Inteligencia Artificial 23(66), (2020), 97-111
doi: 10.4114/intartif.vol23iss66pp97-111

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Novel Approach for Generating Hybrid Features Set to Effectively Identify Hate Speech

Shruthi P(1,*) , Anil Kumar K M(2)
Department Of Computer Science and Engineering, JSS Science and Technology University, Mysuru, Karnataka, India.
[1]shruthip.sjce@gmail.com (*Corresponding Author) , [2]anilkm@jssstuniv.in

Abstract Automating hate speech or inappropriate text detection in social media and other internet platforms is gaining a lot of interest and becoming a valuable research topic for both industry and academia in recent years. It is more important for applications to identify the disruptive contents, understand sentiment analysis, identify cyber bullying, detect flames, threats, hatred towards people or particular communities or groups etc. Text classification is a very challenging task due to the nature and complexities with languages, especially its context, micro words, emojis, typo error and sarcasm present in the text. In this paper, we have proposed a model with a novel approach for generating hybrid features for an effective feature representation to classify hate speech. We have combined features learned from deep learning methods with the semantic features like word n-grams and tweets specific syntactic features to form hybrid feature sets. We have also improvised preprocessing steps to reduce the number of missing embeddings to increase the vocabulary for efficient feature learning. We have experimented with the various neural networks for feature learning and machine learning models with hybrid features for classification. Our work delivers hybrid features and appropriate preprocessing techniques for an efficient classification of the standard dataset of 16k annotated hate speech tweets. The combination of Long Short Term Memory (LSTM) trained on Random Embeddings for deep learning features extraction and Logistic Regression (LR) as a classifier with the hybrid features is found to be the best model and it outperforms the state of the art reported in the literature.

Resumen La automatización del discurso de odio o la detección de texto inapropiado en las redes sociales y otras plataformas de Internet está ganando mucho interés y se está convirtiendo en un tema de investigación valioso tanto para la industria como para el mundo académico en los últimos años. Es más importante que las aplicaciones identifiquen los contenidos disruptivos, comprendan el análisis de sentimientos, identifiquen el acoso cibernético, detecten llamas, amenazas, odio hacia personas o comunidades o grupos en particular, etc. La clasificación de textos es una tarea muy desafiante debido a la naturaleza y la complejidad de los idiomas , especialmente su contexto, micropalabras, emojis, error tipográfico y sarcasmo presentes en el texto. En este artículo, hemos propuesto un modelo con un enfoque novedoso para generar características híbridas para una representación de características efectiva para clasificar el discurso de odio. Hemos combinado características aprendidas de métodos de aprendizaje profundo con características semánticas como n-gramas de palabras y características sintácticas específicas de tweets para formar conjuntos de características híbridas. También hemos improvisado pasos de preprocesamiento para reducir la cantidad de incrustaciones faltantes para aumentar el vocabulario para un aprendizaje eficiente de funciones. Hemos experimentado con las diversas redes neuronales para el aprendizaje de funciones y modelos de aprendizaje automático con funciones híbridas para la clasificación. Nuestro trabajo ofrece funciones híbridas y técnicas de preprocesamiento adecuadas para una clasificación eficiente del conjunto de datos estándar de 16.000 tweets de incitación al odio anotados. La combinación de Long Short Term Memory (LSTM) entrenada en Embeddings aleatorios para extracción de características de aprendizaje profundo y Regresión logística (LR) como clasificador con características híbridas se considera el mejor modelo y supera el estado de la técnica informado en el literatura.

ISSN: 1988-3064 (on-line)

Keywords: Hybrid Features, Deep Learning, Hate Speech, Cyber-bullying, Deep Learning Features.
Palabras clave: Características híbridas, aprendizaje profundo, discurso de odio, ciberacoso, características de aprendizaje profundo.

1 Introduction

Social media is a platform for consuming and sharing of contents. In the present scenario, the Internet has become an important part of our lives and everybody is able to access the online communication media such as Twitter, Facebook, WhatsApp etc., and are free to express their thoughts and opinions. There has been a massive increase in the use of the Internet and social media. By the beginning of 2020, more than 4.5 billion people use the Internet, while the users of social media have reached the 3.8 billion mark, almost 60% of the world's population is online already [16]. As these platforms provide freedom to post comments and responses to comments, providers face the problem of managing enormous posts and replies. One of the major concerns is content filtering and specifically detecting hate speech from the posts and responses as the contents posted should not offend or threaten and lead to harm or hate others in the society. For Example, Anne Longfield, England's Children's Commissioner, wrote to social media companies, urging them to do all means possible to prevent posting of disturbing online contents. Her letter follows the suicide of 14-year-old after viewing distressing self-harm contents on Instagram [30].

There is no formal definition for Hate speech but there exists a consensus that, "Speech targeting disadvantaged social groups in a manner that is potentially detrimental to them" [31][32]. Waseem and Hovy[4] have defined tweet as offensive or hate, if it uses racial or sexist slurs, criticizes or attacks minority, promotes indirectly violence, contains problematic hashtags like "#whoriental", "#BanIslam", "#whitegenocide" , defends xenophobia or sexism, gives distorted views on minority etc. Davidson et al. [1] defined Hate speech as "Language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group". Badjatiya et al [6] have also defined it as "Hateful speeches are those that contain abusive speech targeting individuals (a politician, cyber-bullying, a product, a celebrity) or particular groups (LGBT, a country, a religion, an organization, gender etc.)

There is a distinction between free speech and hate speech. Any invocation of the term free speech may be referring to any one of a variety of meanings. One is the moral right to freedom of expression, a fundamental moral requirement that agents be free to express themselves and communicate with others [29]. Phrases like "I hate you man", "Good night asshole" etc., are commonly expressed between peers or friends and deemed as free speech. However, phrases like "I hate you disgusting, terrorist, piece of filth", "Good night asshole hope you get run over by a bus" [4][1] are a few examples of Hate speech. Free speech is a human right, however this should not be demeaned as hate speech. To have peace and harmony in the society, it is necessary for social media and other internet forums to filter offensive or hateful contents from the posts or replies and prompt actions against those involved. Otherwise it may lead to chaos, violence, communal riots etc., and also impacting business prospects of social media platforms and internet forums. For example, Sri Lanka has witnessed vigilance inspired by online spread of rumours, aimed at Tamil Muslims from a minority viewpoint. In March 2018, during the outbreak of violence, the Srilankan government had blocked Facebook and WhatsApp access; as well as Viber messaging app, for a week, stating that Facebook wasn't attentive or sensitive enough during Emergencies [14].

Another incident is, Parliamentary committees in Germany and the UK sharply criticized leading social media platforms such as Twitter, Facebook and Youtube (Google) in the spring of 2017 for failing to take appropriate and decisive action against hate speech, with the German government threatening social networks with a fine of up to EUR 50 million a year, if they keep failing to act on hateful posts (and posters) within a week [15].

Classification of hate speech is a difficult task because of the complexity of the natural language, sarcasm, emojis, typo error, micro words, ambiguity etc. There could be a chance of misclassifying non-hate speech as hate speech and erroneously implying non hate speakers to be hate speakers. It is important to accurately distinguish between them in view of the moral and legal consequences of hate speech.

The manual filtering of hateful contents is not reliable and scalable, as there are billions of messages posted everyday between billions of senders and receivers leading to huge data. This has motivated the researchers across the world to develop tools based on artificial intelligence and others to identify the hate speech contents. There are a lot of research works undertaken for automating this process using Natural Language Processing (NLP) and Machine Learning (ML) [1][2][3][4][18][19] with the focus on manual extraction of features and representation learning methods followed by classification using linear classifiers. However, some past works have shown that deep learning models perform better to learn the context with word embeddings[6][7][8][21]. Motivated with these observations from the past research works, we have proposed a model with a hybrid feature set for hate speech detection.

In our experiments, we have introduced the concept called Hybrid Features, which are learned from deep learning models and are used with other semantic features [9][25][28] like Parts of Speech (POS), Term Frequency - Inverse Document Frequency(TF-IDF), and tweet specific syntactic features for an effective feature representation. We have explored different kinds of neural networks for improving feature learning, starting with single architecture model like Convolution Neural Network (CNN), Long Short Term Memory (LSTM), Bi-directional LSTM (BiLSTM) and hybrid models like Convolution Neural Network + Long Short Term Memory (CNN+LSTM), Convolution Neural Network + Gated Recurrent Unit (CNN+GRU) and also focused on improving preprocessing steps to reduce the number of missing embeddings and to increase the vocabulary for efficient feature learning.

The major contributions of this paper are: 1) To show the significance of preprocessing techniques in obtaining feature sets. 2) Explore different deep learning models for fine tuning the word embeddings i.e. for feature learning. 3) Development of hybrid features set for an effective classification task. 4) Identify the better classifier for the given task at hand in terms of classification performance.

The paper is organized as follows. Section 2 describes the Related Work. In Section 3, we discuss the Methodology followed in the paper. Section 4 gives the details of the Experiments and Results. Finally, Conclusion is discussed in section 5.

2 Related Work

Hateful content detection is a bit of a challenge because of the complex nature of the natural language, humans can easily find the context and the structure of the language but machines will not be able to identify all the contexts, sarcasm behind the speech or text. A considerable amount of research has been done in the past, researchers have worked on abusive or hate speech to train the classifier to understand the context as much as possible and then identify the hate content. In this section, we discuss a few works that are closely related to our work.

For classification of texts, usage of supervised and machine learning methods is not novice any more. A lot of research works have been conducted in classification of texts. A supervised classification model is proposed by Davidson et al.[1], based on the state-of-the-art features incorporating TF-IDF, POS tags and other linguistic features using Support Vector Machines(SVM), Logistic Regression and Naive Bayes to differentiate between offensive and hate speech. They have used a twitter dataset of 25k tweets for this purpose. It is observed that they were the first to consider hate classification as a multiclass classification problem, rather than a binary classification problem of classifying the text as 'Hate' or 'Non Hate'. Nobata et al.[19] have proposed supervised, machine learning based methods for detecting abusive contents on online user comments by using n-grams, various linguistic, syntactic and semantic features in the comments. They have used Yahoo news and finance comments which have been annotated as abusive or clean, by trained Yahoo employees. Djuric et al.[18], have also classified hate speech on Yahoo Comments using continuous Bag-of-Words (BOW) neural language model and comment embeddings or paragraph2vec to learn distributed representations of comments and words. They have performed binary classification between the clean and hate comments using logistic regression.

Also, work was carried out in identifying hate speech with unsupervised learning as well. Natural Language Processing (NLP) techniques are employed in unsupervised learning strategy to detect offensive or hate messages or posts in texts. Chen et al.[2] have used NLP approaches for obtaining lexical syntactic features of offensive text. Warner and Hirschberg [3] used a template-based strategy, an unsupervised NLP-based algorithm, to generate features for detecting hate speech in web content. These approaches are not efficient as they fail to detect spelling or syntax issues caused unintentionally or intentionally by the users.

Waseem and Hovy [4] have also worked on the detection of hate speech. In their approach, they have used lexical features such as word-grams, character n-grams of different length along with other features like gender (demographic) related information, geographical features like location of the user. In their research, they have shown how demographic and geographic information of users has a marginal impact on the performance of hate detection. Further Waseem [5] has worked to expand the existing corpus by generating an annotated dataset for hate detection with the help of crowd-sourcing, and has shown how the knowledge of annotators has influenced the classification task.

Senarath and Purohit[25] have ensembled the various diverse features for hate speech detection on two popular Twitter Dataset using SVM. Salminen et al. [26] have addressed the lack of model for hate speech detection across multiple platforms by combining multi-platform data using different feature representations with several machine learning classifiers and neural networks.

Badjatiya et al.[6] explored deep neural networks for hate content classification, they have used word embedding technique called Global Vectors for Word Representation (GloVe) [11] with Deep Learning techniques for feature extraction and classified the tweets using machine learning methods. They have conducted experiments with 3 different neural networks such as CNN, LSTM and Fast Text for further fine tuning the word embeddings and these new features are given to the traditional classifiers like GBDT, SVM, Logistic Regression etc. They have reported quite a better F1 score than any other methods for 16k twitter dataset with the best model - LSTM, Gradient Boosting Decision Tree (GBDT) and Random Embedding. They were inspired from the works of Kim [20] that highlighted a series of experiments with CNN, trained on pre-trained word vectors for sentence classification.

Sikdar and Gamback[21] also used deep learning models to detect hate speech on Twitter. However, they have only experimented with CNN using character n-grams, word vectors and combination of both. Park and Fung[22] have also used CNN models with character and word level inputs. However, they have approached the hate speech classification problem as a two step classification task. They initially separated the abusive speech from the non-abusive speech and sub-classified the abusive tweets as racism or sexism and compared against one step multi class classification. Founta et al.[24] worked on detecting hate speech using different dataset and proposed a unified deep learning architecture, which uses a wide variety of metadata and combines it with automatically extracted hidden patterns within the tweets.

Zhang et al. [7] have done extensive experiments on different dataset available for hate or offensive speech using hybrid architecture CNN+GRU with word2vec embedding [13] . In their work, they have demonstrated the multi-step approach with the combination of deep neural networks in which CNN is used for feature extraction and GRU for learning this representation and dense layer for classification.

Rizos et al. [8] and Konstantin Hemker [10] in their work have proposed data augmentation to classify the short text effectively and to balance the class imbalance, where the data available for different dataset is not equal. When there is a class imbalance, it may lead to a biased classification for which more data is available and for other classes accuracy rate will be very low. They have addressed this issue by data augmentation. i.e. generating samples by methods like threshold based, pos-tagging, language generation models to increase the dataset. Then they have applied the multi-step Deep learning methods CNN+GRU(LSTM) followed by a dense layer with different embedding techniques like GloVe [11], word2vec [13] etc., for feature learning and classification. Yenala H et al. [9] have worked on detecting inappropriate text in web search queries and chat data using Deep Learning Techniques. They have used Convolution-BiLSTM and BiLSTM respectively for their experiments.

For any text classification, feature extraction is an important task, text may contain information that may or may not be needed. In social media posts or responses, there may be human typo errors, micro words, hashtags etc. Careful preprocessing and efficient feature extraction techniques are very important, making the machine to learn the context. Another important thing to consider is the ambiguity present in the language, i.e. the same word may express different meanings in different contexts, therefore simple word based methods will not perform well in classifying the texts.

We observed that from the previous works, NLP is one of the popular approaches employed for hate speech detection. However, the performances of these models are not very efficient as they fail to understand the hidden context in the text. Hence there is a scope to use deep learning techniques along with pre-trained word embeddings as they can further contribute to address this issue more efficiently and to extract features specific towards the task. However, in deep learning approaches, most of the existing works have focused on multi-step methods like CNN for feature extraction and LSTM or GRU for learning and classification. From the literature, we found that the combination of deep learning features along with semantic features have not been explored for feature set representation and subsequently for classification tasks. Also, ensemble of other deep learning models like BiLSTM and Hybrid models with machine learning were not explored in the past works. In this paper, we propose a model for hate speech detection using the hybrid features, that are learnt using deep learning methods along with the basic semantic features. Machine learning algorithms are used later for the process of classification.

3 Methodology

3.1 Dataset

For our experiments, we have used the hate speech dataset made available by ‘Wassem and Hovy’ [4]. It is the largest dataset on hate speech text that is publicly available and considered as the significant baseline paper by researchers[6][7][22][23][27]. This dataset contained a total of 16k (16907) tweets on hate speech categorized into Sexism , Racism and None. The corpus was collected manually over two months by searching the tweets with frequently occurring terms that contain hate speech and references to specific entities. It was then annotated through the crowd-sourcing of over 600 users[7]. They have provided only the IDs of the tweets and not the actual tweets.

We have used Twitter API1 to extract tweets from those IDs. From this process, we were able to obtain 10k tweets, out of which only 37 tweets were related to the racism category, which was found to be insufficient for the task of classification. Unfortunately, we were unable to scrap all these tweets, as many of the accounts were blocked by twitter. However, we were able to find the same dataset with both tweets and IDs at Github2. We have cross checked the tweet IDs by writing python programs, which confirmed it to be the same dataset as used by Waseem and Hovy [4]. We obtained almost 16k (16135) tweets from this Github account, a substantial number of tweets, to start the experimentation. The class distribution of the dataset used in Waseem and Hovy [4] and the dataset obtained from the Github account2 is shown in Table 1:

Table 1. Twitter Hate Speech Dataset

Class	**Number of Tweets (Waseem and Hovy [4] Dataset)**	**Number of Tweets (Our Dataset)**
Racism	1972	**1935**
Sexism	3383	**3167**
None	11552	**11033**
Total	16907	**16135**

3.2 Baseline Methods

We have used techniques described in Badjatiya et al. [6], current state of the art, as one of the baseline work for our experiment. They have used GloVe [11] for embedding technique, deep learning models LSTM and CNN for feature learning to fine tune the word embeddings. Machine Learning models are employed for classification with the fine tuned embeddings as features and they have reported better results till date. Also, we have referred to the work of Davidson et al. [1] as another baseline and have experimented with a 16k twitter dataset using semantic features like TF-IDF, POS and other linguistic features as described in the paper. In addition, we have compared our results with other works reported in the literature.

3.3 Proposed Methodology

3.3.1 Preprocessing Modifications

Preprocessing is a very important step in classification of texts, the need to obtain quality vocabulary from raw data and to understand the context is a complex process. Basic preprocessing steps have been applied to obtain the tokens suitable for GloVe [11] embeddings that includes:

- Tagging : Replacing user mentions, hashtags, numbers, emojis etc., with appropriate tags to convert those special texts in the tweets to meaningful words and to get the appropriate embeddings from the GloVe.
- Tokenize : Removing Stopwords, Punctuations and generating tokens.

During the course of the experiment, we observed that there were missing embeddings for many words, due to many improper and meaningless words, for which we could not find the embeddings from the GloVe. We started to analyze this situation and initiated a process to reduce the number of missing embeddings. The first step

[1] https://developer.twitter.com/en/docs.html

[2] https://github.com/amyhemmeter/AutomaticDetectionofHateSpeech/blob/master/hatespeech.csv

towards this is fine tuning the preprocessing step. This is done by proposing a few modifications to the Tagging step in the basic preprocessing. We have proposed the following modifications as part of preprocessing:

1. Hashtag Splitting (HS): We have included this as part of preprocessing using wordsegment(WS) during the Tagging process. For Example, in the dataset, we have hashtags like ”#fingersinthemouth”, ”#committokitchentoo” , ”#nosexist ” etc., were not segmented in the basic preprocessing and resulted in missing many meaningful word embeddings. After applying HS, hashtags in the tweets will be split into meaningful words. For example, hashtags like “#fingersinthemouth” was translated to ”fingers in the mouth”, making it meaningful and find the corresponding embeddings to enable the machine to understand the vocabulary properly. This has reduced the number of missing embeddings significantly from 2547 to 1857 (27.1%) and effectively uses them for the task at hand.

2. Space for all tags: We have observed that there are missing spaces between words and special texts like emojis, user mentions, numbers etc. In many tweets, when these special texts were replaced with appropriate tags in the preprocessing steps, words and tags were getting combined, leading to misinterpretation of words.

For example phrases like, ”3000years old” is preprocessed as ”numberyears old” (3000 is a number and replaced with the tag <number>) using the basic preprocessing, which does not give the proper meaning and leads to the missing embedding for ”numberyears” in the GloVe vectors. After introducing our proposed preprocessing modification technique, we obtain the following phrase ”number years old”, providing 3 meaningful words for which we can find the embeddings from the GloVe and this subsequently improves the vocabulary for better understanding. There are many such instances in the missing embedding words list, after identifying these instances, spaces are added before and after the tags to avoid the misinterpretation of words. When this step is used with hashtag splitting, the number of missing embeddings is further reduced to 1445 from 1857 (i.e. a total of 1102 words found out of 2547 missing embeddings, 43%) and thereby increasing meaningful words in the vocabulary.

3. Random Spacing: Further, we carried out the experiment with addition of spaces to a few specific tags. For instance: numbers, capital letter words, emojis are common in tweets. We considered addition of spaces between words that are numbers, capital letter words and emojis. With these additions along with hashtag splitting, we obtained better results.

3.3.2 Deep Learning Models for Deep Learning Features Extraction

We have learnt that the vocabulary and feature learning matters a lot for efficient classification from the baseline methods. We have incorporated deep learning models for feature learning as discussed in Badjatiya et al. [6], this ensemble concept with machine learning has reported the best result. However, we have further extended and explored the different deep learning models like BiLSTM, Hybrid Models other than CNN and LSTM for fine tuning the word embeddings to learn features specific towards the task, as they were not explored in the previous works. The different deep learning models applied are:

a. CNN, LSTM: We have used the same experimental settings as described in [6].

b. BiLSTM : inspired by Yanala H et al.[9], as it learns the input context in two directions, forward and backward, then the information is combined from both the ends resulting in the improved feature learning. The experimental settings employed were the same as that of LSTM, as the results obtained were found to be better with these settings. We also did the experiments with variations of BiLSTM models using different hyper parameters and various combinations of layers like global max pooling layer, dense layer instead of dropout layer before the output layer , the results obtained were not that encouraging.

c. Hybrid Models: We have found that various hybrid models [7][8][10] comprising CNN+GRU, CNN+LSTM have been employed for hate speech detection. We also have explored the hybrid models for deep learning feature extraction in identifying the hate speech. It is a multi-step approach, CNN was used for feature extraction and these extracted features were fed to GRU (LSTM) for fine tuning the word vectors to obtain Deep Learning Features. However, the results obtained from the hybrid models did not perform better than the existing state of the art approach.

3.3.3 Proposed Novel Hybrid Features Generation Technique

After obtaining features from deep learning models, we enhanced them with semantic features like TF-IDF, POS and other tweet specific syntactic features to generate hybrid features before applying it to classifiers. We have proposed the following novel approaches for generating hybrid features set:

1. TF-IDF Weighted Addition (TF-IDF_WA): In this approach, TF-IDF Features are generated for the dataset and for each word, the fine tuned vector values learnt from the deep learning models are added with the corresponding TF-IDF score. The dimension of word vectors remains unchanged, only the fine tuned vectors will get supercharged with TF-IDF score. Addition of TF-IDF score increases the weightage of each word as it is learnt based on the context and helps to improve the performance. Algorithm for TF-IDF Weighted Addition for hybrid features generation is as follows:

Algorithm 1. TF-IDF Weighted Addition to generate hybrid features.

Step1: Get the fine tuned embeddings (deep learning features) from the saved model, deep_trained_embedding.

Step2: Extract TF-IDF features with unigram for the preprocessed dataset, tfidf_feat.

Step3: Calculate Sentence Vector of the tweet. For each word in the preprocessed tweet if it is found in deep_trained_embedding, get the fine tuned vector and add it with its corresponding TF-IDF score and sum up all these new word vectors. Also sum up the tfidf scores of all the found words.

$$sent\_vec += deep\_trained\_embedding[word] + tfidf\_feat[word] \quad (1)$$

$$weight\_sum += tfidf\_feat[word] \quad (2)$$

Step4: Divide the sentence vector by sum of TF-IDF scores of all the words in a tweet to get the final sent_vec value.

$$sent\_vec /= weight\_sum \quad (3)$$

Step5: Repeat the Steps 3 and 4 for each tweet in the dataset and append them to form an array of hybrid features.

2. Concatenation of Features (CF): In this approach, features like TF-IDF, POS with ngrams (bi-grams, tri-grams) and tweet specific syntactic features such as number of URLs, User mentions, Hashtags, Sentiments etc., are extracted for the dataset and concatenated with the deep learning features. It creates the feature set of larger dimension and thus the effective feature representation contributes towards the performance improvement. Algorithm for Concatenation of Features for hybrid features generation is as follows:

Algorithm 2. Concatenation of Features (CF) to generate hybrid features.

Step1: Get the tuned embeddings (deep learning features) from the saved model, deep_trained_embedding.

Step2: For each word in the preprocessed tweet if it is present in the saved model get the fine tuned word vector and sum up all these word vectors.

$$sent\_vec += deep\_trained\_embedding[word] \quad (4)$$

Step3: Repeat the Step 2 for all the tweets in the dataset and append them in an array to form tweet vectors, tweet_vec.

Step4: Extract TF-IDF, POS features with ngrams (1-3gram) and syntactic features from the dataset; tfidf_feat, pos_feat, syntactic_feat.

Step5: Concatenate the features obtained from Step3 and Step4 to form hybrid feature set.

$$hybrid\_feat = concat(tweet\_vec, tfidf\_feat, pos\_feat, syntactic\_feat)$$

3.3.4 Classifiers

We have used three different machine learning classifiers namely multinomial Logistic Regression, Gradient Boosting Decision Tree (GBDT), Support Vector Machine (SVM) with class_weight set as balanced with both linear and rbf kernel and having all other parameters as default. These classifiers were found to be effective from the previous works[1][6][18]. The two sets of hybrid features are fed into the classifiers and trained them using cross validation.

In brief, algorithm followed for an effective classification of hate message is described as below:

Step1: Apply the proposed preprocessing steps to reduce the missing embeddings as discussed in section 3.3.1.

Step2: Initialize the network weights with GolVe [11] embeddings and Random embeddings of GloVe.

Step3: Train the different deep learning models described in section 3.3.2 to learn the task specific word embeddings fine tuned to hate speech labels i.e. to obtain Deep Learning Features.

Step4: Extract semantic features which includes TF-IDF and POS and other tweet specific syntactic features.

Step5: Use features obtained from step 4 and 5 with novel methods mentioned in section 3.3.3 to generate hybrid features.

Step6: Train and test traditional machine learning classifiers described in section 3.3.4 using hybrid features with k-fold cross validation.

4 Experiments and Results

An optimizer called 'Adam' and a batch size = 128 have been set for all the deep learning methods. We tested Adam optimizer with different learning rates and found that it provides better results with default values (lr=0.001, beta 1=0.9, beta 2=0.999, epsilon=1e-08, decay=0.0). We also conducted the experiment using 'RMSProp' optimizer with LSTM, the results obtained were not encouraging and did not proceed with it. We observed that the Hybrid models (CNN+GRU, CNN+LSTM) performed well at kernel size=3, pool length=4, convolution filters =100 with dropout layer after LSTM (GRU), instead of global max pooling and dense layer.

GloVe, a pre-trained model, is used as the embedding based method. In the Random Embedding method, GloVe vectors are shuffled and assigned the values randomly. GloVe word embedding model have been trained on 2 billion (includes 27 billion tokens and 1.2 million vocabulary which is uncased) tweets. We implemented the methods discussed in state of the art [6] by taking different embedding dimension sizes. The results obtained were very close to those discussed in the literature using CNN with GBDT and LSTM with GBDT methods for the embedding dimension size of 200. Hence, we continued the subsequent experiments with different methods for embedding dimension size of 200. We have used 10 fold Cross Validation and 10 epochs for conducting the experiments. Weighted Precision, Recall and F1-score is used as the performance metrics similar to works reported in the literature [1][6].

The results obtained are compared with the results reported in the literature and the same is shown in Table 2. We have also tried applying Logistic Regression (LR) to the best performing method (LSTM + Random Embedding + GBDT) as reported in the literature [6], forming a new method LSTM + Random Embedding + LR and found the results of this method to be better by 1% in F1 score.

Table 2. Reproduction and Comparison of Results - Baseline Methods

Literature	Method	Paper Results			Our Implemented Results		
		Prec	**Recall**	**F1**	**Prec**	**Recall**	**F1**
Part A : Baselines Badjatiya et al.[6]	TFIDF + Balanced SVM	0.816	0.816	0.816	0.815	0.819	0.815
	TFIDF + GBDT	0.819	0.807	0.813	0.827	0.818	0.802
	BoWV + Balanced SVM	0.791	0.788	0.789	0.757	0.688	0.701
	BoWV + GBDT	0.800	0.802	0.801	0.765	0.774	0.758
Part B : DNN Only Badjatiya et al.[6]	CNN + Random Embedding	0.813	0.816	0.814	0.815	0.815	0.814
	CNN + GloVe	0.839	0.840	0.839	0.831	0.832	0.830
	LSTM + Random Embedding	0.805	0.804	0.804	0.798	0.792	0.795
	LSTM + GloVe	0.807	0.809	0.808	0.826	0.825	0.825
Part C : DNN + GBDT Badjatiya et al.[6]	CNN + GloVe + GBDT	0.864	0.864	0.864	0.857	0.857	0.851
	CNN + Random Embedding + GBDT	0.864	0.864	0.864	0.902	0.901	0.899
	LSTM + GloVe + GBDT	0.849	0.848	0.848	0.858	0.857	0.851
	LSTM + Random Embedding + GBDT	**0.930**	**0.930**	**0.930**	**0.920**	**0.920**	**0.919**
	LSTM + Random Embedding + LR	#	#	#	**0.929**	**0.930**	**0.929**
Davidson et al.[1]	Baseline Features + LR	0.910	0.890	0.900	0.820	0.820	0.820
Extension of [6] inspired by [7,8,9,10]	**BiLSTM + Random Embedding + LR**	#	#	#	**0.929**	**0.929**	**0.929**
	CNN+LSTM + Random Embedding + LR	#	#	#	0.908	0.908	0.907
	CNN+GRU + Random Embedding + LR	#	#	#	0.909	0.910	0.908

#indicates that the paper results are not available [we have conducted these experiments exclusively in our work]

The experiments were continued to explore other deep learning models like BiLSTM and hybrid models such as CNN+LSTM, CNN+GRU inspired by the works reported in the literature [7][8][9][10] with both GloVe and random embeddings for feature learning and applied ensemble methods with LR. Table 2 shows the results obtained for BiLSTM and hybrid models using Random Embeddings, as they are found to provide better results compared to GloVe embeddings, along with other methods. We observed that the results obtained for BiLSTM with random embeddings to be better and considered it as one of the potential models for feature generations. In addition, we also implemented the methods discussed in [1] and obtained 82% F1 score with cross validation, the results obtained are shown in Table 2.

Instead of using features or fine tuned word vectors learnt from deep learning models directly to the machine learning classifier, we continued the experiments with our novel proposed algorithms TF-IDF_WA and CF for generating hybrid features. We initiated the experiments by applying basic preprocessing methods with hybrid features. The results obtained using two different algorithms are as shown in Table 3. From the results, we observe that the hybrid features set has improved the classification accuracy due to enhanced feature representation. We continued the experiment further by applying the proposed preprocessing modifications with hybrid features. The results obtained are shown in Table 4. We observe that the results have improved, due to reduction in the number of missing embeddings.

Table 3. Results of Different Methods with Basic Preprocessing Techniques and Hybrid Feature

Methods with Basic Preprocessing	Prec	Recall	F1
LSTM + Random Embedding + TF-IDF_WA + LR	**0.935**	**0.936**	**0.935**
LSTM + Glove + TF-IDF_WA + LR	0.906	0.907	0.906
CNN + Random Embedding + TF-IDF_WA + LR	0.925	0.926	0.925
CNN + Glove + TF-IDF_WA + LR	0.899	0.900	0.899
BiLSTM + Random Embedding + TF-IDF_WA + LR	**0.935**	**0.936**	**0.935**
BiLSTM + Glove + TF-IDF_WA + LR	0.910	0.910	0.910
LSTM + Random Embedding + CF + LR	**0.941**	**0.941**	**0.940**
LSTM + Glove + CF + LR	0.910	0.911	0.909
CNN + Random Embedding + CF + LR	0.932	0.932	0.931
CNN + Glove + CF + LR	0.904	0.905	0.903
BiLSTM + Random Embedding + CF + LR	**0.938**	**0.939**	**0.938**
BiLSTM + Glove + CF + LR	0.910	0.911	0.910

Table 4. Results of Different Methods with Proposed Preprocessing Techniques and Hybrid Features

Proposed Preprocessing	Methods	Prec	Recall	F1
Hashtag Splitting (HS)	**LSTM + Random Embedding + TF-IDF_WA + LR**	**0.936**	**0.937**	**0.936**
	LSTM + Glove + TF-IDF_WA + LR	0.907	0.908	0.907
	CNN + Random Embedding + TF-IDF_WA + LR	0.924	0.925	0.924
	CNN + Glove + TF-IDF_WA + LR	0.903	0.904	0.903
	BiLSTM + Random Embedding + TF-IDF_WA + LR	**0.939**	**0.939**	**0.939**
	BiLSTM + Glove + TF-IDF_WA + LR	0.914	0.915	0.915
	LSTM + Random Embedding + CF + LR	**0.943**	**0.943**	**0.943**
	LSTM + Glove + CF + LR	0.911	0.912	0.911
	CNN + Random Embedding + CF + LR	0.930	0.930	0.929
	CNN + Glove + CF + LR	0.905	0.906	0.905
	BiLSTM + Random Embedding + CF + LR	**0.943**	**0.943**	**0.942**
	BiLSTM + Glove + CF + LR	0.917	0.917	0.916
HS + Space for all tags	LSTM + Random Embedding + TF-IDF_WA + LR	0.939	0.939	0.938
	BiLSTM + Random Embedding + TF-IDF_WA + LR	0.938	0.938	0.938
	LSTM + Random Embedding + CF + LR	**0.944**	**0.944**	**0.943**
	BiLSTM + Random Embedding + CF + LR	**0.941**	**0.941**	**0.940**
HS + Random Spacing	LSTM + Random Embedding + TF-IDF_WA + LR	0.942	0.942	0.942
	BiLSTM + Random Embedding + TF-IDF_WA + LR	0.939	0.940	0.939
	LSTM + Random Embedding + CF + LR	**0.945**	**0.945**	**0.945**
	BiLSTM + Random Embedding + CF + LR	**0.944**	**0.944**	**0.944**

We also conducted the experiments with hybrid models: CNN+GRU(LSTM) using basic preprocessing technique and the proposed Hashtag Splitting preprocessing technique. The results obtained for the hybrid models are shown in Table 5. These models did not perform well as they failed to extract deep learning features effectively as compared to the earlier discussed single architecture deep learning models.

Table 5. Results of Hybrid Models with Hybrid Features using both Basic and Proposed Preprocessing Techniques

Basic Preprocessing	**Prec**	**Recall**	**F1**
CNN+GRU + Random Embedding + TF-IDF_WA + LR	0.916	0.917	0.916
CNN+LSTM + Random Embedding + TF-IDF_WA + LR	0.914	0.914	0.914
CNN+GRU + Random Embedding + CF + LR	0.917	0.918	0.916
CNN+LSTM + Random Embedding + CF + LR	0.918	0.919	0.918
Proposed Preprocessing (Hashtag Splitting)			
CNN+GRU + Random Embedding + TF-IDF_WA + LR	0.920	0.921	0.920
CNN+LSTM + Random Embedding + TF-IDF_WA + LR	0.920	0.920	0.920
CNN+GRU + Random Embedding+ CF + LR	0.922	0.922	0.921
CNN+LSTM + Random Embedding + CF + LR	0.922	0.922	0.921

We measured our system evaluation using a weighted average of precision, recall and f1- score across 10 fold cross validation and the results obtained are compared with the best results reported in the literature [6]. We have conducted experiments with different classifiers and have reported the result of the LR, as it provided the better results amongst other classifiers. We found from the experiments that the LR with hybrid features performed better compared to GBDT and SVM. The performance of the classifier is purely dependent on the data and Linear classifiers such as LR tend to perform very well on extremely sparse datasets [17].

We have achieved the better results with our proposed hybrid features set approach. Also, the aforementioned preprocessing techniques have further improved the accuracy as shown in Table 3 and 4. Our proposed preprocessing techniques and hybrid features outperforms the best methods discussed in literature. We found that the "LSTM + Random Embedding + CF + Logistic Regression" to be the best model providing better results with both basic and proposed preprocessing methods. Our best model with proposed preprocessing Hashtag Splitting technique has outperformed the state of the art [6] with a significant improvement of 1.3~ in f1-score (94.3%). We also observed that Random Spacing preprocessing technique has increased the performance of our best model by 0.2% in f1-score (94.5%). Also, the performance of another proposed neural network BiLSTM with hybrid features is found to be very similar to that of LSTM, the results of BiLSTM were found varying between 0.1% to 0.3% compared to LSTM in terms of f1-score. It is clearly evident from the results reported in the aforementioned tables and the same is shown in Figure 1. It highlights the performance of LSTM and BiLSTM with random embedding, hybrid features and classifier LR with both basic and proposed preprocessing Hashtag Splitting (HS) techniques. In addition, we have compared our results with the other works reported in literature in terms of F1 score, our best performing model (LSTM + Random Embedding + CF + Logistic Regression) is found to be better than the others reported in literature as shown in Figure 2.

Our best performing model is able to detect the hate speech effectively by gaining the knowledge from the hybrid features set. For example, tweets like: "To be honest I like the idea of the wife cooking and cleaning and the guy earning all the money to support their family, ?", This tweet is already annotated as hate speech and our model was successful in classifying it as hate (sexism), even though there are no swear words in the tweet. This is due to the features it has already learnt. There are many tweets related to women or girls and kitchen in the data set, we infer that by using the knowledge or context (wife = woman, cooking= kitchen) and the bi and tri-gram in the TF-IDF features i.e. using hybrid features, our model was able to correctly classify it as hate speech. Also the hashtag splitting has played a major role in effective classification of hate speech. As discussed earlier, hashtag "#committokitchentoo" is split properly for obtaining an effective feature set. There are many instances in the tweets that targeted women using feminine words like daughter, sister, children (I know...Julianne Moore's pudgy little sister is severely overvaluing herself here...) and communities like Muslims, Jews, Islam etc., are correctly classified with the knowledge gained by the classifier using hybrid features.

There are instances where our model was not able to identify hate speech correctly as the hate words were rarely used and not associated or occurred with other frequently used hate words. In this case, the training data was insufficient and proper classification of hate tweets was not possible. For example words like Taquiyya, Imam, Kurds, Hamas, Zionist, Caliphatalism etc., are used rarely and tweets containing these and similar words were misclassified as non-hate by our best performing model. The dataset used in the experiments, baseline methods, our methods with basic preprocessing and hybrid features technique TF-IDF_WA are made available at this Github account3 and all other remaining preprocessing and hybrid features techniques will be made available in due course for the benefit of researchers working on hate speech detection.

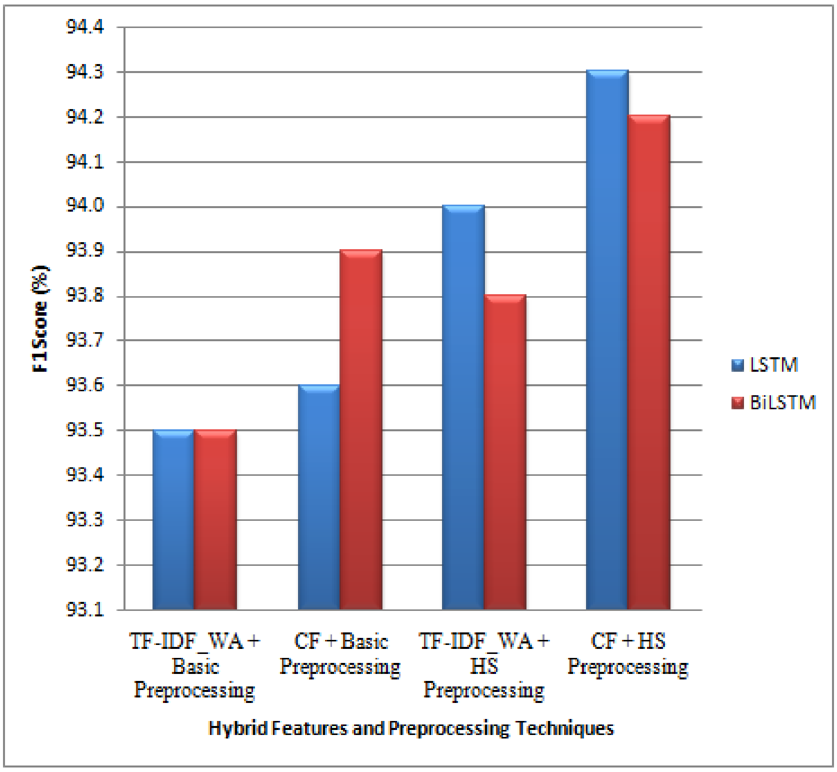

Figure 1. Comparison of F1 Score of LSTM and BiLSTM + Random Embedding + Hybrid Features + LR Classifier with Basic and Proposed Preprocessing (HS).

[3] https://github.com/ShruthiPrabhu29/hateSpeechDetectionTwitter

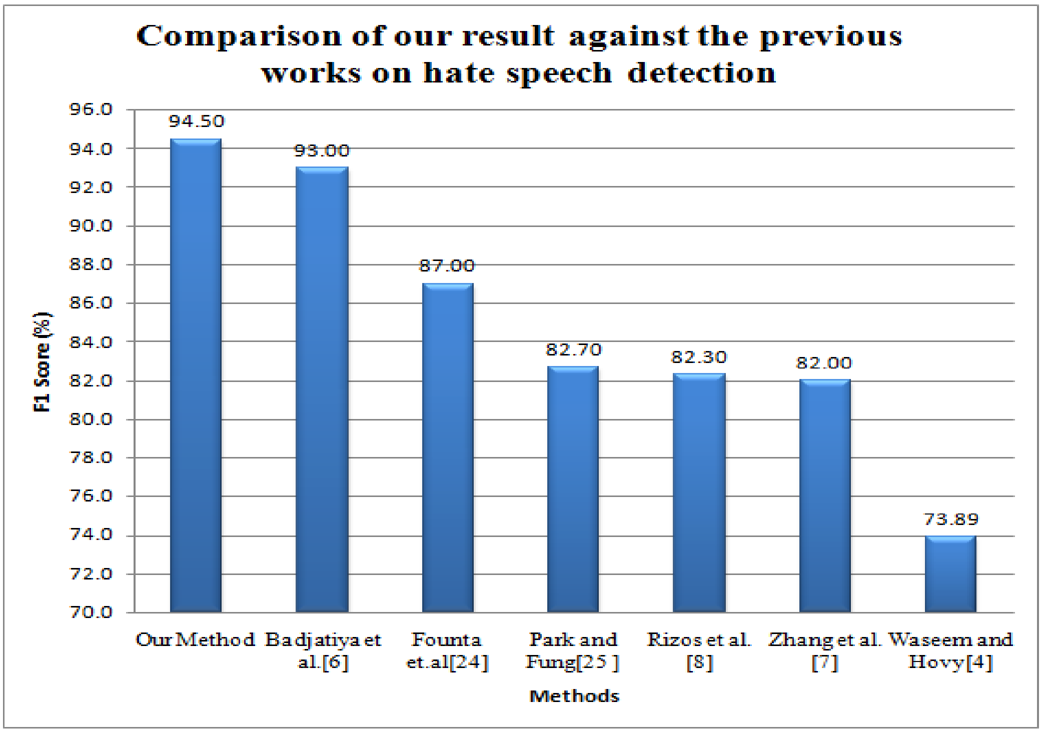

Figure 2. Comparison of Our Method with Others in the Literature.

5 Conclusion

In this paper we have presented the Deep Learning methods ensembled with Machine learning classifier for hate speech detection using hybrid features on social media platforms. We have experimented with different deep learning models starting with single architecture models to hybrid models for deep learning features extraction. These features are combined with the semantic features to form hybrid features using our novel approach and are applied to simple traditional classifiers for training and prediction purposes. Our experiments highlight the importance of data preprocessing to improve the vocabulary for feature learning and for better understanding the context (i.e. reducing the missing embeddings) to provide better results.

Our proposed preprocessing techniques like hashtag splitting and adding space for tags have reduced the missing embeddings from 2547 to 1445 (43%) and have increased the vocabulary for the feature set. We have explored the different deep learning methods like CNN, LSTM, BiLSTM and Hybrid Models for feature learning to fine tune the word vectors. We have developed two approaches for generating hybrid features and have achieved effective classification with a significant increase in F1 score by 1.0% - 1.5% (94% - 94.5%). LSTM + Random Embedding + CF + LR is found to be the best model and the results obtained are better compared to the state of the art. Also, amongst the experimented Machine Learning classifiers, we found Logistic Regression with hybrid features as the best classifier for hate speech detection.

References

[1] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” *arXiv preprint arXiv:1703.04009*, 2017.

[2] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, " Detecting offensive language in social media to protect adolescent online safety ", *in 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing. IEEE*, 2012, pp. 71–80.

[3] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web", *in Proceedings of the second workshop on language in social media. Association for Computational Linguistics, 2012*, pp. 19–26.

[4] Z. Waseem and D. Hovy, “Hateful symbols or hateful people? predictive features for hate speech detection on twitter,” *in Proceedings of the NAACL student research workshop, 2016*, pp. 88–93.

[5] Z. Waseem, “Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter,” *in Proceedings of the first workshop on NLP and computational social science, 2016,* pp. 138–142.

[6] P. Badjatiya, S. Gupta, M. Gupta, and 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.

[7] Z. Zhang, D. Robinson, and J. Tepper, “Detecting hate speech on twitter using a convolution-gru based deep neural network,” *in European Semantic Web Conference*, pp. 745–760, Springer, 2018.

[8] Georgios Rizos , Konstantin Hemker, Björn Schuller, "Augment to Prevent: Short-Text Data Augmentation in Deep Learning for Hate-Speech Classification," *in Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 991-1000. doi: https://doi.org/10.1145/3357384.3358040.

[9] Yenala, H., Jhanwar, A., Chinnakotla, M.K. et al. "Deep learning for detecting inappropriate content in text". Int J Data Sci Anal 6, 273–286 (2018). doi: https://doi.org/10.1007/s41060-017-0088-4 .

[10] Konstantin Hemker, "**Data Augmentation and Deep Learning for Hate Speech Detection**" IMPERIAL COLLEGE LONDON , Sept. 2018.

[11] J. Pennington, R. Socher, and C. D. Manning. “GloVe: Global Vectors for Word Representation,” *In EMNLP*, volume 14,2014, pp. 1532-43.

[12] Chollet, Franccois, and others, 2015. “Keras.” https://keras.io

[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient estimation of word representations in vector space,” *in Proceedings of Workshop at ICLR*, 2013, arXiv preprint arXiv:1301.3781.

[14] Zachary Laub, Jun 7th 2019, “Hate Speech on Social Media : Global Comparisons”**,** https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons.[Accessed: 2020-06-13]

[15] Emma Thomasson. 2017. “German cabinet agrees to fine social media over hate speech,” Reuters, Apr 5. http://uk.reuters.com/article/idUKKBN1771FK. [Accessed: 2020-06-13]

[16] Simon Kemp, Jan 30th 2020, “Digital 2020: Global Digital Overview”, https://datareportal.com/reports/digital-2020-global-digital-overview. [Accessed: 2020-06-13]

[17] Rich Caruana , Nikos Karampatziakis, Ainur Yessenalina, “An Empirical Evaluation of Supervised Learning in High Dimensions,” *in Proceedings of the 25th International Conference on Machine Learning*, 2008, pp.96-103.

[18] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate speech detection with comment embeddings,” *in Proceedings of the 24th international conference on world wide web, ACM*, 2015, pp. 29–30.

[19] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, “Abusive language detection in online user content,” *in Proceedings of the 25th international conference on world wide web. International World Wide Web Conferences Steering Committee*, 2016, pp. 145–153.

[20] Y. Kim, “Convolutional Neural Networks for Sentence Classification, ” *in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746-1751.

[21] Gamb¨ack and U. K. Sikdar, “Using convolutional neural networks to classify hate-speech,” *in Proceedings of the first workshop on abusive language online*, 2017, pp. 85–90.

[22] Anna Schmidt and Michael Wiegand, “A Survey on Hate Speech Detection using Natural Language Processing,” *in Proceedings of the 5th International Workshop on Natural Language Processing for Social Media*, 2017, pp. 1–10.

[23] Ji Ho Park and Pascale Fung, “One-step and two-step classification for abusive language detection on twitter,” arXiv preprint arXiv:1706.01206, 2017B.

[24] Antigoni-Maria Founta, , Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis, “A Unified Deep Learning Architecture for Abuse Detection,” 2018.

[25] Y. Senarath and H. Purohit, "Evaluating Semantic Feature Representations to Efficiently Detect Hate Intent on Social Media," *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, San Diego, CA, USA, 2020, pp. 199-202, doi: 10.1109/ICSC.2020.00041.

[26] Salminen, J., Hopf, M., Chowdhury, S.A. *et al.* "Developing an online hate classifier for multiple social media platforms, " *Hum. Cent. Comput. Inf. Sci.* 2020. https://doi.org/10.1186/s13673-019-0205-6.

[27] Paula Fortuna and Sérgio Nunes. 2018. "A Survey on Automatic Detection of Hate Speech in Text," ACM Comput. Surv. 51, 4, Article 85 (2018), 30 pages. https://doi.org/10.1145/3232676.

[28] Peter D Tureney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews", *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, pp.417–424.

[29] Scanlon T.M. "Comment on Shiffrin's thinker-based approach to freedom of speech," Const. Comment. 27, 2011, pp. 327–35

[30] Glazzard, Jonathan & Stones, Samuel. "Social Media and Young People's Mental Health," 2019. 10.5772/intechopen.88569.

[31] Jacobs, J. B., and Potter, K. "Hate crimes: Criminal Law and Identity Politics." Oxford University Press, 2000.

[32] Walker, S. "Hate Speech: The History of an American Controversy." U of Nebraska Press, 1994.