Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages using Machine Learning Models*

Gunjan Kumar[1,*,†], Jyoti Prakash Singh[1,†]

[1]National Institute of Technology Patna, Bihar-800005, India

Abstract

The social media platform is widespread among users to share information, opinion, and comments. Hate speech harms society, so its detection is crucial. The HASOC (Hate Speech and Offensive Content Identification) develop a multilingual dataset of hate speech. It can be exceedingly difficult to identify hate speech, cyber-aggression, and offensive language in codemix language posted by social media users. This paper presents the HASOC task for Hindi-English datasets. We are intrigued to offer a model to distinguish between hate speech, offensive language, stand-alone hate, and contextual hate because it is essential for online social health. We have experimented with two different feature extraction: character level feature and word level. These experiments have been associated with comments on code-mixed Hindi-English social media text. The combined word-level and character-level features performed better than pre-trained fastText embedding and GloVe embedding for the code-mixed Hindi-English dataset.

Keywords

Hatespeech, Offensive Language, Machine Learning, Deep Learning, Multilingual

1. Introduction

Social media is a prevalent platform to share/gather information, thinking, views, and comment with each other. Social media users have the freedom to share anything (text, audio, video, and images) with their family and friends without paying any cost. Facebook, Instagram, Youtube, WhatsApp, and Twitter are popular social media platforms. During the pandemic, every mobile user started using social media to connect with society without paying any cost. But with the positive aspect of social media, it also has some negative aspects such as Hate-speech, Cyber-bulling, Cyber-aggression and Offensive language. Some social media users take disadvantage of language-based social boundaries. They use offensive and hurtful language in their native language to hurt other people or communities. Sometimes users use hate speech unintentionally and unconsciously, but it can hurt some innocent people. If it is not detected on time, it could damage our social health. So, we need to identify this abusive content from social media platforms before it spreads.

Forum for Information Retrieval Evaluation, December 9-13, 2022, India

*Corresponding author.

†These authors contributed equally.

gunjanp2205@gmail.com (G. Kumar); jps@nitp.ac.in (J. P. Singh)

Hate content on social media deteriorates the health of the targeted people. The victim of hate speech suffers from anxiety, depression, mental health problem and, in some saviour cases, suicidal tendencies. India is a multilingual country where 22 languages are spoken. The translated randomized text of the native language with English as the binding language is termed codemix text. Social media users sometimes use their mother tongue or codemix text to post their comments. Identification of hate speech written in multilingual is challenging for the researcher. It is tough to restrict such an offensive language from the native codemix text messages from the Indo-Aryan language. We proposed a model based on Machine Learning (ML) to identify Hate and Offensive (HOF) and Non-Hate Offensive (NOT) content on multi-domain social media platforms collected from Twitter for task-1. Further, classify the HOF tweets into Standalone-Hate (SHOF) and Contextual-Hate (CHOF) in task-2 [1]. To Validate the model, we use multilingual HASOC Corpus [2, 3]. For the code-mixed Hindi corpus, we normalized Devanagari to Roman script using two different forms of text embeddings and transliteration tools.

The rest of the paper is organized as follows. The related works for identifying hate speech and offensive language are presented in Section 2, and our proposed framework for identifying hate speech and offensive language is presented in Section 3. The results of the suggested scheme are shown in Section 4. In Section 5, we have discussed the result obtained, and the last one is Section 6, in which we have concluded the paper and the future directions for these tasks.

2. Related Work

The most significant social issue on social media is hate speech, and many studies are being done in this area [4, 5, 6, 7, 8]. This section provides an overview of existing methods for automatically detecting hate-speech in multilingual and multi-model. The automated approaches for hate-speech detection could be categorized into ML [9, 10, 11] and deep learning (DL) based methods [12, 13, 7, 14]. The ML models are effective in hate-speech identification. The multi-model is very effective in hate-speech identification [15, 16, 17, 18, 2, 19]. Malmasi et al. [20] classifying hate-speech as frequent profanity on social media posts. They created a lexical baseline for discriminating between hate-speech and profanity using the supervised classification method. Character n-grams, word n-grams, and word skip-grams are all used in feature extraction. While classifying postings in three classes, they achieve an accuracy of 78%. Sindhu et al. [21], done the comparative study, evaluates which feature engineering technique and ML algorithm performs best on a common publically available dataset by comparing several feature engineering strategies and ML algorithms. They found that the support vector machine (SVM) method performed best when combined with bigram features, with an overall accuracy rate of 79%. Kumari et al. [22] work on multilingual (Hindi, English, and Bangla) code-mixed text. They suggested two DL systems: Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN). *One-hot* and *FastText* embeddings serve as the two different inputs in text representations. They found that LSTM performs better with *FastText* embedding for Hindi and Bangla datasets, and CNN works better for English datasets. Kumari et al. [23] focus on finding the aggression level of the comment posted on social media. They classified the aggressive comments into three additional classes, whether a statement is non-aggressive,

Covertly aggressive, or Overtly aggressive. They applied LSTM, CNN with *FastText*, and *One-hot* embeddings for text representations to categorize the comments into three groups. LSTM with *FastText* embedding improves model performance for the Hindi and Bangla datasets. For the English dataset, CNN with *FastText* embedding performs better. Social media content analysis and hate speech identification have greatly benefited from using ML algorithms. Subtask A is a binary classification task used to classify social media posts into hate or non-hate. Subtask-B categorizes the results of Subtask-A into three groups: profane, offensive, and hateful. Therefore, to assess their performance on a publically available dataset with two separate classes, this research evaluates the performance of three feature engineering techniques and four ML algorithms. We build a baseline model utilizing these two categories of hatred and non-hate and then use several optimization strategies to raise model performance ratings.

3. Methodology

This section describes the details of the dataset given for training and methodology. The framework of the proposed modal is shown in Figure 1. The Twitter data is preprocessed then the feature is extracted and passed through the ML models. The ML classifier classifies the tweets into two classes (NOT, HOF) in task-1 and three classes (NOT, SHOF and CHOF) in task-2, shown in the Figure 1 Flow diagram of the proposed model.

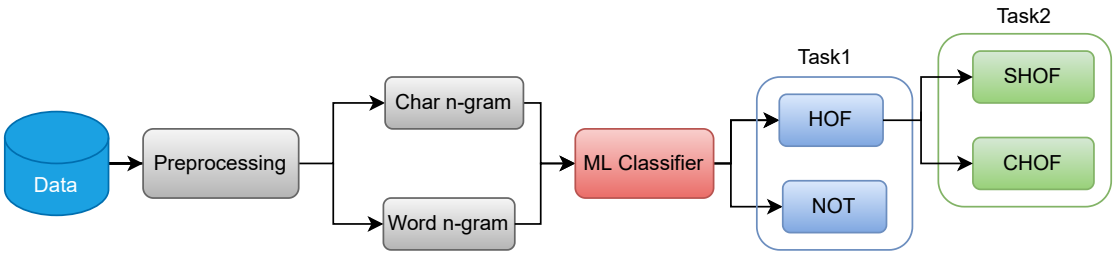

Figure 1: Flow diagram of proposed model

3.1. Dataset Discription

The Hindi-English dataset of HASOC 2022 for task-1 and task-2 contains the text content of tweets in Hindi and English language, tweet-ids, and the labels for task-1 and task-2, respectively. The training dataset's statistics are presented in Table 1. Moreover, the test dataset contains 1281 tweets which should be categorized into one of the classes based on the task. Task-1 is a binary classification task to classify the tweets into two classes, NOT (Non-Hate Offensive) and HOF (Hate and Offensive). Task-2 is a multi-class classification on the same dataset, ICHCL (Identification of Conversational Hate speech in Codemixed Language), classifying the tweets into three classes SHOF, CHOF, and None (Non-Hate). The data would include links, usernames, emojis, and hashtags that refer to a Twitter user. Table 2 presents a dataset sample in different categories. We have a total of 4914 data samples, of which 2390 are from the NOT class, and

Table 1
Detail Description of Dataset

Task	# of classes	# of tweets
Task-1	NOT	2390
	HOF	2524
Task-2	NONE	2390
	SHOF	1636
	CHOF	888

Table 2
Sample of the dataset for task1

Tweets	Label
This comes just days after a terrorist held Jewish congregants hostage at a synagogue	NOT
What an anti-Semitic piece of scum, I am calling for a full forensic audit of Lauren Boebert's GED exam.	NOT
She's the one who desecrates the memory of Holocaust victims	HOF
Bigotry has nothing to do with education. In fact if you are educated and a bigot, it's a deadly cocktail	HOF

2524 are from the HOF class for task-1. For task-2, the HOF is further classified into two classes, SHOF and CHOF.

3.2. Preprocessing

In pre-processing the data, we used the same approach as the previous year's experiments conducted for the competition [14]. Pre-processing of data mainly includes removing the phrase' username,' emojis, and hyperlinks and replacing multiple white spaces with a single white space. We convert all the upper case into lower case alphabet and do not remove the stopwords because, sometimes, it changes the meaning of the sentences. Task-2 data is a contextual dataset, so we merge those with identical ids, and their corresponding labels are also similar. These steps are applied to the train and test datasets to facilitate the training process.

3.3. Machine learning models used for classification

We developed a model, which is shown in Figure 1, to categorize tweets into two groups, NOT and HOF; we used seven ML classifiers. The classifiers are (i) SVM, (ii) Logistic Regression (LR), (iii) K-Nearest Neighbor (KNN), (vi) Random Forest (RF), (v) Decision Tree, (vi) Gradient Boosting (GB), and (vii) AdaBoost (AB) classifier. We extracted the word and character level features To train the ML classifier. The *tf-idf* vectorizer with word n-gram (1-3) is used to get the word level features and takes a maximum of 10000 features. The second is the character level feature which is again extracted using *tf-idf* vectorizer with character n-gram (1-6) and taken at a maximum of 10000. To train the ML models, we use the training data. We have taken 80% of the data to train the ML classifier, and the rest of the 20% of data was used for validating

Table 3
Hyper-parameter settings for the CNN.

Hyper-Parameter	Value
Maximum sequence length	30
Maximum No. of words	10000
No. of filter	512
Filter size	2,3,4
Activation Function	ReLU
Max-pooling window	5
Dropout Rate	0.2
Epochs	100
Batch size	32
Loss	Categorical cross-entropy
Optimizer	Adam

the model. After training the model with individual features, we found that the model was not performing better with word n-gram, whereas character n-gram is slightly better. Then, we concatenated both the individual features (character n-gram and word n-gram), again trained the model, and found that it performed better than the individual one. The experimental results of these classifiers on the Hindi-English datasets are shown in Table 4, respectively.

3.4. Deep learning models used for classification

We used CNN to classify tweets into their output classes to improve the proposed model's classification performance. We implemented 1D-CNN to Classify the tweets into two or three classes. Created embedding for the input layer of 1D-CNN using a pre-trained GloVe (GloVe 42B.300d) model [24]. We concatenate the output feature of the three 1D-CNN layers, followed by max-pooling of size 5 with the activation function ReLU. The applied 1D-CNN has 512 no. of filters with filter sizes 2,3,4. The concatenated CNN featured is passed into one more 1D-CNN layer with no. of filter 256, and the filter size is 1 to extract the vital features. The activation function used in this layer is ReLU having a max-pooling of size 5. Then flatten the layer, pass it from the dense layer, and classify the tweets into binary or ternary classes. The used hypermeters are mentioned in the table 3. CNN performs poorly compared to a simple ML classifier; their performance is explained in the section 4.

4. Result

This section details the outcomes for all two activities in the Hindi- English dataset. We have used two feature extraction characters, n-gram and word n-gram, for the ML classifier for both tasks. Classification models have been validated for each task using precision, recall and Acc. We randomly split the training dataset into training and validation sets in an 80:20 ratio, and the obtained results are shown in Tabel 4.

First, we implemented all seven ML classifiers with the word n-gram; we again classified the tweets using the character n-gram and found the Acc. is improved by 2%. Then we applied

Table 4
Comparison analysis of the proposed model with different ML and DL models

Ml classifier	Class(Task1)	Precision	Recall	Acc.	Class(Task2)	Precision	Recall	Acc.
	NOT	0.71	0.71	0.71	NOT	0.65	0.79	0.62
SVM	HOF	0.71	0.71		SHOF	0.53	0.42	
					CHOF	0.70	0.56	
	NOT	0.73	0.68	0.71	NOT	0.68	0.75	0.62
RF	HOF	0.70	0.75		SHOF	0.49	0.44	
					CHOF	0.65	0.59	
	NOT	0.73	0.69	**0.72**	NOT	0.66	0.81	**0.64**
LR	HOF	0.70	0.75		SHOF	0.57	0.44	
					CHOF	0.68	0.56	
	NOT	0.69	0.70	0.69	NOT	0.63	0.78	0.60
KNN	HOF	0.70	0.68		SHOF	0.55	0.34	
					CHOF	0.56	0.58	
	NOT	0.72	0.71	0.72	NOT	0.63	0.86	0.62
GB	HOF	0.72	0.72		SHOF	0.54	0.33	
					CHOF	0.70	0.49	
	NOT	0.66	0.63	0.65	NOT	0.66	0.66	0.57
DT	HOF	0.65	0.68		SHOF	0.43	0.44	
					CHOF	0.55	0.54	
	NOT	0.69	0.65	0.68	NOT	0.57	0.81	0.54
AB	HOF	0.67	0.72		SHOF	0.39	0.22	
					CHOF	0.59	0.39	
	NOT	0.72	0.65	0.69	NOT	0.69	0.69	0.60
CNN	HOF	0.66	0.73		SHOF	0.47	0.55	
					CHOF	0.65	0.44	

the combined feature of character n-gram and word n-gram; it improved the classification Acc. 4%. The results of the various analyses performed for this report are presented in Table 4. Most models performed well and gave similar results with very slight differences. We found that when concatenating the feature (character and word n-gram) and passing through the ML classifier then, 4% training accuracy was increased. LR achieved the highest Acc., 0.72 and 0.64 for task-1 and task-2, respectively. The performance of outperforming ML model (LR) with different features is shown in Table 5. We used CNN for both tasks to improve the proposed model's accuracy (Acc.), which is explained in Section 3.4. But CNN is performing poorly compared to a simple ML classifier, obtained Acc. 0.69 and 0.60 for task-1 and task-2, respectively. We implemented CNN with two dimensions of pre-trained GloVe embeddings, 300 and 100. GloVe 300 performs better than 100. On the other hand, the *tf-idf* vectorizer performs similarly with both SVM and RF classifiers for task-1 and task-2.

5. Discussion

The proposed tweet classification models have been evaluated for each task using a Macro-F_1 and Macro Precision. The results of suggested methods for testing samples with various

Table 5
Results of best performing model (LR) with differnt feature

Task	Classifier	Feature	Macro F_2	Macro precision
task1	LR	(char + word) n-gram	**57%**	**57%**
	LR	char n-gram	55%	55%
	LR	word n-gram	54%	54%
task2	LR	(char + word) n-gram	**29%**	**28%**
	LR	char n-gram	28%	27%
	LR	word n-gram	27%	26%

Table 6
Position score on HASOC FIRE-2022.

Task	Rank	Team-name	Macro F_2	Macro precision
task1	1	nlplab-isi	**70%**	**71%**
	32	gunjan	57%	57%
	42	nitk-it	32%	24%
task2	1	ub-cs	**49%**	**52%**
	24	gunjan	29%	28%
	25	sakshi hasoc(2022)	21%	49%

embedding combinations used for training and testing are shown in Table 5. CNN's performance is poor than the many ML models (SVM, LR, GB, and RF) because DL models need a vast amount of data to train. Acc achieved by the CNN is 0.69 and 0.60 for task-1 and task-2, respectively, which is lower than LR. LR performs better than all the other implemented ML and DL models for both task-1 and task-2 on training data. The precision and recall for task-1 are 0.73, 0.69, and 0.70, 0.75 for the NOT and HOF classes, respectively. Task-2 is multiclass classification; still, the LR model performs better and achieved precision and recall of 0.66, 0.81, 0.57, 0.44 and 0.68, 0.56 for NOT, SHOF and CHOF, respectively. So we submitted the testing results to the competition held by HASOC on FIRE-2022 on the best-performing LR model. We submitted our results using the team name **gunjan** and held positions 32 and 24 out of 42 and 25 for task-1 and task-2, respectively, shown in Table 6. Our proposed model obtained an approximate *Macro-F1* of 57% and 29%, for task-1 and task-2, respectively, with the combined feature of character n-gram and word n-gram. We achieved 12^{th} and 10^{th} rank among the top participants lists of shared task-1 and task-2, respectively. Task-2 is not performing well as task-1, which shows misclassification instances more in task-2; data imbalance may be a reason behind this which is shown in Table 1. The main reason for the misclassification of task-2 is that it is multiclass with a contextual classification task. Identifying SHOF and CHOF tweets among the HOF tweets is challenging. Only recognising the keywords will result in many false positive cases because context plays a significant role in detecting Hate and Offensive language in the conversational text.

6. conclusion

The tremendous amount of content published on social media platforms makes it hard to manually screen such harmful content, requiring platform providers to turn to automatic methods for identifying hateful and offensive content. Identifying conversational- hate among the hate and offensive language is a very challenging task. Hate speech is a very prominent area of research among researchers, but only in the English language. Very few research works have been done on multilingual and English code-mixed text. This work focuses on identifying Hate and Offensive content on multilingual (English and Hindi) data using ML models. We extracted word-level and character-level features from the text and trained the seven ML models. The LR model has been found to perform better than the others when features were extracted using the combined feature of character n-gram and word n-gram. We participated in the competition held in 2022, and the obtained result is shown in Table 6.

7. Acknowledgments

The first author would want to acknowledge the Ministry of Human Resource Development (MHRD), Government of India for the financial support during the research work through the MHRD Ph.D Scheme for computer science and IT.

References

[1] M. S. Satapara, Shrey, T. Mandl, H. Madhu, P. Majumder, Overview of the HASOC Subtrack at FIRE 2021: Conversational Hate Speech Detection in Code-mixed language , in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[2] J. García-Díaz, C. Caparros-Laiz, R. Valencia-García, Umuteam at semeval-2022 task 5: Combining image and textual embeddings for multi-modal automatic misogyny identification, in: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), 2022, pp. 742–747.

[3] S. Satapara, P. Majumder, T. Mandl, S. Modha, H. Madhu, T. Ranasinghe, M. Zampieri, K. North, D. Premasiri, Overview of the hasoc subtrack at fire 2022: Hate speech and offensive content identification in english and indo-aryan languages, in: FIRE, ????

[4] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the international AAAI conference on web and social media, volume 11, 2017, pp. 512–515.

[5] P. Fortuna, S. Nunes, A survey on automatic detection of hate speech in text, ACM Computing Surveys (CSUR) 51 (2018) 1–30.

[6] A. M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, N. Kourtellis, Large scale crowdsourcing and characterization of twitter abusive behavior, in: Twelfth International AAAI Conference on Web and Social Media, 2018.

[7] K. Kumari, J. P. Singh, Y. K. Dwivedi, N. P. Rana, Aggressive social media post detection

system containing symbolic images, in: Conference on e-Business, e-Services and e-Society, Springer, 2019, pp. 415–424.
[8] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: Proceedings of the fifth international workshop on natural language processing for social media, 2017, pp. 1–10.
[9] M. Anzovino, E. Fersini, P. Rosso, Automatic identification and classification of misogynistic language on twitter, in: International Conference on Applications of Natural Language to Information Systems, Springer, 2018, pp. 57–64.
[10] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, M. Shrivastava, A dataset of hindi-english code-mixed social media text for hate speech detection, in: Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media, 2018, pp. 36–41.
[11] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the hasoc subtrack at fire 2021: Hate speech and offensive content identification in english and indo-aryan languages and conversational hate speech, in: Forum for Information Retrieval Evaluation, 2021, pp. 1–3.
[12] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th international conference on World Wide Web companion, 2017, pp. 759–760.
[13] J. Chen, S. Yan, K.-C. Wong, Verbal aggression detection on twitter comments: convolutional neural network for short-text sentiment analysis, Neural Computing and Applications 32 (2020) 10809–10818.
[14] S. Mohtaj, V. Woloszyn, S. Möller, Tub at hasoc 2020: Character based lstm for hate speech detection in indo-european languages., in: FIRE (Working Notes), 2020, pp. 298–303.
[15] G. Kumar, J. P. Singh, A. Kumar, A deep multi-modal neural network for the identification of hate speech from social media, in: Conference on e-Business, e-Services and e-Society, Springer, 2021, pp. 670–680.
[16] K. Kumari, J. P. Singh, Y. K. Dwivedi, N. P. Rana, Towards cyberbullying-free social media in smart cities: a unified multi-modal approach, Soft computing 24 (2020) 11059–11070.
[17] K. Kumari, J. P. Singh, Identification of cyberbullying on multi-modal social media posts using genetic algorithm, Transactions on Emerging Telecommunications Technologies 32 (2021) e3907.
[18] K. Kumari, J. P. Singh, Y. K. Dwivedi, N. P. Rana, Multi-modal aggression identification using convolutional neural network and binary particle swarm optimization, Future Generation Computer Systems 118 (2021) 187–197.
[19] P. Vijayaraghavan, H. Larochelle, D. Roy, Interpretable multi-modal hate speech detection, arXiv preprint arXiv:2103.01616 (2021).
[20] S. Malmasi, M. Zampieri, Detecting hate speech in social media, arXiv preprint arXiv:1712.06427 (2017).
[21] S. Abro, S. Shaikh, Z. H. Khand, A. Zafar, S. Khan, G. Mujtaba, Automatic hate speech detection using machine learning: A comparative study, International Journal of Advanced Computer Science and Applications 11 (2020).
[22] K. Kumari, J. P. Singh, Ai ml nit patna at hasoc 2019: Deep learning approach for identifi-

cation of abusive content., FIRE (working notes) 2517 (2019) 328–335.
[23] K. Kumari, J. P. Singh, Ai_ml_nit_patna@ trac-2: deep learning approach for multi-lingual aggression identification, in: Proceedings of the second workshop on trolling, aggression and cyberbullying, 2020, pp. 113–119.
[24] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.