learning models for hate speech detection in the Bengali language. In the survey on hate speech detection in Bengali, Abdullah et al. [2] highlighted the challenges of limited data availability and the complexity of capturing cultural context in automated detection systems. These challenges are compounded in code-mixed languages, where interleaving between different linguistic systems adds further complexity. For instance, Hossain et al. [3] developed machine learning models to detect hate speech in videos, combining neural networks and traditional algorithms, achieving promising results.Kumari et al. [4]proposed a deep learning approach based on pre-trained BERT models to identify hate speech and offensive language in code-mixed Hindi-English social media text. Their work demonstrated the effectiveness of fine-tuning BERT models for this task.

In recent work, Ahammed et al. [5] explored the application of machine learning techniques for identifying hate speech in Bangla, demonstrating the efficacy of Support Vector Machines (SVM) combined with Term Frequency-Inverse Document Frequency (TF-IDF) features in curbing online hate speech in regional languages. Similarly, Barman et al. [6] emphasized the difficulties posed by recognizing code-mixed languages on social media, noting that code-switching presents unique challenges for Natural Language Processing (NLP) models, particularly when languages are interleaved in unpredictable ways. A lexicon-based approach introduced by Gitari et al. [7] for hate speech detection demonstrated the effectiveness of such models but underscored the need for more sophisticated techniques in future research, as lexicon-based methods might struggle with the nuances of HS in different contexts.

Furthermore, Islam et al. [8] examined various NLP and machine learning methods for detecting hate speech in Bangla social media texts, finding that advanced models such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) significantly improved detection accuracy.Their findings suggest that while basic machine learning models provide a strong foundation, deep learning techniques offer greater potential for handling complex language patterns and nuances in hate speech, particularly in under-resourced languages and dialects.

3. Methodology

This section consists of an overview of data visualization, data preprocessing, feature extraction techniques, and the methods used to train models.

3.1. Data Preprocessing and Data Visualization:

Data preprocessing and visualization are vital steps in natural language processing (NLP). Cleansing the dataset is crucial as it helps in preparing raw data for model training. This initial step ensures that any further analysis is based on accurate and well-organized information. Pre-processing is essential for addressing irregularities and uncertainties in raw textual data, ultimately leading to more reliable and insightful results. The dataset distribution of different classes of comments class for Training and

Developments are presented in Table 1. The detailed explanation about tasks and datasets are seen in articles [9] [10].

Table 1
Dataset Distribution

Comments	Train Dataset	Development Dataset
Hate/Offensive	1954	427
Not Offensive	2046	573
Individually Targeted	957	236
Group Targeted	806	148
Untargeted	192	43

To analyze the dataset, we developed a thorough pre-processing pipeline. This step was essential to convert unstructured social media text into a suitable format for machine learning and deep learning models. The preprocessing steps shown in the Figure 1.

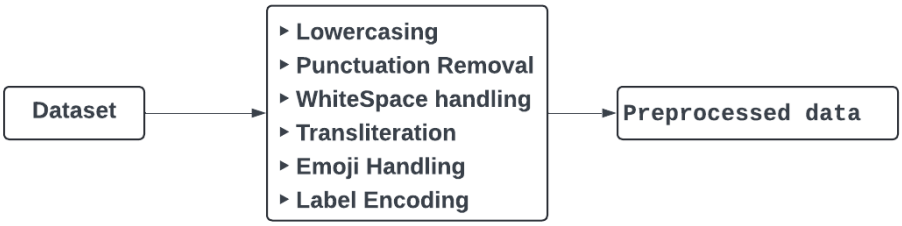

Figure 1: Data Preprocessing Pipeline

3.2. Feature Engineering

Transforming text data into a numerical format is a critical step in building effective machine learning (ML) models, as these models are incapable of interpreting raw characters or words. To this end, we employed TF-IDF (Term Frequency–Inverse Document Frequency) vectorization to extract both word n-gram features (ranging from unigrams to trigrams) and character n-gram features (ranging from bigrams to 5-grams) from the preprocessed text. To reduce noise and improve efficiency, we restricted the extraction to only the most frequent features in both categories, as suggested by Kumari et al. [11].

The selected frequent features were then stacked to form a composite feature vector, effectively capturing both word-level and subword-level textual patterns. By focusing on high-frequency n-grams, we not only reduced the dimensionality of the feature space but also significantly decreased the training time of the classifiers. Furthermore, this approach mitigates the risk of overfitting, thereby enhancing the generalization performance of the models.

3.3. Model Construction

For the hate speech detection in bangla language, we implemented a hierarchical classification approach to manage the complexity of the multilabel multiclass problem. This approach includes a hybrid model that combined lexicon-based analysis and machine learning (ML) algorithms. The Model Construction Pipeline For Label 1 in Hierarchical Classification shown in the Figure **??**.

Our dataset contains many misspelled Bengali words, which TF-IDF struggled to handle effectively. To address this, a lexicon-based Hierarchical approach was implemented. A manually curated dictionary, encompassing both correct and incorrect spellings of Bengali words, was created. By augmenting the training data with these dictionary entries, the model was exposed to a wider range of linguistic variations. This enabled the model to learn the patterns of misspellings and their correct counterparts.

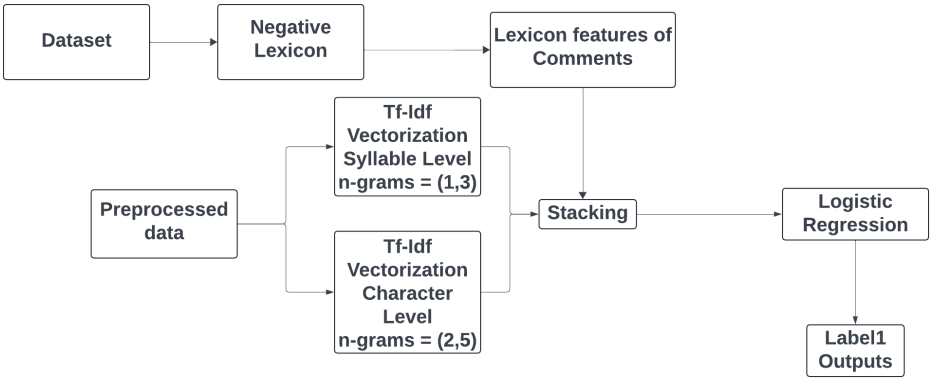

Figure 2: Model Construction Pipeline For Label 1 in Hierarchical Classification

This enhancement can be attributed to the richer feature representation, improved generalization capabilities, and increased robustness to noise and errors in the input data.

Logistic Regression was used in tandem with the lexicon to classify the data by examining both the presence of hate-related terms and features extracted from the text, such as term frequencies and patterns indicative of hate speech. Once the initial label (hate speech or not) was predicted, we had to prepare the input data for the label 2 prediction. To achieve this, the existing input data, represented using TF-IDF features, was combined with the "offensive gold label" into a single matrix. This augmented dataset served as the input for the second prediction task. This enhanced feature set was then used as input for the second stage of classification. The detail steps for model construction for Label 2 is shown in Figure 3.

In the second stage, we employed a Soft Voting Classifier using the Scikit-learn library. The Soft Voting Classifier was composed of three base classifiers: Logistic Regression, Decision Tree Classifier, and Extra Tree Classifier. The objective in this phase was to predict the second label, which involved determining whether the hate speech was targeted at an individual, a group. By using soft voting, we aimed to aggregate the predictions from each individual classifier, where the final prediction was a weighted combination of the probabilities from each model. This ensemble method allowed for more robust decision-making, as it leveraged the strengths of multiple algorithms to improve classification accuracy.

After training the model on the labeled training set, we evaluated its performance on a development (dev) dataset using the macro F1 score.

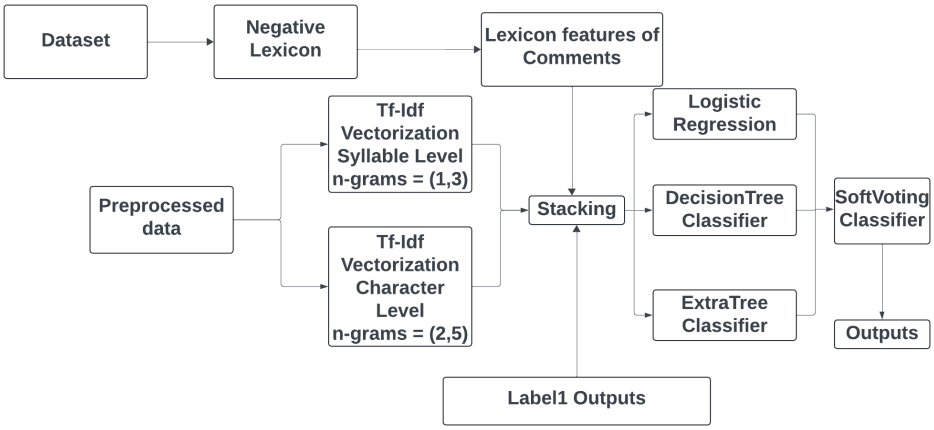

Figure 3: Model Construction Pipeline For Label 2 Classification

4. Experiments and Results

The dataset provided by the HASOC 2024 shared task organizers contains train, development, and test set which includes a mixed Bangla code text which must be classified for Hate\Not Hate for the first label classification and the individual target categories for the second label classification, which was multiclass classification, making the overall problem as multilabel and multiclass classification. We experimented with two approaches:(i) Using Scikit-Learn Chain Classifier and (ii) Hierarchal Approach

Chain Classification: In our approach, we employed a chain classification with different machine learning classifiers to tackle this multilabel and multiclass problem. Chain classification involves predicting the first label (Hate/Not Hate) and then using that prediction to inform the second label (target type). By combining these predictions, the models can effectively capture the hierarchical relationships between the labels. Table 2 presents the performance results for various ML classifiers using chain classification on the development set. These results, computed using the Scikit-learn library, demonstrate the models' effectiveness in predicting both labels and give insight into which classifiers perform best based on their macro F1-scores.

Table 2
Results of Chain Classification

Classification Model	F1 Score (label 1)	F1 Score (Label 2)
Logistic Regression	70	41
MultinomialNB	65	38
DecisionTreeClassifier	55	31
RandomForest	69	42
ExtraTreeClassifier	66	42
AdaBoostClassifier	65	21
Soft Voting Classifier(LR, RF, ET)	69	42

In our analysis, we found that the best F1 score of 0.70 for the first label (Hate/Not Hate classification) was achieved using Logistic Regression. However, we sought to further improve the classification of label 2 accuracy by integrating a lexicon-based hierarchical analysis with Logistic Regression. This combination proved effective, increasing the accuracy from 0.70 to 0.72. The integration process involved applying sentiment analysis techniques, incorporating contextual word embeddings to capture nuanced meanings in the text, and leveraging sentiment scores derived from a lexicon specifically designed for hate speech detection. These combined features allowed the model to better understand the context and emotional tone of the code-mixed Bangla text, improving its ability to detect hate speech more accurately.

Additionally, the predictions obtained from the first label classification were used to enhance the feature set for the second label classification. The second label aimed to identify whether the hate speech was directed towards an individual, a group, or if it was untargeted. By incorporating the first label's predictions, we created a more comprehensive and enriched feature set, which significantly boosted the model's performance for this second label. This approach not only enhanced the overall classification accuracy but also provided deeper insights into the linguistic patterns associated with hate speech, contributing to a better understanding of the types of targets in hate speech scenarios.

Table 3 showcases the results for the 2nd label prediction using various ML classifiers on the enhanced feature set, highlighting the achieved F1 scores on validation dataset and rank we got on test dataset provided by the HASOC organizers.

5. Conclusion

In our paper, we outline the comprehensive strategy devised by Team *AI_ML_IIITRANCHI* for the HASOC 2024 shared task. Our approach involves meticulously selecting the top frequent character and word n-grams from the texts, then consolidating and transforming them into TF-IDF vectors to train

Table 3
Results of Hierarchical Classification

Classification Model	F1 Score (Label 1)	F1 Score (Label 2)	Rank
Logistic Regression	72	41	-
MultinomialNB	69	40	-
DecisionTreeClassifier	59	43	-
RandomForest	71	44	-
ExtraTreeClassifier	62	44	-
AdaBoostClassifier	65	22	-
Soft Voting Classifier(LR, RF, ET)	71	45	3 (for 1st Label) and 4(for 2nd Label)

the ML classifiers, which are complemented by a Lexicon-based strategy.

Notably, Team *AI_ML_IIITRANCHI* actively participated in Task 2 and demonstrated impressive performance by securing 3rd place for 1st label prediction and 4th place for 2nd label prediction. Our proposed strategy surpassed most models submitted by other participants in the shared task, positioning our team as one of the top performers. Furthermore, our work serves as an example of the efficacy of feature reduction algorithms, even those that are relatively simple, in classification tasks. Moving forward, our goal is to investigate statistical feature selection algorithms and diverse feature sets to further enhance the performance of ML classifiers.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. Das, S. Banerjee, P. Saha, A. Mukherjee, Hate speech and offensive language detection in bengali, arXiv preprint arXiv:2210.03479 (2022). URL: https://arxiv.org/abs/2210.03479.

[2] A. A. Maruf, A. J. Abidin, M. M. Haque, Z. M. Jiyad, A. Golder, R. A. Z. Aung, Hate speech detection in the bengali language: a comprehensive survey, Journal of Big Data 11 (2024) 53. doi:10.1186/s40537-024-00956-z.

[3] M. I. H. Junaid, F. Hossain, R. M. Rahman, Bangla hate speech detection in videos using machine learning, in: 2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), 2021. doi:10.1109/UEMCON53757.2021.9666550.

[4] K. Kumari, J. P. Singh, Ai ml nit patna at hasoc 2019: Deep learning approach for identification of abusive content., FIRE (working notes) 2517 (2019) 328–335.

[5] S. Ahammed, M. Rahman, M. H. Niloy, S. M. M. H. Chowdhury, Implementation of machine learning to detect hate speech in bangla language, in: 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), 2019. doi:10.1109/SMART46866.2019.9117214.

[6] U. Barman, A. Das, J. F. Joachim Wagner, Code mixing: A challenge for language identification in the language of social media, in: Proceedings of the First Workshop on Computational Approaches to Code Switching, 2014. doi:10.13140/2.1.3385.6967.

[7] N. D. Gitari1, Z. Zuping, H. Damien, J. Long, A lexicon-based approach for hate speech detection, International journal of multimedia and ubiquitous engineering 10 (2015) 215–230. doi:10.14257/ijmue.2015.10.4.21.

[8] M. S. Islam, S. Saha, M. M. Alam, N. K. Datta, M. H. Ali, M. D. Hossain, M. G. Moazzam, Natural language processing and machine learning approaches to detect bangla hate speech on social

media, in: 2023 26th International Conference on Computer and Information Technology (ICCIT), IEEE, 2023, pp. 1–6. doi:10.1109/ICCIT60459.2023.10441452.

[9] K. Ghosh, N. Raihan, S. Modha, S. Satapara, T. Gaur, Y. Dave, M. Zampieri, S. Jaki, T. Mandl, Overview of the HASOC Track at FIRE 2024: Hate-Speech Identification in English and Bengali, in: FIRE '24: Proceedings of the 16th Annual Meeting of the Forum for Information Retrieval Evaluation. December 12-15, Gandhinagar, India, Association for Computing Machinery (ACM), New York, NY, USA, 2024.

[10] N. Raihan, K. Ghosh, S. Modha, S. Satapara, T. Gaur, Y. Dave, M. Zampieri, S. Jaki, T. Mandl, Overview of the HASOC Track at FIRE 2024: Hate-Speech Identification in English and Bengali, in: K. Ghosh, T. Mandl, P. Majumder, D. Ganguly (Eds.), Forum for Information Retrieval Evaluation (Working Notes) (FIRE 2024) December 12-15, Gandhinagar, India, CEUR-WS.org, 2024.

[11] K. Kumari, J. P. Singh, Ai_ml_nit_patna@ hasoc 2020: Bert models for hate speech identification in indo-european languages., in: FIRE (Working Notes), 2020, pp. 319–324.