Int. j. inf. tecnol. (March 2023) 15(3):1319–1331
https://doi.org/10.1007/s41870-023-01211-z

ORIGINAL RESEARCH

Hate speech recognition in multilingual text: hinglish documents

Arun Kumar Yadav[1] · Mohit Kumar[1] · Abhishek Kumar[1] · Shivani[1] · Kusum[1] · Divakar Yadav[2]

Received: 27 August 2022 / Accepted: 21 February 2023 / Published online: 13 March 2023

Abstract The Internet is a boon for mankind but its misuse has been increasing drastically. Social networking platforms such as Facebook, Twitter and Instagram play a predominant role in expressing views by the users. Sometimes users wield abusive or inflammatory language, that may provoke readers. This paper aims to evaluate various machine learning and deep learning techniques to detect hate speech on various social media platforms in the Hinglish (English-Hindi code-mix) language. In this paper, we apply and evaluate several machine learning and deep learning methods, along with various feature extraction and word-embedding techniques, on a consolidated dataset of 20600 instances, for hate speech detection from tweets and comments in Hinglish. The experimental results reveal that deep learning models perform better than machine learning models in general. Among the deep learning models, the CNN-BiLSTM model with word2vec word embedding provides the best results. The model yields 0.876 accuracy, 0.830 precision, 0.840 recall and 0.835 F1-score. These results surpass the recent state-of-art approaches.

Keywords Hate speech · Deep learning · Machine learning · Word2Vec · CNN · BiLSTM

✉ Divakar Yadav
dsy99@rediffmail.com

[1] Department of Computer Science & Engineering, NIT Hamirpur (HP), Hamirpur, India

[2] Present Address: School of Computer and Information Sciences, Indira Gandhi National Open University, Delhi, India

1 Introduction

In the era of Information Technology (IT), the Internet and social media platforms are not merely a source of entertainment or chatting but also an integral part of people's social life. Children and youth, especially in India, are highly fond of social media platforms such that the annual growth in active social media users in India is 31.2 percentage that is more than 78 million users [1]. With the increasing number of users, misuse of these platforms is also increasing tremendously. One of the biggest problems that we are facing today on the Internet is hate speech. According to Wikipedia [2], Cambridge dictionary defines hate speech as "*public speech that expresses hate or encourages violence towards a person or group based on something such as race, religion, sex, or sexual orientation*". Hate speech may include different forms of expressions that advocate, incite, promote or justify hatred, violence and discrimination against a person or group of persons for a variety of reasons [3]. Many times, users confound free speech with hate speech. Forms of speech that can evolve into hate speech are not limited to spoken word and include any nature of "*attacks [that are] printed, published, pasted up, or posted on the Internet – expressions that become a permanent part of the visible environment in which our lives, and the lives of members of vulnerable minorities, have to be lived*" and which effect them badly [4].

Hate speech can be propagated by sharing messages, images or videos. It can take place on social media, various messaging platforms, mobile phones or gaming platforms. Hate speech can be inflicted face to face or through internet. It can often happen simultaneously but online, it leaves a digital footprint that plays a crucial role in stopping it.

There is a potential of violence and hate crimes due to hateful speech. Its exposure can have profound psychological impacts such as heightened stress, anxiety, depression, and desensitization. Victimization, direct or indirect, has also been associated with increased rates of alcohol and drug use [5].

Tackling such vast data on social media is not easy to do manually. Approximately 6,000 tweets/second are generated on Twitter. There were around 1.386 billion active users of Instagram in 2021, 2.853 billion of Facebook and 1.3 billion on Facebook Messenger [5].In the past, applications of Artificial Intelligence have increased in hate speech detection fake news detection and Text summerization [6–8], especially machine learning and deep learning, may prove as a boon for automatic hate speech detection. After doing tremendous research we go to the conclusion that there is a significant amount of research done on the English language and they are successfully implemented in various twitter bots and chat bots of tech giants. But in India, their performance is poor due to the type of language used by users being Mixed up. A majority of Indian users share their thoughts in Hindi mixed English i.e Hinglish. On the basis of literature review, it is observed that very less research has been carried out using machine learning and deep learning based methods for hate speech detection. However, there is a lot of gap present in improvement of the performance in hate speech detection.

The prime focus of this work is to detect hate speech in Hinglish (Hindi and English mixed) data. Machine learning and deep learning based approaches are applied on a hybrid dataset – consolidated by merging 3 publicly available datasets – and their results are compared on various performance metrics such as accuracy, precision, recall, F1-score, etc. The contributions of the paper may be summarized as follows:

- In the previous paragraphs, we have outlined the lack of significant amount of work specifically with English and Hindi code mix data. One possible cause of such a situation could be the lack of significantly large datasets to train the models. In this work, we first address this gap by employing a consolidated dataset that is created by merging 3 publicly available and relatively smaller datasets, namely **Bohra 2018 dataset**, **Kumar 2018 dataset** and **HASOC 2021 Hindi-English Coded dataset**. There is no obvious issue observed that would prevent such consolidation to be extended to more than 3 datasets.
- We have considered 8 different machine learning models with 4 feature extraction methods and 4 different deep learning models with 3 word-embedding techniques for experimental evaluations. This enabled the authors to attack the problem from various perspectives. The result of all the experiments are discussed in Sect. 4.
- Due to the comprehensive set of methods used in experiments, the authors report the best performance as compared to the state of art methods for English-Hindi code mix data. The state of art comparison is shown in Sect. 5.

The implications for the current work is manifold. We can use hate speech detection in chatbots of messengers and other related applications to automatically filter out hateful content. Also, it may be used by enforcement agencies and police to detect hate speech in order to manage law and order situation, especially in time of protests and social unrest. As is commonly seen, hate speech is used to provoke the mob against a specific religion, caste, or gender for violence. This research may help mitigate such adverse instances of in society, especially against minorities. The paper is organised into following sections: Sect. 2 discusses the recent contributions in the literature, machine learning and deep learning methods. Section 3 describes the pre-processing, vectorization, and model details used in the paper. Section 4 discusses analysis the results obtained. Section 5 compares the results of the proposed approach with recent state of art methods. Section 6 concludes the paper with final thoughts.

2 Literature survey

Nowadays, hate speech is a major concern now a days on various social media platforms. In the past, applications of ML (machine learning) and DL (deep learning) setup the milestone in related tasks as well [9]. This section describes literature review on hate speech detection using ML and deep learning methods along with papers introducing relevant datasets for English-Hindi code mix datasets. In the past, researchers have focused on how to identify hate speech [10–17]. Whereas many earlier research studies have focused on hate speech detection in single languages (mostly English), there have been far fewer works in hate speech detection from Hindi–English code-mixed data, primarily due to challenges of code-mixed languages and the lack of datasets. This section focuses on hate speech detection methods specifically on Hindi-English code mix data. Some of authors used sentimental analysis the form of number of classes to evaluate the hate sentences[18].

A Hindi-English code mix dataset is released in the paper [19]. In this paper, the authors develop annotated database of Hindi-English data from Facebook and Twitter. The corpus is created at 3 level tags (Aggression, over Aggression and Non aggression) with 18k Tweets and 21k Facebook comments. This research has not given any experimental evaluation on mentioned dataset.

An annotated corpus of YouTube video comments in automated vehicle is proposed in the paper [20]. The proposed dataset contains 50k YouTube videos comments,

with inclusion of data format and its possible uses. Also, the authors discuss the case study on proposed the corpus to understand the public opinion on self-driving and reactions on accidents using cars.

In the paper [21], the authors proposed text classification using Hinglish text written in Roman script, collected random Hinglish data from the news and Facebook comments. They proposed various combinations of feature identification methods using TF-IDF representation and concluded that Radial Basis Function Neural Network as the best combination to classify in the Hinglish text.

In this paper [22], the authors discuss issues during hate speech detection in Hindi–English code mix texts. They proposed Hind–English mix data collected from twitter. The tweets are annotated at word level with Hate and Normal speech classification. Finally, they proposed a ML based system for hate speech detection with accuracy of 71.7%.

In the paper [23], the authors proposed hatred detection mechanism in three languages (English, Spanish and Italian). They proposed the methods for evaluating the relationship between misogyny and abusive language and discuss the scope misogyny detection in cross-lingual platform. They implement their experiments on Automatic Misogyny Identification (AMI) datasets.They conclude their research with remark that misogyny is type of abusive language and proposed architecture provides robust performance across the languages.

Hate speech detection for Hindi–English code mix data in proposed in the paper [24]. In this paper, authors collected hate and non-hate data from various sources (Twitter and shared task HASOC) and applied popular pre-trained word embedding. They compare the proposed model with various feature extraction methods and commented that fastText features outperform with Support Vector Machine (SVM) - Radial Basis Function (RBF) classifier accuracy 0.8581%, precision of 0.8586%, recall 0.8581 and F1-score 0.858%.

In the paper [25], the authors proposed DL (deep Learning) methods for detection of hate speech from Hindi-English code mix data on benchmark dataset. They experimented DL models using domain-specific embeddings and received results with accuracy 82.62%, precision 83.34 and F-score 80.85% with CNN model.

In this paper [26], the authors proposed a deep learning model for offensive speech detection. In this paper, they created self made Hindi–English code mix dataset with annotation and applied ML models as baseline model. Finally, they proposed Multi-Channel Transfer Learning based model (MIMCT) and concluded that proposed model outperforms state-of-art methods.

Again, a deep learning model for detecting offensive tweets in Hindi–English language is proposed in the paper [27]. In this paper, the authors introduce a novel tweet dataset, titled Hindi–English Offensive Tweet (HEOT). The tweets were labelled into three categories: non offensive, abusive and hate-speech. Further, they evaluate the results on CNN model and reported accuracy 83.90%, precision 80.20%, recall 69.98% and F1-score 71.45%.

The evaluation of Hindi–English code mix data from social media is proposed in the paper [28]. In the first part of research, the monolingual embedding was used, however in second part they used supervised classifier with transfer learning on English dataset and tested on code-mix dataset. They reported results with improvements of F1-score of 0.019.

A deep learning model is proposed for hate speech detection in the social media text [29]. The authors used HASOC 2019 corpus to evaluate the proposed model and reported a macro F1 score of 0.63 in hate speech detection on the test set of HASOC.

In the paper [30], the authors proposed pipeline for hate speech detection in Hindi–English code-mix data (Hinglish) on social media platforms. Before finalizing the proposed system, the authors experimented regress comparison against several available benchmark datasets. Also, they evaluated the relationship of hate embeddings along with social network-based features, and reported that proposed system outperform with state of the art.

A deep learning approach for hate speech detection in Hindi–English code-mix is proposed in the paper [31]. In this paper, the authors used character level embedding for feature extraction. They implemented various deep learning classifiers and commented that hybridisation of GRU (Gated Recurrent Unit) with Attention Model performed best among all experimented models.

In the paper [32], the authors deal with identification of hate speech from code-mixed text using deep learning models. They used publicly available datasets and perform two sub-word level LSTM model and Hierarchical LSTM model and reported F1-score of 48.7%.

A deep learning approach is proposed for hate speech emotion detection in the paper [33]. The authors collected more than 10,000 Hindi–English code mix dataset and annotate them with happy, sad and anger. They used bilingual model for feature vector generation with deep neural network as a classification model.They reported results that CNN-Bi-LSTM performs better with 83.21% classification accuracy. A transfer learning with LSTM based model is for hate speech classification in Hindi–English code mix data [34]. The authors reported that proposed system improve the performance as the state-of-the-art method.

In the paper [35], the authors discuss the relationship among - aggression, hate, sarcasm, humor, and stance in Hinglish (Hindi–English) text. They evaluated various existing deep learning methods of hate speech detection in code-mix texts. Furthermore, they proposed evaluation scheme

of identifying the offensive keywords from Hindi–English code mix data.

Hate speech in Hindi–English mix text is proposed in the paper [36]. To design the structure framework, the authors used existing algorithms to develop the 'MoH' (Map Only Hindi). They evaluated the models on three three datasets, and computed performance using precision, recall and F1-score. The final results of proposed model reported 15% higher that baseline model. They commented that results demonstrate a significant improvement in the state-of-the-art scores on all three datasets.

On the basis of literature review, it was observed that there are considerable gaps in hate speech detection in reference to English–Hindi code mix data. Three major issues were identified and addressed as described here. First, the lack of large datasets for training is one of the most important issue. It is addressed by consolidating a large dataset by merging 3 publicly available datasets. Second, the challenges of code-mix data have been addressed by using established as well as novel approaches in machine learning and deep learning. Finally, we evaluate the methods comprehensively using all available performance metrics. As can be seen in Table 11, many previous works have not calculated performance using all popular metrics.

3 Methodology

This section describes the datasets used in the work. It also expresses the evaluation of machine learning and deep learning approaches for hate speech detection on the selected datasets. The section concludes by describing a custom, novel model that provides the state-of-art performance for hate speech detection on English–Hindi code mix data.

3.1 Dataset description

Research in hate speech detection has been pursued for many years now, especially in English language. Consequently, there are many datasets available for English hate speech detection. In contrast and as observed from literature, it has been found that there are limited and comparatively smaller datasets for English–Hindi code mix data. In order to overcome the issue of smaller datasets, the authors use a consolidated English–Hindi code mix dataset in this study, derived from 3 publicly available datasets. Below, we detail the three corpora that we utilized in our experiments, including Bohra [22], Kumar [37] and HASOC [38].

Bohra 2018 dataset [22]: This is the first dataset that has been used in the present study. It contains 4500 tweets. Each tweet is labelled either as "Yes" for a tweet containing hate speech or as "No" for a tweet not containing hate speech. Out of 4500 instances, there are 2345 instances of "Yes" (hate speech) and 2155 instances of "No" (non-hate speech). Most of the tweets in the dataset are in Hindi –English mixed language and are written using standard Roman alphabet. Some examples of the dataset along with their ground truth labels are shown in Table 1.

Kumar 2018 dataset [37]: Dataset is mainly extracted from YouTube comments and other social media platforms. Originally, the dataset is divided into three classes i.e., Not Aggressive (NAG), Covertly Aggressive (CAG) and Overtly Aggressive (OAG). To make it similar to the other two datasets, we have merged the two classes (CAG and OAG) into

Table 1 Sample tweets from Bohra [22] dataset along with labels

Text of the Tweet	Label
Aise logo se sakht nafrat karta hu Jo caste ko naam ke sath jod ke chaude hote h but real me vo piddu hote h	Yes
I am very sorry to say saaf dil shilpa ke fans hiten ke dil mein kya hai woh bhi samjhte hain hadh hoti hai nafrat ki bhi	No
Achha bta diya hum n show hi nhi dekha tha. Khbri krta kuch ni sirf mAA bnta h kr hina ko hate	No
Bhai, ye log honour killing mein vishwas rakhne walo me se hain, khud dusri ki beheno ko din raat chedege, kisine inki behen pe tippani bhi kardi toh murder vegere ki dhamki dete hain	Yes

Table 2 Sample tweets from Kumar [37] dataset along with labels

Text of the tweet	Label
It is a good gesture for rewarding to the individual who have made us glorious but at the same time to improve the fate of Indian sports in the international arena some generosity is also needed to improve the infrastructure to facilitate many such talents	No
Indian express did it again...have some moral please	Yes
Really motivating programme, congratulations to CNBC 18, and ofcourse the business ticun and legend Mukeshbhai	No
What's wrong with you secular idiots	Yes
Maybe keralites eats infants as well along with beef and pork ?	Yes

Table 3 Sample tweets from HASOC 2021 dataset along with annotations

Text of the tweet	Label
@ashokepandit @Lawyer_Sandeep @srivatsayb इनका धंधा पानी तो चीन ही चला रहा है क्यों बोलेंगे ये?	Yes
@sptweetz @srivatsayb Ram Rajya us coming to lakshya dweep. Secularists are scared.	No
@ashokepandit लूटतंत्र खतरे मे आ गया फिरसे..😂	No
@kapsology Is there anything that is allowed in Korea	No
@kapsology Oh, neechta ki koi kami nai hai.. inko bhi peeche kar de.. itna neech hai humare wala	Yes

one class i.e., Hate speech and NAG as Non-hate speech. In this dataset, there are a total of 11,100 instances, out of which 5834 instances are in "Hate speech" category and 5266 instances are in "Non-hate" category. Some samples from the dataset are shown in Table 2.

HASOC 2021 Hindi–English Coded dataset [38]: In this dataset, social media tweets are taken in Hindi–English code mix language. The Hind-English code mix data consists of 5000 instances that are labelled either as "Hate" (2258 instances) or "Non-hate" (2742 instances). Some examples from the dataset along with their ground truth labels are depicted in Table 3.

The final consolidated dataset contains 20600 instances. It is divided into training(70%), validation (15%) and testing(15%) splits. The overall description of the dataset are shown in Fig 1.

3.2 Text preprocessing

The aforementioned datasets were pre-processed before they were fed into model. The pre-processing involves removal of certain elements from the tweets and comments of the dataset. Extra spaces, missing values and unreadable characters were removed systematically and carefully. Hyperlinks are often unnecessary in hate speech classification. These were scanned using regular expressions and dropped. Hashtag characters (#) and emoticons/emojis were also removed. Again, we investigated the contribution of emojis to hatefulness and evaluated the contribution of text with and without emojis. It was concluded that emojis were not useful for our task. After that, the processed data was tokenized using an `NLTK` library-based tokenizer and punctuation marks were removed. Finally, `NLTK` library-based `PorterStemmer()` was used to stem every word in the word list. During

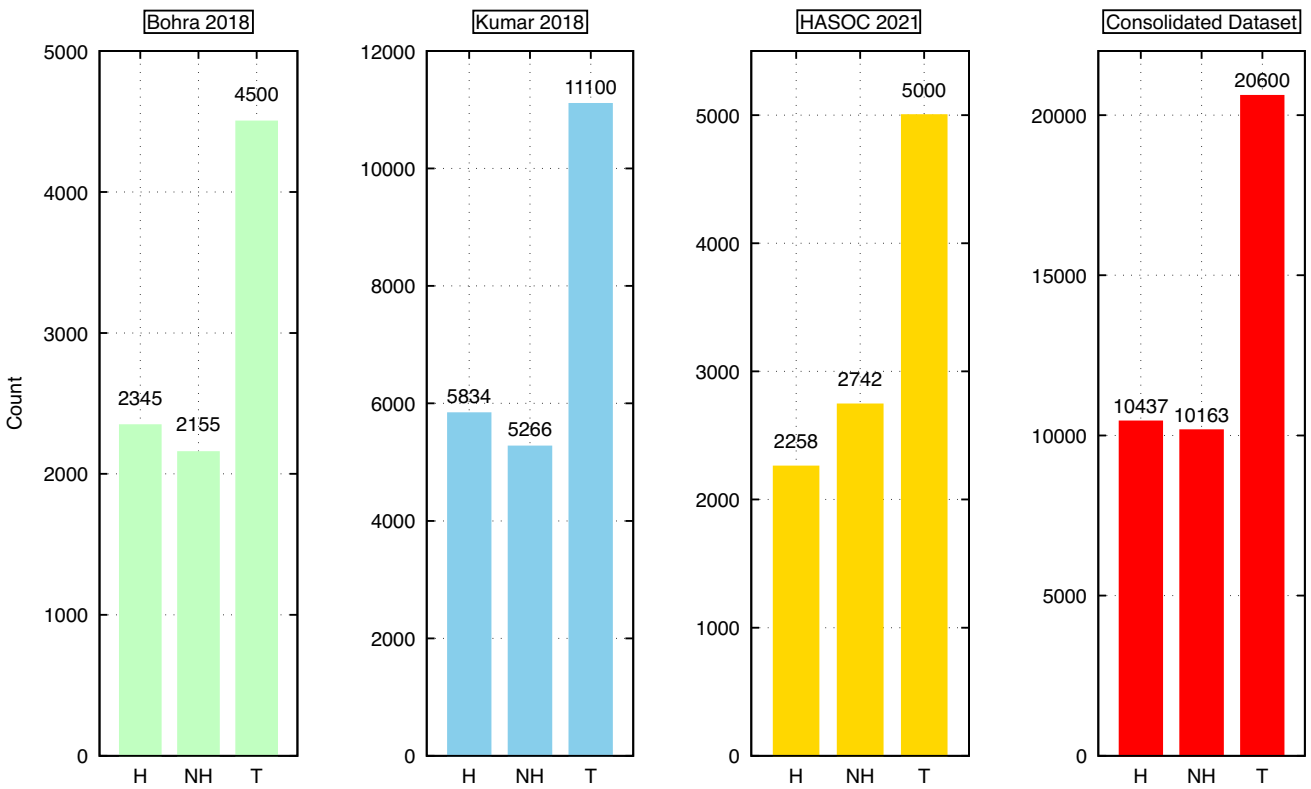


Fig. 1 Distribution of Hate (H) and Non hate (NH) speech tweets and comments in the datasets. Total is indicated by T

the pre-processing step, certain errors were observed at the stemming stage. This was due to the fact that Porter stemmer is suited to English language while the data contains mixed and Hindi words as well. Thus, such words were not being stemmed properly, leading to erroneous outputs. To resolve the mentioned issue, some of the Hindi tweets were translated to equivalent English tweets using the Google Translate free service.

3.3 Feature extraction for machine learning techniques

Discriminating features are critical for machine learning methods to function well and can often be the difference between success and failure of a task. In this work, we used 4 different techniques for the purpose of extracting the features from the tokenized text data. First, we apply Count Vectorizer that is used to convert text documents into matrix of token counts. Second, we used Hashing Vectorizer that is used for converting text documents to a matrix of token occurrences. Third, we applied Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer that is used to convert a collection of raw documents into a matrix of TF-IDF features. Finally, we applied TF-IDF Transformer that transforms a count matrix into a normalized TF or TF-IDF representation.

3.4 Word embedding for deep learning techniques

Word embeddings help to express the syntactic and semantic context of a word or term in documents. It helps to understand how similar a term is in relation to others in a document. It is a word representation method that assigns each term in the text data a vector form or numeric features for modelling purposes. The assigned vector is intended to capture the semantic meaning of the terms. In this work, we have applied three word embedding techniques that are briefly discussed as follows.

Fastext: An open-source library that allows high level models to utilize text representation for various text processing tasks. The English-based algorithm is being used for the vectorization of words.

Glove 60b 100D: It is an unsupervised learning algorithm primarily used for converting text data into vector form. It is trained on global word corpus, and utilizes relationships

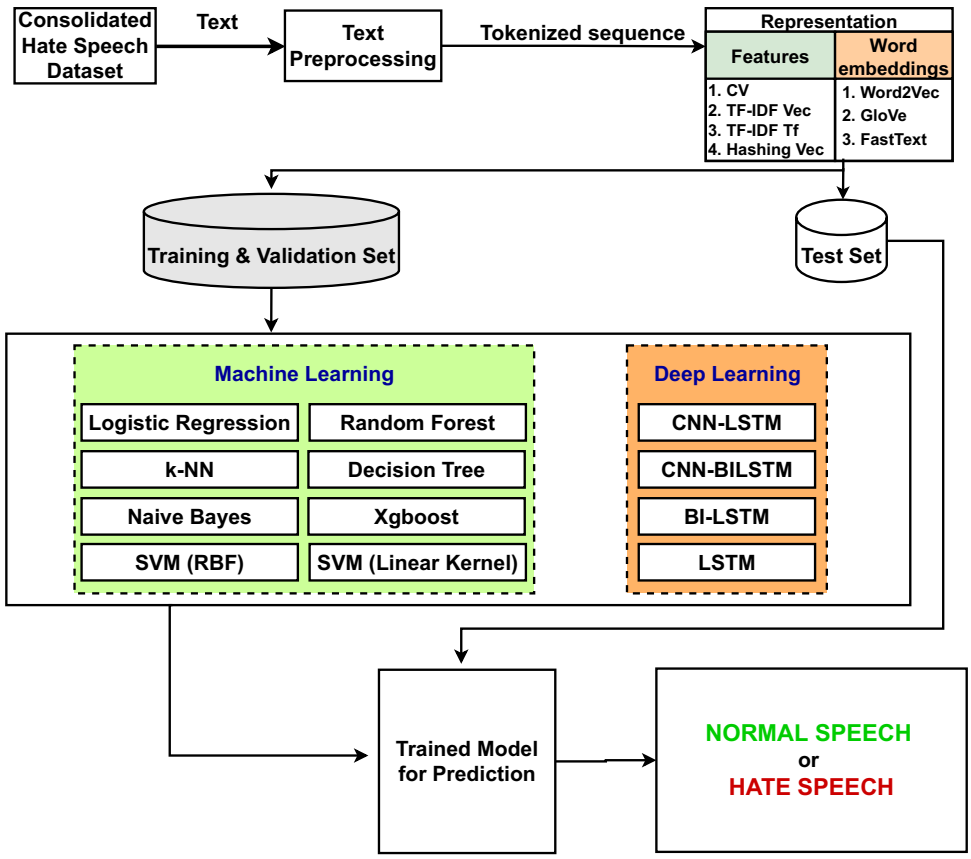


Fig. 2 Overall experimental setup used in the study. *CV* stands for Count Vectorizer, *TF-IDF Vec* is TD-IDF Vectorizer, *TF-IDF Tf* means TF-IDF Transformer and Hashing Vec represents Hashing Vectorizer

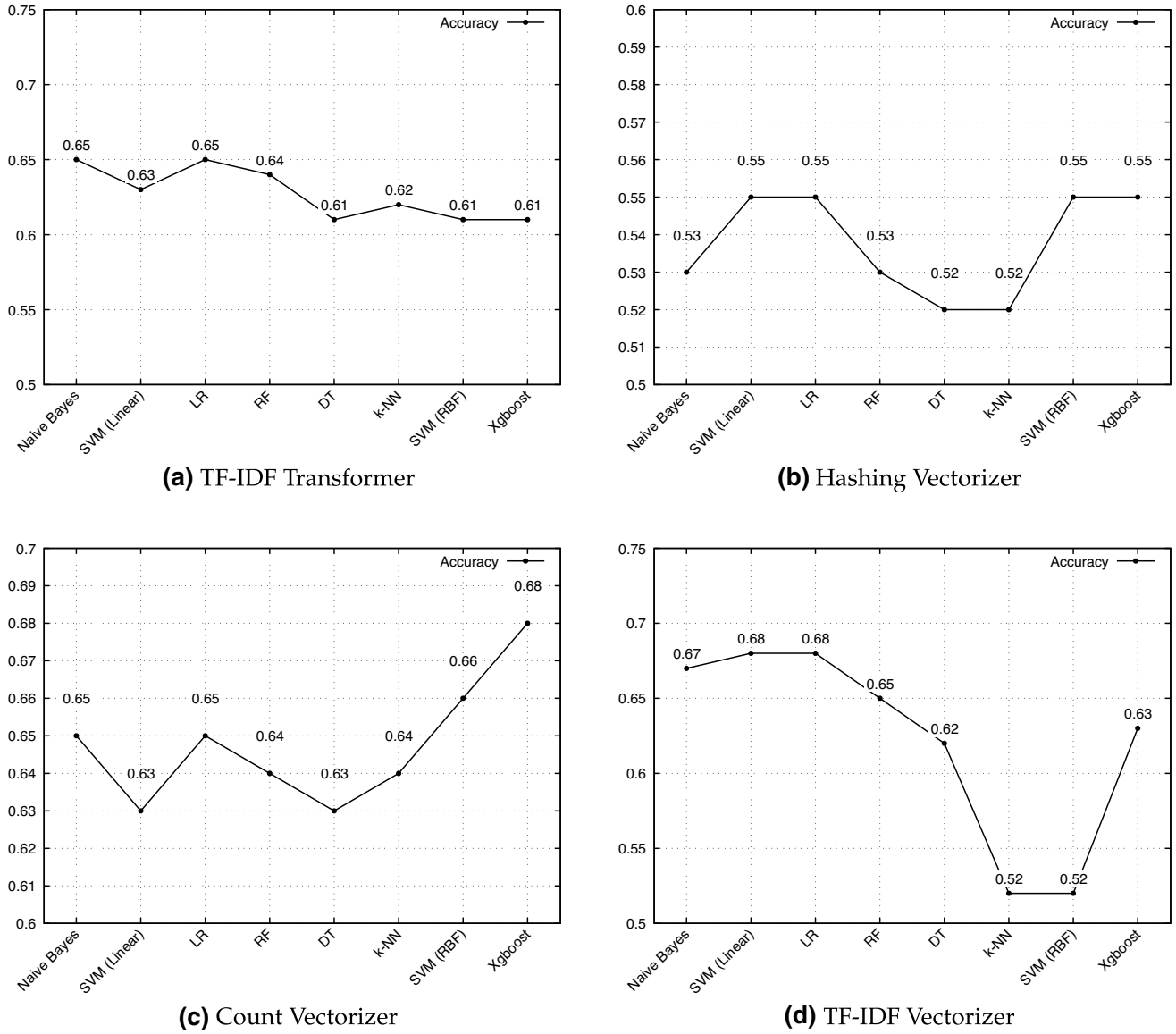
among words such as linear substructures in the word vector space.

Word2vec: It is based on a neural network model to find patterns in word association in a large text corpus. After it was trained, it detects partial sentence and synonymous words. The words are mapped into vector space in such a way that the opposite words are oriented in opposite directions and similar words are oriented towards the same direction.

3.5 Machine learning approach used

Our objective of implementation of machine learning models is to establish the baseline approach with the mentioned feature extraction methods. We split the final pre-processed dataset into train, validation and test sets as 70%, 15% and 15% respectively. Python's `scikit` implementations of Naïve Bayes (NB), Decision Tree (DT), Logistic Regression (LR), Random Forests (RF), Xgboost (XG) and Support Vector Machine (SVM) were used in this work. Brief descriptions of these methods are given below.

The simple Logistic Regression [39] is a binary classifier that classifies the examples into two classes. A tree structure classification model called Decision Tree is also used in this work. The advantage of decision trees is that they are easy to understand and can be efficiently induced from data. It is one of the oldest model and popular technique for learning discriminatory models [40]. XGBoost is the gradient boosted decision tree that is employed to increase the performance of machine learning operations [41].



(a) TF-IDF Transformer

(b) Hashing Vectorizer

(c) Count Vectorizer

(d) TF-IDF Vectorizer

Fig. 3 Graphs depicting the accuracy of machine learning methods with different feature extraction methods (*LR* logistic regression, *DT* decision tree, *RF* random forest)

Table 4 Results of machine learning models with TF-IDF transformer feature

Model	**Accuraczy**	Precision	Recall	F1 Score	ROC_AUC
Naïve Bayes	0.65	0.71	0.56	0.63	0.68
SVM (Linear)	0.63	0.66	0.62	0.64	0.66
Logistic regression	0.65	0.68	0.63	0.66	0.70
Random forest	0.64	0.67	0.62	0.64	0.68
Decision tree	0.61	0.66	0.63	0.62	0.71
k-NN	0.62	0.67	0.62	0.64	0.62
SVM (RBF)	0.61	0.65	0.63	0.62	0.65
Xgboost	0.61	0.68	0.64	0.62	0.67

Table 5 Results of machine learning models with Hashing vectorizer feature

Model	Accuracy	Precision	Recall	F1 Score	ROC_AUC
Naïve Bayes	0.53	0.50	0.64	0.58	0.56
SVM (Linear)	0.55	0.57	0.52	0.60	0.58
Logistic regression	0.55	0.56	0.66	0.60	0.59
Random forest	0.53	0.56	0.48	0.52	0.60
Decision tree	0.52	0.55	0.54	0.54	0.57
k-NN	0.52	0.55	0.54	0.54	0.60
SVM (RBF)	0.55	0.55	0.73	0.63	0.57
Xgboost	0.55	0.54	0.65	0.60	0.58

Table 6 Results of machine learning models with count vectorizer feature

Model	Accuracy	Precision	Recall	F1 Score	ROC_AUC
Naïve Bayes	0.65	0.71	0.56	0.63	0.71
SVM (Linear)	0.63	0.66	0.62	0.64	0.72
Logistic regression	0.65	0.68	0.63	0.66	0.68
Random forest	0.64	0.67	0.62	0.64	0.64
Decision tree	0.63	0.66	0.65	0.68	0.77
k-NN	0.64	0.64	0.68	0.63	0.65
SVM (RBF)	0.66	0.67	0.62	0.65	0.71
Xgboost	0.68	0.68	0.65	0.63	0.72

Ensemble learning method i.e., Random Forests [42] is applied in the current work. The method fits many similar decision trees by identifying random samples from training set. Prediction for test example is assigned based on highest vote of individual trees within the forest. The two class hyperplane based classification algorithm, named as Support Vector Machine (SVM) [43], is used in this work. It treats two classes as positive and negative (S_i = Yes, No)

Table 7 Results of machine learning models with TF-IDF vectorizer feature

Model	Accuracy	Precision	Recall	F1 Score	ROC_AUC
Naïve Bayes	0.67	0.72	0.60	0.65	0.71
SVM (Linear)	0.68	0.70	0.69	0.70	0.69
Logistic Regression	0.68	0.69	0.71	0.70	0.64
Random Forest	0.65	0.68	0.60	0.64	0.67
Decision Tree	0.62	0.64	0.62	0.63	0.66
k-NN	0.52	0.55	0.48	0.51	0.55
SVM (RBF)	0.52	0.52	0.55	0.69	0.57
Xgboost	0.63	0.65	0.85	0.71	0.68

and strives to obtain a hyperplane that separate those two classes with at least a constant distance. The *k*-NN (*k*-Nearest Neighbour) [44] is a method which is used to define the classes on the basis of the class of their neighbours. It searches the *k* nearest samples over all train dataset, and assigns the class based on those nearest classes to the test sample. Naïve Bayes [45] assumes predictor independence and is a probability based classification method. It is based on Bayes' Theorem and therefore called as Naïve Bayes due its assumption of variable independence. In simple terms, a Naïve Bayes classifier assumes no relation or dependence between the occurrence or non-occurrence of individual features. All the mentioned models are finally evaluated on all popular metrics such as precision, recall, accuracy, F1-score, ROC-AUC score.

3.6 Deep learning methodology

In wake of tremendous success of deep learning approaches in various domains in general and NLP, in particular, four customized models along with 3 different word embeddings were utilized for experiments. First, Long Short-Term Memory (LSTM) [46] model was used due to the fact that this architecture has recurrent connections and helps to model sequences well. It further eliminates the issues faced in basic Recurrent Neural Network (RNN) based models, especially the issue of slow learning over long sequences. Bidirectional LSTM (BiLSTM) [47] is simply the augmented form of the basic LSTM. Instead of one, two LSTMs are used in conjunction, one for forward learning and another for the backward sequence. It aims to reduce the shortcomings faced by basic LSTM by adding to its learning capability. Convolutional Neural Networks (CNN) [48] are commonly associated with image based tasks. However, they are widely recognized for their ability to learn intermediate feature. One-dimensional CNNs were utilized in this study as a feature extractor in the initial stage of the model. The results and analysis of all the models used with different feature

extraction and word embeddings are discussed in the next section.

After evaluating several different models, it was indicated by the results that the CNN-BiLSTM model with word2vec embedding show the best performance. The implementation details of this models is as follows. The model is implemented using Keras API. The Sequential model is composed of several layers. The embedding layer is the first layer of the network. This is the input layer where the model is fed the training data. The pre-trained word embeddings are used by giving the prepared embedding matrix as starting weights. To reduce the effect of overfitting, the next layer is a Dropout layer with a rate of 0.3. The 3rd layer is a one-dimensional CNN layer (`Conv1D`) that has 64 filters of size 5x5 to extract local features, along with `ReLU` as the activation function. In the next layer, vectors are pooled (`MaxPooling1D`) with a window size of 4. The BiLSTM layer that follows receives the pooled feature maps. This information is utilised to train the BiLSTM that outputs long-term dependent features of input while keeping memory. The dimension of the output is set to 128. Next, another Dropout layer is added with a rate of 0.3. The final layer of model is a dense layer. Here the vectors are classified as real or fake. Sigmoid is used as a Activation function in this layer. Binary cross-entropy is used as the loss function and Adaptive Moment Estimation (`Adam`) is used as the optimizer. Training of the model is performed with a batch size of 64. Figure 2 and Algorithm 1 shows the overall experimental design of the paper.

Algorithm 1 Steps in proposed methodology

1: **for** each input sample in the training set, **do**
2: Pre-process the input text
3: Extract features using one of CV, TF-IDF Vec, TF-IDF Tf, Hashing Vec, Word2Vec, GloVe, FastText
4: Pass the features to the 1D Convolution layer for further features extraction from existing features
5: Send the features from the CNN layer to the BiLSTM layer for modeling dependencies between words in a sentence
6: Forward the outputs from BiLSTM layer to the Dense layer for final classification of hate speech
7: **end for**

4 Results and analysis

In order to explore the problem from multiple perspectives and achieve the best possible performance on the task, eight (8) different machine learning models, in conjunction with 4 different feature extraction methods, were applied. The results of the machine learning models are shown in Table 4, 5, 6 and 7. The machine learning accuracy graphs are depicted in Fig. 3. It was observed that Naïve Bayes method perform well in terms of precision. In regard to recall, the performance is distributed among different methods but Xgboost shows potential by yielding best results with two of the feature extraction techniques. The performs of the models in reference to F1-score and ROC-AUC metrics is also divided and no single model is providing best results with all feature extraction techniques. In terms of accuracy, the performs is not conclusive by Xgboost has an edge among other methods. Overall, Xgboost seems to be the most promising among the machine learning techniques with good performance in most metrics across all four features extraction techniques.

Due to the inconclusive results of the machine learning experiments to indicate a single best performing model for the task, the authors were motivated to employ several

Table 8 Results of deep learning models with FastText word embedding

Model	Accuracy	Precision	Recall	F1-Score	ROC_AUC
LSTM	0.65	0.65	0.66	0.65	0.70
BiLSTM	0.65	0.68	0.62	0.65	0.71
CNN-LSTM	0.64	0.71	0.57	0.63	0.71
CNN-BiLSTM	0.67	0.67	0.67	0.67	0.71

Table 9 Results of deep learning models with Glove-6B-100d word embedding

Model	Accuracy	Precision	Recall	F1-Score	ROC_AUC
LSTM	0.72	0.70	0.71	0.70	0.74
BiLSTM	0.71	0.71	0.70	0.71	0.75
CNN-LSTM	0.68	0.68	0.68	0.68	0.72
CNN-BiLSTM	0.72	0.72	0.71	0.72	0.77

Table 10 Results of deep learning models with word2vec word embedding

Model	Accuracy	Precision	Recall	F1-Score	ROC_AUC
LSTM	0.760	0.750	0.740	0.745	0.780
BiLSTM	0.770	0.770	0.746	0.744	0.800
CNN-LSTM	0.780	0.778	0.760	0.760	0.810
CNN-BiLSTM	0.876	0.830	0.840	0.835	0.880

promising deep learning approaches based on the suggestions from the literature. Consequently, 4 different model architectures, along with 3 different word embedding techniques, were finalized for experiments. The results of the deep learning models are shown in Table 8, 9 and 10. The accuracy graphs for deep learning methods are depicted in Fig. 4. It may be observed that the CNN-BiLSTM model provides the best results among all the models, across all the metrics. Moreoever, with word2vec word embedding, it provides the best values for the metrics. It is observed from the results that BiLSTM improves the results with respect to the basic LSTM. This is expected due to the inherent dual modeling capability of the BiLSTM architecture. Furthermore, it is seen that adding a CNN feature extraction layer with BiLSTM adds to the performance. Thus, a 1-D CNN layer as a feature extractor seems promising. However, the same trend is not observed

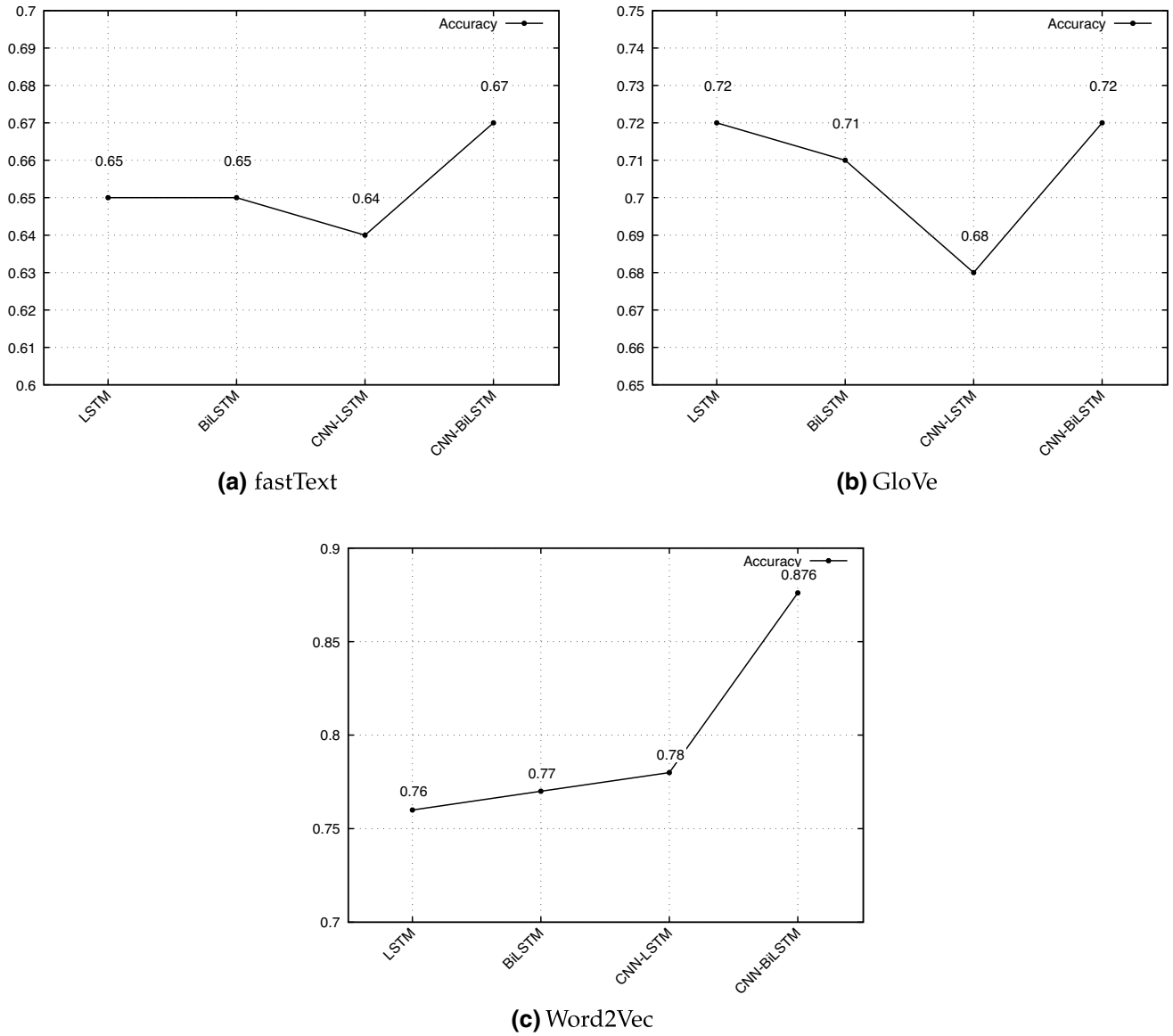


Fig. 4 Graphs depicting the accuracy of deep learning methods with different word embeddings

Table 11 Comparison of the proposed approach with recent state-of-art methods. (* indicates that results evaluated by Chopra et al. [30])

Year	Model	Dataset	Accuracy	Precision	Recall	F1-score
2018 [25]	CNN-1D	HS[e]	0.826	0.833	0.785	0.808
2018 [22]	SVM	HS	0.717	–	–	*0.620
2018 [22]	Random forest	HS	0.667	–	–	–
2018 [27]	TT-CNN[c]	HEOT	0.839	0.802	0.698	0.714
2019 [34]	LSTM based	HEOT	0.870	–	–	*0.730
2019 [34]	LSTM based	HS	*0.740	–	–	*0.710
2019 [32]	H-LSTM-Att[d]	HS	0.666	–	0.451	0.487
2019 [32]	H-LSTM-Att	HEOT	*0.630	–	–	*0.520
2020 [33]	CNN-BiLSTM	CD[f]	0.832	0.832	0.832	0.832
2020 [28]	Bi-LSTM	SE[g]	–	0.639	0.632	0.635
2020 [30]	DW-Debias[a]	HS	0.780	–	–	0.730
2020 [30]	Debias[b]	HEOT[h]	0.850	–	–	0.770
2023 [49]	TL-XLMR [i]	SE[g]	0.713	–	–	0.716
2023 [49]	TL-XLMR [j]	SE[g]	0.660	–	–	0.644
Proposed	**CNN-BiLSTM**	**CoD**[k]	**0.876**	**0.830**	**0.840**	**0.835**

[a]FT+CNN+BiLSTM+Attn+PV+DW+Debias

[b]FT+CNN+BiLSTM+Attn+PV+Debias

[c]Ternary transfer learning with CNN

[d]Hierarchical LSTM model with attention based on phonemic sub-words

[e]Bohra 2018 dataset

[f]Custom dataset containing 12000 code mixed English–Hindi tweets

[g]SemEval-2020 code-mixed Hinglish Data

[h]Mathur et al. 2018 dataset

[i]Transfer learning-based XLMR multitask learning(Sentiment)

[j]Transfer learning-based XLMR multitask learning(Emotion)

[k]Our consolidated dataset

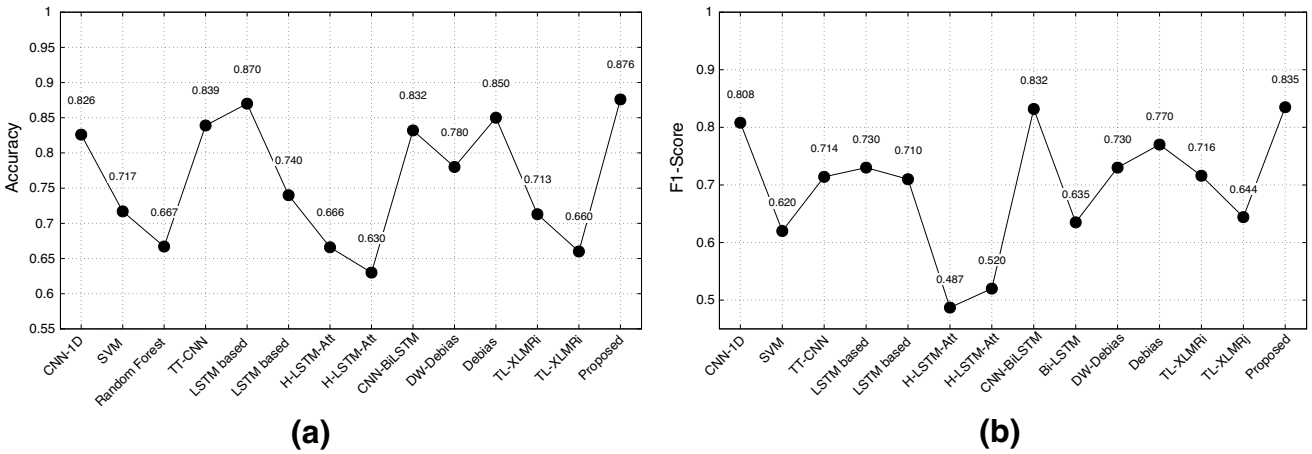


Fig. 5 Comparison of (**a**) accuracy (**b**) F1 score of the proposed approach with recent state-of-art methods

with the basic LSTM, especially with GloVe and FastText word embeddings. It may be interesting to precisely determine the contribution of CNN output to the overall architecture in future experiments.

5 Comparison with state of art

This section describes the comparison of the proposed work with recent state of art methods. As shown in Table 11, the proposed approach surpasses the state-of-art methods, in all popular metrics such as accuracy, precision, recall and F1-score, for the detection of hate speech from English-Hindi code-mix data. It is important to note that works with greater accuracy exist for code-mix languages other than English and Hindi. However, in order present a fair comparison of the proposed approach, the works done on the same language pair – English and Hindi – and on similar datasets are shown in the comparison. Furthermore, the graphs of accuracy and F1-score is shown in Fig 5a and b.

6 Conclusion

Research in hate speech detection in single language, especially English, has reasonably matured. It is not the same case, however, for other low-resourced and code mixed languages such as English–Hindi mixed language. This paper presents an attempt to address hate speech detection in Hinglish (English–Hindi mixed) language. To address the issue of small sized datasets, a consolidated dataset was created by merging 3 publicly available datasets, leading finally to more than 20,000 sample points in the final dataset. Furthermore, several machine learning and deep learning models were applied to detect hate speech. After the experiments, it was concluded that CNN-BiLSTM model provides the best accuracy among all methods. The proposed CNN-BiLSTM based approach outperforms all recent state of art methods for hate speech detection in English–Hindi code mix datasets, yielding 87.6% accuracy on the consolidated dataset.

Author Contributions AK, Shivani, Kusum: software, investigation, data curation, writing – original draft. AKY: conceptualization, methodology, supervision, writing – review & editing. MK: methodology, writing – review & editing, visualization. DY: conceptualization, methodology, formal analysis

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. India Social Media Statistics (2021) The Global Statistics. 2021 Dec. Available from: https://www.theglobalstatistics.com/india-social-media-statistics/
2. Hate speech (2021) Wikimedia Foundation; Available from: https://en.wikipedia.org/w/index.php?title=Hate_speech &oldid=1059042962
3. Council of Europe;. Available from: https://www.coe.int/en/web/portal/home
4. Available from: https://www.coe.int/en/web/portal/home
5. 2020.Available from: https://backlinko.com/instagram-users
6. Khanday AMUD, Khan QR, Rabani ST (2021) Identifying propaganda from online social networks during COVID-19 using machine learning techniques. Int J Inform Technol 13:115–22
7. Hamid Y, Elyassami S, Gulzar Y, Balasaraswathi VR, Habuza T, Wani S (2022) An improvised CNN model for fake image detection. Int J Informat Technol 15:5–15
8. Yadav AK, Singh A, Dhiman M, Kaundal R, Verma A, Yadav D (2022) Extractive text summarization using deep learning approach. Int J Informat Technol 14(5):2407–15
9. Bharti S, Yadav AK, Kumar M, Yadav D (2021) Cyberbullying detection from tweets using deep learning. Kybernetes 51(9):2695–2711
10. Poletto F, Basile V, Sanguinetti M, Bosco C, Patti V (2020) Resources and benchmark corpora for hate speech detection: a systematic review. Lang Res Eval 55:477–523
11. Shah SR, Kaushik A (2019) Sentiment analysis on indian indigenous languages: a review on multilingual opinion mining. arXiv preprint arXiv:1911.12848
12. Kaur S, Singh S, Kaushal S (2021) Abusive content detection in online user-generated data: a survey. Procedia Comp Sci 189:274–81
13. Yadav A, Vishwakarma DK (2020) Sentiment analysis using deep learning architectures: a review. Artifi Intel Rev 53(6):4335–85
14. Drias HH, Drias Y (2020) Mining twitter data on COVID-19 for sentiment analysis and frequent patterns discovery. medRxiv 18:2020
15. Thakur V, Sahu R, Omer S (2020) Current State of Hinglish Text Sentiment Analysis. In: Proceedings of the International Conference on Innovative Computing & Communications (ICICC)
16. Srivastava V, Singh M (2021) Hinge: A dataset for generation and evaluation of code-mixed hinglish text. arXiv preprint arXiv:2107.03760
17. Akuma S, Lubem T, Adom IT (2022) Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets. International Journal of Information Technology 14, 3629–3635.
18. Kumar P, Vardhan M (2022) PWEBSA: Twitter sentiment analysis by combining Plutchik wheel of emotion and word embedding. International Journal of Information Technology 14, 69–77.
19. Kumar R, Reganti AN, Bhatia A, Maheshwari T (2018) Aggression-annotated corpus of hindi-english code-mixed data. arXiv preprint arXiv:1803.09402
20. Li T, Lin L, Choi M, Fu K, Gong S, Wang J (2018) Youtube av 50k: an annotated corpus for comments in autonomous vehicles. In:2018 international joint symposium on artificial intelligence and natural language processing (iSAI-NLP). IEEE 2018:1–5
21. Ravi K, Ravi V (2016) Sentiment classification of Hinglish text. In: 2016 3rd International Conference on Recent Advances in Information Technology (RAIT). IEEE; p. 641-5
22. Bohra A, Vijay D, Singh V, Akhtar SS, Shrivastava M (2018) A dataset of hindi–english code-mixed social media text for hate speech detection. Proceeding of second workshop on computational modeling of people's opinions personality and emotions in social media. IEEE
23. Pamungkas EW, Basile V, Patti V (2020) Misogyny detection in twitter: a multilingual and cross-domain study. Info Process Manag 57(6):102360

24. Sreelakshmi K, Premjith B, Soman K (2020) Detection of hate speech text in Hindi-English code-mixed data. Procedia Comp Sci 171:737–44
25. Kamble S, Joshi A (2018) Hate speech detection from code-mixed hindi-english tweets using deep learning models. arXiv preprint arXiv:1811.05145
26. Mathur P, Sawhney R, Ayyar M, Shah R (2018) Did you offend me? classification of offensive tweets in hinglish language. In: Proceedings of the 2nd Workshop on Abusive Language Online (ALW2); p. 138-48
27. Mathur P, Shah R, Sawhney R, Mahata D (2018) Detecting offensive tweets in hindi-english code-switched language. In: Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media; p. 18-26
28. Singh P, Lefever E (2020) Sentiment analysis for hinglish code-mixed tweets by means of cross-lingual word embeddings. In: Proceedings of the The 4th Workshop on Computational Approaches to Code Switching; p. 45-51
29. Kovács G, Alonso P, Saini R (2021) Challenges of hate speech detection in social media. SN Comp Sci 2(2):1–15
30. Chopra S, Sawhney R, Mathur P, Shah RR (2020) Hindi–english hate speech detection: Author profiling, debiasing, and practical perspectives. Proc AAAI Conf Artif Intell 34:386–93
31. Gupta V, Sehra V, Vardhan YR, et al (2021) Hindi-English Code Mixed Hate Speech Detection using Character Level Embeddings. In: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). IEEE; p. 1112-8
32. Santosh T, Aravind K (2019) Hate speech detection in hindi-english code-mixed social media text. In: Proceedings of the ACM India joint international conference on data science and management of data; p. 310-3
33. Sasidhar TT, Premjith B, Soman K (2020) Emotion detection in hinglish (hindi+ english) code-mixed social media text. Procedia Comp Sci 171:1346–52
34. Kapoor R, Kumar Y, Rajput K, Shah RR, Kumaraguru P, Zimmermann R (2019) Mind your language: abuse and offense detection for code-switched languages. Proc AAAI conf Artif Intell 33:9951–2
35. Sengupta A, Bhattacharjee SK, Akhtar MS, Chakraborty T (2021) Does aggression lead to hate? Detecting and reasoning offensive traits in hinglish code-mixed texts. Neurocomputing 488:598–617
36. Sharma A, Kabra A, Jain M (2022) Ceasing hate with MoH: hate speech detection in hindi-english code-switched language. Inf Proc Manag 59(1):102760
37. Zhu AZ, Thakur D, Özaslan T, Pfrommer B, Kumar V, Daniilidis K (2018) The multivehicle stereo event camera dataset: an event camera dataset for 3D perception. IEEE Robot Automat Lett 3(3):2032–9
38. Mandl T, Modha S, Shahi GK, Madhu H, Satapara S, Majumder P, et al (2021) Overview of the HASOC subtrack at FIRE 2021: Hate speech and offensive content identification in English and Indo-Aryan languages. arXiv preprint arXiv:2112.09301
39. Grimm LG, Yarnold PR (1995) Reading and understanding multivariate statistics. American Psychological Association;
40. Fürnkranz J (2010) In: Sammut C, Webb GI, editors. Decision Tree. Boston, MA: Springer US; p. 263-7. Available from: https://doi.org/10.1007/978-0-387-30164-8_204
41. Chen T, Guestrin C (2016) XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. New York, NY, USA: ACM; p. 785-94. Available from: http://doi.acm.org/10.1145/2939672.2939785
42. Liaw A, Wiener M et al (2002) Classification and regression by randomForest. R news. 2(3):18–22
43. Bahlmann C, Haasdonk B, Burkhardt H (2002) Online handwriting recognition with support vector machines-a kernel approach. In: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition. IEEE; p. 49-54
44. Samet H (2007) K-nearest neighbor finding using MaxNearestDist. IEEE Transactions on Pattern Analysis and Machine Intelligence. 30(2):243–52
45. Webb GI (2010) In: Sammut C, Webb GI, editors. Naïve Bayes. Boston, MA: Springer US; p. 713-4. Available from: https://doi.org/10.1007/978-0-387-30164-8_576
46. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural computation. 9(8):1735–80
47. Xu G, Meng Y, Qiu X, Yu Z, Wu X (2019) Sentiment analysis of comment texts based on BiLSTM. Ieee Access. 7:51522–32
48. Albawi S, Mohammed TA, Al-Zawi S, Understanding of a convolutional neural network. In, (2017) international conference on engineering and technology (ICET). Ieee 2017:1–6
49. Ghosh S, Priyankar A, Ekbal A, Bhattacharyya P (2023) Multitasking of sentiment detection and emotion recognition in code-mixed Hinglish data. Knowledge-Based Systems. 260:110182