Computer Speech and Language 74 (2022) 101365

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Hate speech detection on Twitter using transfer learning

Raza Ali [a,*], Umar Farooq [a], Umair Arshad [a], Waseem Shahzad [a], Mirza Omer Beg [a]

[a] National University of Computer and Emerging Sciences, Foundation for the Advancement of Science and Technology, 3 A.K. Brohi Road, H-11/4, Islamabad, 46000, Pakistan

ARTICLE INFO

Keywords:
Deep learning
Transfer learning
Hate speech
Social Media
Machine Learning

ABSTRACT

Social Media has become an ultimate driver of social change in the global society. Implications of the events, that take place in one corner of the word, reverberate across the globe in various geographies. This is so because the huge amount of data generated on these platforms, reaches the far corners of the world in the blink of an eye. Developers of these platforms are facing numerous challenges to keep cyber space as inclusive and healthy as possible. However, in recent years, the phenomena of offensive speech and hate speech have risen their ugly heads. Despite manual efforts, the scope of this problem is so immense that it cannot be tackled by using concerted teams. In fact, there is a need that an automated technique is designed that detects and removes offensive and hateful comments before the materialization of their harmful impacts. In this research work, we develop an Urdu language hate lexicon, on the basis of this lexicon we formulate annotated dataset of 10,526 Urdu tweets. Furthermore, as baseline experiments, we use various machine learning techniques for hate speech detection. In addition, we use transfer learning to exploit pre-trained FastText Urdu word embeddings and multi-lingual BERT embeddings for our task. Finally, we experiment with four different variants of BERT to exploit transfer learning, and we show that BERT, xlm-roberta and distil-Bert are able to achieve encouraging F1-scores of 0.68, 0.68 and 0.69 respectively, on our multi class classification task. All these models exhibited success to varying degree but outperform a number of deep learning and machine learning baseline models.

1. Introduction

With the exponential growth of users on Social Media (SM) platforms, information transmission has increased manifold and access to updated information is now just a click away from the users. Users use these platforms not just for social interaction but also for recreation and information searching (Sajjad et al., 2019). As is the case with emerging technological platforms, there is a beneficial side to this widespread usage of SM and also negative phenomena that have sprung up its face in recent years. Users of these democratic platforms often indulge in behaviour that is often detrimental, offensive and sometimes outright hateful to numerous segments of society. The problem with such content on social media is that it promotes violent crimes in the society, therefore their early prevention is critical (Waseem and Hovy, 2016). Since the size of data is growing exponentially, therefore it is almost impossible to filter out such aberrant speech manually. In addition, research has shown that manual annotation and filtering of such data has led to post-traumatic stress disorder-like symptoms in people (Zampieri et al., 2019). This situation merits an automated Hate speech detection mechanism that detects and blocks such behaviors that are derogatory and have the potential to inspire violent crimes in

* Corresponding author.
E-mail address: i191231@nu.edu.pk (R. Ali).

https://doi.org/10.1016/j.csl.2022.101365
Received 25 June 2021; Received in revised form 30 November 2021; Accepted 10 February 2022
Available online 16 February 2022

society. Moreover, this problem becomes quite daunting when it comes to low-resource regional languages such as the Urdu language, widely used in the South Asian region. To the best of our research information there is a lexicon of hate words specified in Roman Urdu (Rizwan et al., 2020) but no such resource exists in the Urdu language for research advancement purposes. On the basis of these words we gather dataset and design an automated system to filter Hate Speech in the Urdu language from Twitter – a widely used SM platform, this way such behaviors would be kept in check Fig. 1.

Scholars have conflicting views as to what constitutes hate speech but in essence, there is a consensus that it is a speech that targets marginalized and disadvantaged social segments in a manner that is harmful to them (Davidson et al., 2021). More- over, researchers have also found that there is a strong connection between hate speech and violent crimes.[1] In this context, a considerable amount of work has been done in resource rich language such as English (Waseem and Hovy, 2016) (Sajjad et al., 2019) (Benito et al., 2019). Apart from the English language, there has been some work done in Spanish (Winter and Kern, 2019) (Perelló et al., 2019) (Baruah et al., 2019). But when it comes to low-resource regional languages, the research in Hate speech detec- tion is almost insignificant. Although there is some work that has been done in Arabic language (Aljarah et al., 2020) (Chowdhury et al., 2019) and in Indonesian language for abusive language detection as well (Ibrohim and Budi, 2018). The problem that the research community is currently facing is that the research work is getting confined to a few languages and this is resulting in the emergence of biaseness in technology. To cope with this problem it is essential that low-resource languages are also given equivalent research focus. Unfortunately, multi-lingual platforms like twitter have become a breeding ground for unchecked and unhindered hate speech. The effort to devise an automated mechanism to check for such speech is the basic motivation for this research work.

Urdu is a widely spoken language in South East Asia with over 300 million speakers all over the world. It is the national language of Pakistan and it is also spoken and understood in the neighbouring countries of India and Afghanistan. The calligraphic hand used to write Urdu as well as the Persian language is known as Nastaliq (نستعلیق خط). In the context of Natural language Processing (NLP), urdu offers a unique challenge due to its very complex writing style and richness in the morphological script. Unlike other languages like English, Spanish, French etc. Urdu is written from right to left, it is context-sensitive and supports diacritics (M.P. Akhter et al., 2020). It has no capitalization of words and it is a free word order language. Meaning that it is not dependant on the syntactic con- stituents to convey grammatical information.

For the detection of hate speech in the Urdu language, there are a few challenges. For Example, consider the words, black (کالا), Niazi (نیازی), Pathan (پٹھان), etc. are words that are commonly used in urdu to refer to specific colour or caste, but the gathered data has shown that the same words can be used to express hate speech as well 2. Similarly, some of the words like Non-believer (کافر), Sunni (سنی), Shia (شیعہ), etc. are used commonly in urdu language as well as more particularly when hate speech is expressed. Apart from that, generally speaking in the Urdu language it is hard to obtain appropriate tokens. The reason is in English we can tokenize text based on the spaces, as each word is seperated by a space for example, "I am not a traitor" can be tokenized into "I", "am", "not", "a", "traitor". However, when we write it in Urdu, میں وطن میں فروش نہیں ہوں. and then we tokenize it we get the following tokens میں, وطن, فروش, نہیں, ہوں. So we can clearly see that when we tokenize in Urdu, necessary compound words information is lost. We explain at the beginning of the paragraph that some commonly used words in the Urdu language that are not necessarily related to hate speech such as black, Niazi and Pathan can be used to hurl hateful commnets if used in a certain context. Appropriate tokenization of the words would lead to better information retrieval by the model from a sentence and subsequently accurate detection of Hate speech. Therefore, in this context it is our understanding that tokenization, although a problem of language processing, has bearing on our detection problem. Moreover, the current tools that are available for data pre-processing and cleaning are not tailored to deal with other languages such as Urdu, therefore numerous challenges would be encountered and solved while working in this area. Some sample tweets are shown in table 2 Given this context, we make the following contributions.

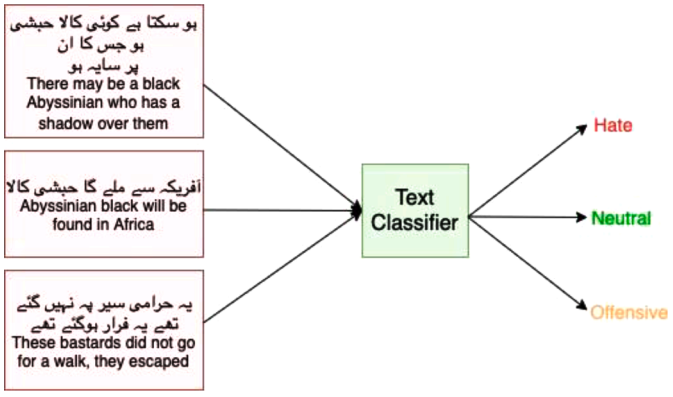

Fig. 1. Problem Diagram Depicting that a Certain Tweet Contains Information that a Classifier can Learn to Distinguish it into Different Category.

[1] www.nohatespeechmovement. org/hate-speech-watch/focus/ consequences-of-hate- speech,

- First, we specify a hate base lexicon of 83 unique words in the Urdu language.
- Second, we provide a carefully crafted dataset of 10,526 tweets called Urdu Hate Speech and Offensive language detection dataset.
- Third, we make use of numerous pre-trained embedding models, some of which are trained on Multi-lingual data while in some cases we explore transfer learning by carrying out thorough experimentation.
- Fourth, we present an ablation study in 2 different settings, making use of BERT in its vanilla form and training network by freezing and unfreezing BERT embeddings making use of transfer learning.
- In second setting of ablation part we modify vanilla BERT into BERT- CNN architecture by making use of CNN layers on top of BERT frozen and unfrozen embeddings.
- Lastly, we use numerous other models such as distil-BERT, FastText, Fast- Text + BiGRU, and multi-lingual model such as XLM-Roberta and we find that both of these multi-lingual models outperform other models.

Rest of the paper is organized as follows Section 2. is related to the literature review of this problem Section 3. pertains to the dataset pipeline from collection to cleaning and labelling Section 4. explains the in-depth details of the experimental setup. In Section 5, we discuss the proposed approach and formal solution definition we use for our task Section 6. discusses our results and discussion of our findings Section 7., we present the ablation study in four different forms of BERT. In Section 8, we perform the error analysis on our findings. Finally, we conclude in Section 9.

2. Literature Review

In the field of Natural Language Processing (NLP) efforts are being directed to reduce the gap between human languages and computers understanding and comprehension. More specifically regional languages such as Urdu, Hindi, Ben- gali, Russian, Chinese etc. are yet to achieve the advancements in NLP that the English language has already achieved (Schmidt and Wiegand, 2017). In addition, with the massive increase in SM platforms and their users, the activities that can be categorized as hateful and offensive towards different groups have increased. For example on Twitter, Hate Speech targeting people from different gender, religion, communities, race etc. has seen monumental growth. Automated mechanisms are yet in the evolutionary phase to achieve the best results. Particularly in regional languages there is a large scope for research work. Much of the libraries that are designed to pre-process data and extract features are written in the English Language. Waseem-hovy et al. (Waseem and Hovy, 2016) have also used 16k tweets as a dataset, pro- vided a dictionary based on commonly used words found in hate speech and they have used extra-linguistic features in conjunction with character n-grams for hate speech detection. After performing grid search over all the possible features they were able to conclude that character n-grams outperforms word n-grams by 5 F1 points. Davidson et al. (Davidson et al., 2021) have found that racist and sexist tweets were more likely to be classified as hate speech than sexist tweets. They have used also classified tweets into three different classes: hate speech, offensive or neither. A number of classifiers were used for example Logistic Regression, SVMs, Naive Bayes, Decision Trees and Random Forests.

As the research is growing in this field and gradually the available data set is increasing, researchers have increasingly shifted focus towards Deep Learning models. Sajjad et al. (Sajjad et al., 2019) researchers have used fusion approach to detect hate speech. They have also used three classes to classify a tweet as either racist, sex- ist or none. They have used Convolutional Neural Networks (CNNs) to extract deep features and then integrated these features with state-of-the-art syntac- tic and word n-gram features. Badjatiya et al. (Badjatiya et al., 2017) have used multiple Deep learning techniques to detect hate speech from Twitter for the English Language corpus. They have classified whether a tweet is racist, sexist or neither.

In SemEval-2019 – the International Workshop on Semantic Evaluation – one of the major challenging task that was put forth to various participants was the classification of Twitter data into either hateful or not hateful. The participants used a mix of data pre-processing, feature engineering, feature selection, modelling and classification techniques to accurately classify the tweets. Popular machine learning models used were, Logistic Regression (LR), Support Vector Machines (SVMs) and Random Forests (RFs) algorithms. While, majority of the researchers preferred to use Deep Learning (DL) techniques to solve this classification problem. Apart from using Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2017), many researchers dived into sequential models such as Recurrent Neural Networks (RNNs) (Rumelhart et al., 1986), Long Short-Term Memory (LSTMs) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRUs) (Cho et al., 2014). These sequential models particularly performed well when they were fed with word embeddings as feature vectors as they generalized better as compared to simple statistical of semantic features. In fact in some cases, the results degraded when semantic features were fed to the neural network.

For example, Gertner et al. (Gertner et al., 2019) used an ensemble of various models to build their final model, their paper was the best ranked paper in SemEval-2019 competition. Their model included BiLSTM (Schuster and Paliwal, 1997) with name embeddings, hashtag prediction model including word2Vec (Mikolov et al., 2013) language-specific embeddings. They also applied word2phrase twice to pick up phrases of length four. According to their findings, shallower network with attention outperformed a deeper network without attention. HaCohen-Kerner et al. (HaCohen-Kerner et al., 2019) used a number of these pre- processing techniques, in

conjunction with n-gram features and varying RNN architectures that included Bidirectional RNN, 128 LSTM units, Dropout layer (0.4) and GloVe embeddings (Pennington and Manning, 2014) of 200d Benito et al. (Benito et al., 2019) used statistical features, semantic features, sentence polarity, word embeddings and linguistic features. For statistical features they collected n-grams, Bag of words and Term Frequency-Inverse Document Frequency (TF-IDF). For content analysis they assumed in their analysis that a tweet having HS will have negative polarity. To find polarity they have used TextBlob library. Similarly, pretrained word embeddings were used to convert words in to vectors. For linguistic features they have considered the number of sentences, length from the tweet, Parts of Speech (POS) statistics and some twitter related features. Finally, for classification they have used LR, SVMs with linear kernel and RFs.

2.1. Religious hate speech detection in arabic

Ghosh-Chowdhury et al. (Chowdhury et al., 2019), have worked on the detection of religious hate speech in the Arabic language. They have proposed an approach constituted by Arabic word embeddings and Social Network Graphs (Truong et al., 2016). They have also put specific emphasis on community features for the detection of hate speech. They were able to retrieve 3950 tweets in the Arabic language, 1685 tweets labelled as HS and 2265 as NHS. For preprocessing they have used an array of various features to remove hashtags, HTML links etc. Finally, for classification they have used 600d word embeddings in conjunction with LSTM and CNN to train the model.

2.2. Hate Speech Detection in Urdu/Roman Urdu

Perhaps the most seminal work in this problem domain is the recently published work by Rizwan et al. (Rizwan et al., 2020). The researchers have used an array of commonly used baseline models and deep learning models to present their findings for both coarse grained and fine grained classification tasks. They showed that transfer learning with their novel model of BERT + CNN-ngram was able to achieve an F1-score of 0.75 on fine grained classification task. Researchers are using an ensemble of various feature extraction and Machine Learning techniques to achieve best possible results. However, when it comes to resource poor languages like Urdu, hindi etc. the work is very limited. Akhter et al. (P. Akhter et al., 2020) the researchers have proposed a dataset for offensive language detection for Urdu and Roman Urdu. They have used character and word n-grams technique to extract features from the dataset. This means that they are only trying to capture local context information, which is unable to capture full context for hate speech detection. Furthermore, after extracting features they have used 17 ML classifiers to find results. Apart from this study, Mustafa et al. (Mustafa et al., 2017) the researchers have used tweets to identify controversial speech. They have used sequential pattern mining and sequential rule mining to find patterns and most repeated words. According to their research Naive Bayes outperform Logistic Regression and Support Vector Machine.

As has been mentioned before, much of the research work is focused on the English language when it comes to hate speech detection. As a result all the feature extraction techniques, pattern finding mechanisms and models are structured for that language. Moreover, the amount of available structured data for doing research in low resource languages is almost non-existent. While there are fairly well sized annotated publicly available datasets available for English, same cannot be said about low resource languages. In addition, to the best of out research, at present, there is no comprehensive published research on Hate Speech detection in the Urdu language. This is the reason why it is difficult at present to judge which mechanism would suit the data set best or how the data set should be tailored to extract best possible predictions possible. The success of transfer learning together with various transformer models is definitely an area we would be interested in exploring further for our dataset.

3. Urdu Hate Speech and Offensive Language Detection Dataset

3.1. Dataset Collection

Open-source tool Twint was used scrape data from Twitter platform. A thorough assessment of the commonly used words and slurs that are used to direct hate speech was made to identify commonly used keywords. Based on these keywords, that are reflective of offensive and hate speech, Twitter search was performed for the relevant tweets. After employing keyword search, some 20k tweets were gathered. The data set also contained many tweets in Farsi and Arabic language, because the Urdu language shares the writing style with these languages. Urdu tweets were retained and the remaining were discarded Fig. 2.

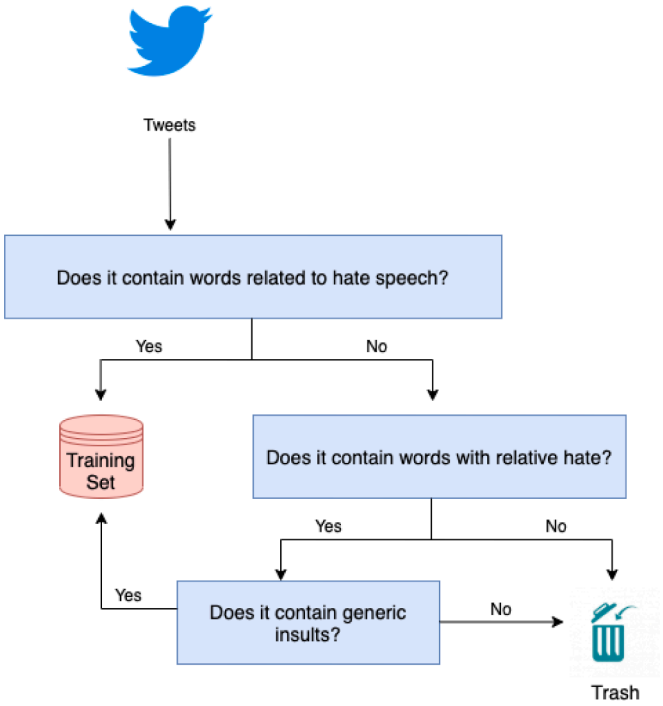

Fig. 2. Gathering Hate Speech Data.

3.2. *Dataset Labelling*

For the purpose of labelling, a total of 6 annotators were given a manual containing the necessary information on the subject of Hate Speech and Offen- sive Speech to annotate the dataset in three categories. Hate Speech labelled as 'h', Offensive speech as 'o', and a third neutral category as 'n'. The annotators were all Master's level researchers with well-reputed institutes from the National University of Sciences and Technology (NUST) and the Foundation for the Advancement of Science and Technology (FAST). To include gender diversity for better annotations, 3 male and 3 female annotators annotated the dataset. Dataset was divided into three equal sets and handed over to the annotators. Each tweet was annotated by two individuals, one male and one female annotator. Furthermore, in the case we have a conflict between two annotators that is the annotators have assigned differing labels, we added a third annotator to break the tie with majority voting. At the end a total of 10,529 labelled corpus with tweet count belonging to each category was obtained as shown in Table 1. In the Fig. 3 we can see that 41.0% of the dataset consists of hate speech tweets, 41.8% tweets belong to offensive category and 17.1% tweets are neutral tweets.

Table 1
Tweet count with respect to labels.

Label	Tweet Count
Hate Speech	4320
Offensive	4404
Neutral	1802
Total	10,526

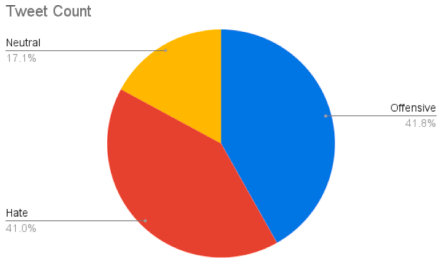

Fig. 3. Tweet Count Pie Chart.

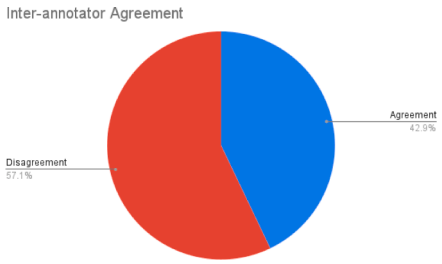

Fig. 4. Inter-Annotator Agreement and disagreement between two different annotators.

3.3. Inter-Annotator Agreement

Inter-Annotator Agreement (IAA) is a measure that is used in labelling tasks to ascertain how well multiple annotators are able to annotate dataset. Since we are dealing with a supervised NLP task, that is why it is necessary to find how accurately data was annotated. For this purpose, two of the most commonly used IAA measures are: Cohen's Kappa and Fleiss' Kappa. Cohen's Kappa (K) is used when two annotators annotate the dataset and Fleiss' Kappa is used when more than two annotators annotate the same dataset. In our case IAA is 42.9% as shown in Fig. 4, which is a moderate agreement according to Landis and kooch 1977 5 and Green 1997 Kappa interpretation Fig. 5.

$$K = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$

Where, *Pr(a)* represents actual agreement and *Pr(e)* denotes expected agreement.

3.4. Preprocessing

The next step after gathering the data set is the preprocessing and cleaning of the data set. For this purpose, a tweet cleaning pipeline as shown in Fig. 6 is designed to take every tweet into account and remove irrelevant and noisy data. The various steps performed are: Noise removal (Html tags, stop words, punctuations, white spaces etc.), Text normalization (tokenization, segmentation etc.). This results in a noise-free tweet, which can now be used for the feature engineering step.

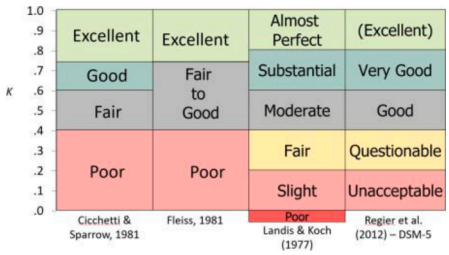

Fig. 5. Inter-rater agreement scale to measure the suitability of provided manual for anno- tations. (Source: Wikiwand).

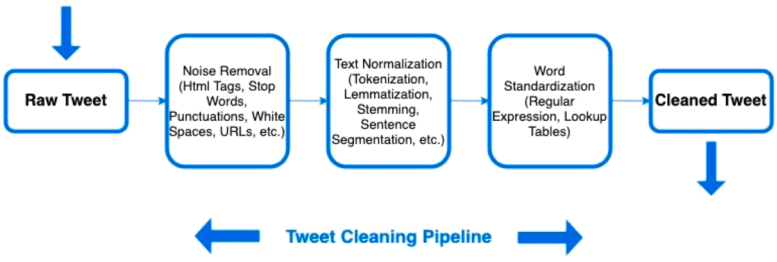

Fig. 6. Tweet cleaning pipeline is designed to remove noise, tokenize and standardize the tweets.

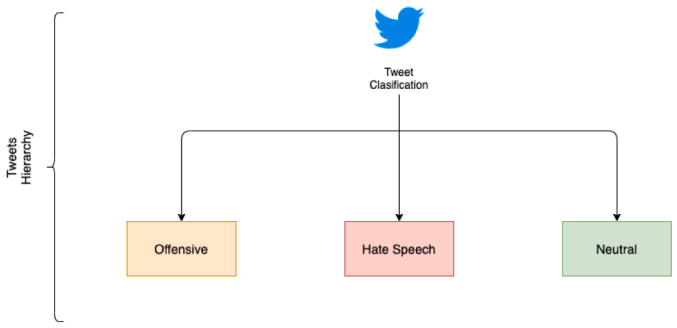

Fig. 7. Tweet Classification Hierarchy.

3.5. Classification

Fig. 7. shows the hierarchy of tweet classification. A single tweet is either labelled as hate speech, offensive speech or neutral (neither) based on the information it contains.

3.6. Samples of Tweets from the Dataset

We often see that whenever there is an incident that involves some controversial issue, Twitter bursts with a myriad of views. Some users use tweets to spew venom against people of other races, castes, creeds, religions, genders etc. A few of the examples are shown in Table 2.

3.7. Evaluation Methodology

For the evaluation purposes we split the dataset into train, validation and test set. Moreover, as is the general rule of thumb that 70% data set is used for training purposes and 10% is used for validation purposes and 20% is used for test purposes. Our dataset is split into 68% training set, 20% test set and 12% validation set as shown in Fig. 8. In addition the metric most commonly used to asses the viability is macro F1-score, hence we use this metric to judge the efficacy of the approach.

4. Experimental Setup

In this section we discuss all the details of the experimental setup that was used to assess the efficacy of various embeddings and baseline models.

Table 2
Hate Speech and offensive speech tweets dataset showing various gathered tweets in Urdu and their corresponding English translation.

Tweet (English)	Tweet (Urdu)	Label
There may be a black Abyssinian with a shadow over them	وہ ہےاس پر ان کا سچ وہ یشبح الاک یئوکے ہے اتکس وہ	Hate
Abyssinian black will be found in Africa	الاک یشبح اگے ملے ےس ہکی رفا	Hate
This is the religion of these illegitimate children	ین دا کا دوں ازامرح ان ےہ ہی	Offensive
The lowest is the one who calls a Muslim a Kafir. Sunni, Shia, Wahhabi, Deobandi are all Muslims	، یبابوہ، ہیعش، ینس ہے اتھک رفاک وکہ نام لسم وچ ہے نیرت لیلذ یہی نام لسم بس ندیبوید	Offensive
Sunni are infidels there, Shia are infidels here. May Allah guide us all	۔رفاکہیعش ہاں ی رفاک ینس ہاں وہ ۔دے تیادہ وکہ بس ھم ہللا	Neutral

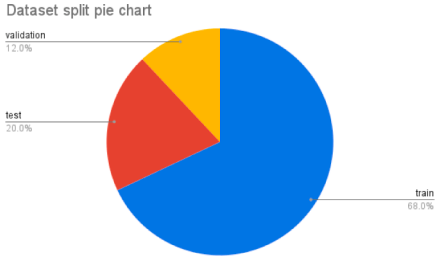

Fig. 8. Dataset split pie chart showing percentage of data split into train, test and validation sets.

As we have extensively discussed in the literature that local contextual information contributes little when it comes to the correct detection of hate speech. Rather, it is the overall context of the word that matters most to accurately detect hateful or offensive content. Therefore, various word embedding techniques come in handy and are able to capture contextual information significantly better Benito et al. (Benito et al., 2019). Apart from earlier word embedding techniques such as Word2Vec (Mikolov et al., 2013), TF-IDF vectorizer and CountVectorizer, that we use for baseline models, we also use FastText (Bojanowski et al., 2017) and BERT embeddings (Devlin et al., 2019). Initially, these embeddings were only pretrained for the English language and a few other languages but with rapid research progress owing to their astonishing results, these pre-trained embeddings are now available for low-resource languages as well. We use HuggingFace library (Wolf et al., 2020) tokenizer for both of these models.

4.1. Baseline Models

For the purpose of acquiring some baseline benchmark results on the dataset, we have used different baselines with our dataset.

TF-IDF + KNN: K-Nearest Neighbors (KNN) is the most obvious choice whenever a classification task is considered as it is amongst the most widely used supervised learning algorithms (Ferreira et al., 2020) (Ma and Cao, 2020). Therefore, as our first baseline we use *(KNN baseline)* with *(TF-IDF)* (Kadhim, 2019) vector representation for tweets classification.

TF-IDF + Random Forest: Our second choice for baseline is the Ran- dom Forests classifier (Breiman 2001) (*RF baseline*) that also uses 2000d *(TF-IDF)* vector representation (Sun et al., 2020).

TF-IDF + SVMs: Our third choice for baseline is the Support vector Machine (Cristianini et al., 2000) (*SVM baseline*) that uses 2000d *(TF-IDF)* vector representation (Tiun, 2017).

TF-IDF + AdaBoost: Our fourth choice for baseline is AdaBoost (Schapire, 1999) (*AdaBoost baseline*) that uses 2000d *(TF-IDF)* vector representation (Gaoa et al., 2021).

TF-IDF + Simple Perceptron: Our sixth choice for baseline is Simple Perceptron (*Simple Perceptron baseline*) that uses 2000d *(TF-IDF)* vector representation (Bounabi et al., 2018).

TF-IDF + GuassianNB: Our sixth choice for baseline is Guassian Process Classifier (*GuassianNB baseline*) that uses 2000d *(TF-IDF)* vector representation (Rezaeian, 2020).

TF-IDF + Quadratic Discriminant Analysis: Our seventh choice for baseline is Quadratic Discriminant Analysis (*Quadratic Discriminant Analysis baseline*) that uses 2000d *(TF-IDF)* vector representation (Fok et al., 2021).

CountVectorizer + KNN: Apart from TF-IDF vecotrs we also experi- ment with CountVectorizer. We use sklearn's 1000d CountVectorizer along with K-Nearest Neighbours (KNN) (Qadi et al., 2019) with *rbf* kernel and the *C* value of 100.

CountVectorizer + Random Forests: We use sklearn's 1000d CountVectorizer along with Random Forests (RF) (Qadi et al., 2019) with *rbf* kernel and the *C* value of 100.

CountVectorizer + SVMs: We use sklearn's 1000d CountVectorizer along with Support Vector Machines (SVMs) (Qadi et al., 2019) with *rbf* kernel and the *C* value of 100.

CountVectorizer + AdaBoost: We use sklearn's 1000d CountVectorizer along with AdaBoost (Qadi et al., 2019) with *rbf* kernel and the *C* value of 100.

CountVectorizer + Simple Perceptron: We use sklearn's 1000d CountVec- torizer along with Simple Perceptron (Qadi et al., 2019) with *rbf* kernel and the *C* value of 100.

CountVectorizer + Quadratic Discriminant Analysis: We use sklearn's 1000d CountVectorizer along with Quadratic Discriminant Analysis with *rbf* kernel and the *C* value of 100.

Word2Vec + KNN: In addition, we also use Word2Vec 200d embeddings trained on the dataset for 100 epochs with a context window of 5, ignoring the tweets where the minimum word count is less than 2, and the baseline model is KNN (Wang et al., 2017).

Word2Vec + Random Forests: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with context window of 5 with Random Forests (Ge and Moh, 2017).

Word2Vec + SVMs: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with a context window of 5 with SVMs (Wang et al., 2017).

Word2Vec + AdaBoost: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with a context window of 5 with AdaBoost (Rustam et al., 2019).

Word2Vec + Simple Perceptron: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with a context window of 5 with Simple Perceptron (Singh, 2018).

Word2Vec + Quadratic Discriminant Analysis: We use Word2Vec 200d embeddings trained on the dataset for 100 epochs with a context window of 5 with Quadratic Discriminant Analysis (Fok et al., 2021).

The results of these baseline models with feature vectors used are shown in Table 3. RandomForest classifier with CountVectorizer gave the best results, as it achieved a macro F1-score of 0.67 and an accuracy of 0.73. analysing its classification report we can observe that since neutral tagged tweets containing hate words are the most difficult to classify correctly, RandomForests gave encouraging F1-score of 0.48. This shows that it was able to handle the neutral tweets relatively better than other baselines. Moreover, we observe that SVMs with CountVectorizer as features also gave an encouraging f1 score of 0.67 and an accuracy of 0.72. Performance of AdaBoost classifier, simple perceptron and GuassianNB is almost identical with various features with a small difference. AdaBoost performed poorly with Word2Vec features as it gave an f1 score of only 0.59 and an accuracy of 0.63. Similarly, GuassianNB also performed very poorly with word2vec features by giving an f1 score of only 0.52 and accuracy of 0.56. In most of the baseline cases we observe that neutral tagged tweets are very difficult to handle. Our main challenge is to devise a method that can handle neutral tagged tweets in a better way.

Table 3
Baseline Classifiers and their Results.

Classifier	Features	P	R	F1	Acc
KNN	TF-IDF	0.71	0.59	0.60	0.68
	CV	0.67	0.59	0.60	0.66
	W2V	0.75	0.59	0.61	0.67
SVM	TF-IDF	0.74	0.60	0.60	0.68
	CV	0.71	0.66	0.67	0.72
	W2V	0.69	0.63	0.64	0.68
RF	TF-IDF	0.77	0.65	0.66	0.74
	CV	0.76	0.65	0.67	0.73
	W2V	0.76	0.61	0.62	0.68
AdaBoost	TF-IDF	0.67	0.63	0.63	0.70
	CV	0.67	0.63	0.64	0.69
	W2V	0.60	0.58	0.59	0.63
Simple Perceptron	TF-IDF	0.66	0.63	0.64	0.68
	CV	0.66	0.65	0.65	0.69
	W2V	0.66	0.62	0.63	0.67
GuassianNB	TF-IDF	0.73	0.61	0.62	0.70
	CV	0.60	0.57	0.56	0.61
	W2V	0.56	0.52	0.52	0.56
QDA	TF-IDF	0.70	0.51	0.50	0.60
	CV	0.55	0.49	0.47	0.52
	W2V	0.68	0.57	0.58	0.62

5. Proposed Model

Much of the literature revolves around some of the baseline machine learning methods such as Logistic Regression (LR), Linear Support Vector Machines (SVMs) and Random Forests (RF) are used to demonstrate the results. However, as it can be seen in the SemEval-2019 Task5 competition that many of the submissions were inclined to use Deep Learning methods such as Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTMs) (Baruah et al., 2019) and Gated Recurrent Units (GRUs) to classify text. The reason is that with the use of current frameworks to extract features, the Neural Network models have consistently outperformed the State-of-the-Art machine learning techniques. In addition to these techniques, recent advancements have shown that transformers based attention models are now considered state of the art models for many of the NLP tasks (Devlin et al., 2019). Attention mechanisms have the ability to learn the context of the words in a much better way and are more reliable in keeping the contextual information intact.

In this proposed approach as shown in Fig. 9 we use three distinct settings to find the best possible solution for our problem. First, we use pre-trained FastText embeddings (Bojanowski et al., 2017) and tokenizer from HuggingFace library (Wolf et al., 2020) to fine-tune these embeddings on our dataset. We then feed these fine-tuned embeddings to a FastText like model for classification from the high-level Ktrain wrapper of Keras (Maiya, 2021). Second, we again use fine-tuned FastText word vectors with Bi-directional Gated Recurrent Unit (BiGRU) model for classification. Finally, we use two different variants of Bidirectional Encoder Representations from Transformers (BERT) and fine-tune them on our dataset before feeding it into a BERT model for classification. In the first variant, we freeze the entire BERT architecture and add a feed-forward network that consists of two fully connected layers followed by softmax activation layer. In the second variant, we use transfer learning by training the entire pre-trained BERT multilingual model on our dataset.

5.1. Loss Function

Since our problem is a multi-class classification problem, therefore we need some kind of probabilistic output from our models pertaining to each class. Cross entropy loss is designed to generate class probabilities at the output layer. In our case we use cross-entropy loss that outputs probability predictions and these predictions, during training time, are compared with the ground truth labels. For wrong predictions, a logarithmic penalty is enforced and in successive training epochs, the purpose is to minimize the loss value.

$$H(p,q) = -\sum_{n=1}^{N_c} p(x)\log(q(x))$$

A cost function based on multi-class log loss for data set of size N would then be

$$J = -\frac{1}{N}\left(\sum_{i=1}^{N_c} y_i \log(\widehat{y}_i)\right)$$

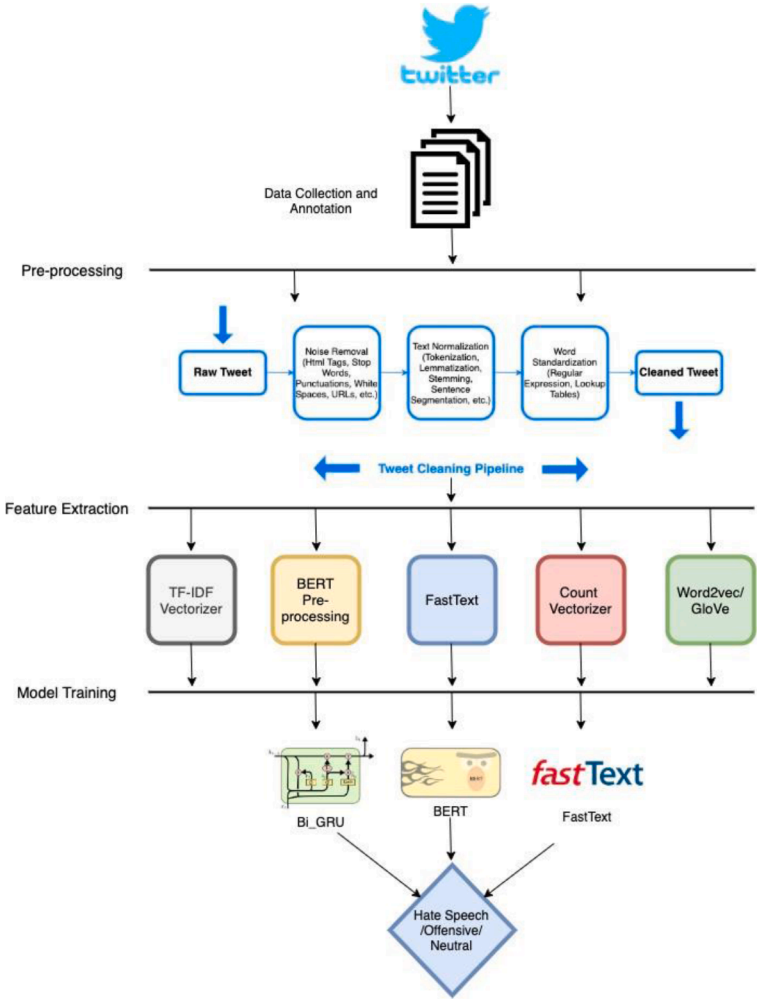

Fig. 9. System architecture diagram that shows multiple stages of tweet collection and an- notation, tweet cleaning, embeddings extraction and lastly classification models are employed.

5.2. *Hyper-parameters Tuning*

For the proposed model we use 'bert' pre-processing mode for pre-processing the dataset as it has to be pre-processed in a certain way. We select a batch size of 6, learning rate of 0.00002 as shown in Fig. 10 and maximum sequence length of 500d for all of the models, these were found to be optimal amongst various choices. In addition, we run BERT pre-trained model on our dataset for 3 epochs, we observe that running for higher epochs led to overfitting on our test dataset but the results on validation and test sets start to deteriorate. Moreover, in the case of FastText like model we find that it needs greater number of epochs to generalize well on the test set. Therefore, we run it for 25 epochs and similar is the case with BiGRU model, which is also run for 25 epochs. In both the cases running these models, on pre-trained embeddings, for higher epochs lead to overfitting on the test dataset.

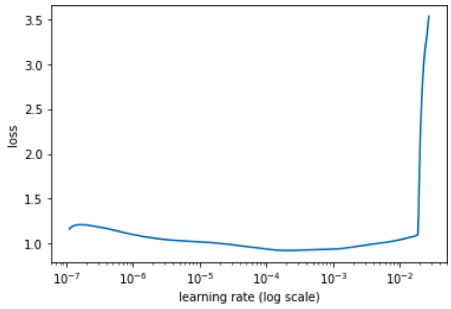

Fig. 10. Plot showing the learning rate and associated loss values on our dataset.

5.3. Formal Solution Definition

- Using a hate lexicon, created by researching the domain, that contains some commonly used words in hate speech

$$W \in \{w_1, w_2, w_3, ..., w_n\}$$

- We have collected Tweets

$$T \in \{t_1, t_2, t_3, ..., t_m\}$$

based on the hate words $\boldsymbol{W}$.

- These collected tweets were then classified into three classes, that is given a tweet $\boldsymbol{T}$ it may have a label

$$l = \{o, h, n\}$$

- Now we have a corpora of tweets $\boldsymbol{T}$ and their corresponding labels $\boldsymbol{l}$

$$\mathbf{T}, \mathbf{l} = \{(t_1, l), (t_2, l), (t_3, l), ..., (t_m, l)\}$$

- Next, we design and apply a tweet cleaning pipeline for the tweets. This pipeline contains commonly used preprocessing techniques represented by $\boldsymbol{D}$, for example noise removal, text normalization etc. as depicted in Fig 6.
- After that we find the word embeddings represented by $\boldsymbol{E}$.

$$E, T = \{(e_1, t_1), (e_2, t_2), (e_3, t_3), ..., (e_m, t_m)\}$$

Such that for every tweet that belongs to the tweet corpus $\boldsymbol{T}$ we get a corresponding vector embedding. And joining these embeddings for each tweet t in $\boldsymbol{T}$ we get a vector representation of $\boldsymbol{E}$.

- Finally, these word embeddings $\boldsymbol{E}$ and their corresponding true labels are fed to baseline machine learning models and Deep Learning architectures for training and subsequent predictions.

$$E, \boldsymbol{l} = \{(e_1, l), (e_2, l), (e_3, l), ..., (e_m, l)\}$$

6. Results and Discussion

In this section, we present discussion on our results obtained on test set of tweets as shown in Table 4. First, with FastText embeddings using a maximum vector length of 500d and FastText model like classifier Joulin et al. (Joulin et al., 2017) we are able to achieve 0.70 accuracy on the test set. The overall macro-f1 score is 0.63 and we can see that the main portion of mis-classification came from neutral tagged tweets. While f1-score for hate tweets is 0.70 and for offensive tweets 0.77, for neutral tweets it drops to 0.41.

Second, for Bi-directional Gated Recurrent Units (Bi-GRU) with 500d Fast- Text word vectors, we achieve an accuracy score of 0.72 and macro-f1 score of 0.67. For neutral tweets the f1-score is 0.50 which is slightly worse than the previous result. Out of the total 376 neutral-tagged tweets, the model was able to classify only 133 tweets correctly. This represents a challenge since the neutral tweets do contain the commonly used hate words but the context is overall neither hateful nor offensive.

Third, using BERT pre-processing in Ktrain wrapper, with BERT multilingual embeddings fine-tuned on our dataset and using standard BERT model, our results show an accuracy score of 0.73 and an overall macro-f1 score of 0.68. In the case of neutral tweets, it was able to reach 0.50 f1 score, slightly better than the RandomForests as mentioned earlier. This is particularly encouraging as out of total 376 neutral tweets, BERT was able to correctly classify 113 tweets.

Furthermore, we also exploit the concept of knowledge distillation by making use of Distil-BERT (Sanh et al., 2021) – a smaller, faster and lighter version of BERT – on our dataset. We observe that the training time is reduced significantly as compared to BERT. However, we do train Distil-BERT for a larger number of epochs than BERT, but even then the time taken is less than the BERT architecture. With Distil-BERT we manage to get an accuracy of 0.72 and F1-score of 0.69 which is the best score on our dataset. One particular success of this model is that it performed quite well on the neutral tagged tweets, as we see that it was able to correctly classify 199 tweets out of a total of 376 tweets as shown in Fig. 12.

Finally, in terms of multilingual models, we explore state-of-the-art XLM- Roberta model published by FacebookAI group (Conneau

Table 4
Results of Various Embeddings on Urdu Hate speech and Offensive Language Dataset.

	Accuracy	F1-sccore	Precision	Recall
FastText	0.70	0.63	0.76	0.61
FastText + BiGRU	0.72	0.67	0.76	0.65
Bert	0.73	0.68	0.73	0.66
DistilBert	0.72	0.69	0.70	0.69
XLMRoberta	0.71	0.68	0.69	0.67

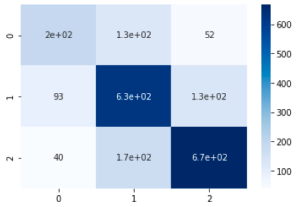

Fig. 11. Confusion Matrix XLM-Roberta.

et al., 2020) on our dataset. In terms of accuracy we get 0.71 and F1 score of 0.68. Both distil-BERT and XLM- Roberta models are run for 25 epochs and both are able to perform well on neutral tagged tweets. In the case of XLM-Roberta 196 neutral tweets out of 376 neutral tweets were classified accurately as shown in Fig. 11.

7. Ablation Study

In this section, we present our findings by using different modifications of BERT architecture. The closest related work that we could find was for Roman Urdu language in paper (Hate-Speech and Offensive Language Detection in Roman Urdu). In this paper the authors used a technique to pick upon the textual information by using 1D CNN layers. Furthermore, the results showed that ensembling BERT with CNN layers improved results in the said paper. Therefore, we also experiment the same with our dataset. In our results we find that infusing CNN layers with BERT could not improve results. Moreover, we also explore whether we can improve results, since our dataset is small, without using tansfer learning and training some Fully connected layers on top of frozen pre-trained BERT embeddings. In our finding we reach to a conclusion that using this technique had no significant impact on improvement of results as shown in Table 5. We present an ablation study in terms of transfer learning in four parts;

- First, we use BERT in its vanilla form by freezing its entire pre-trained architecture and adding a feed-forward network with a softmax activation layer at the end. The loss is propagated to the feed-forward network only and the BERT architecture remains as it is.

Table 5
Results of Ablation Study on Urdu Hate speech and Offensive Language Dataset.

	Accuracy	F1-sccore	Precision	Recall
BERT Freeze	0.51	0.48	0.49	0.49
BERT Unfreeze	0.73	0.68	0.73	0.66
BERT Freeze + CNN	0.57	0.51	0.59	0.50
BERT Unfreeze + CNN	0.67	0.58	0.77	0.58

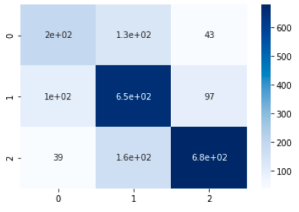

Fig. 12. Confusion Matrix DistilBert.

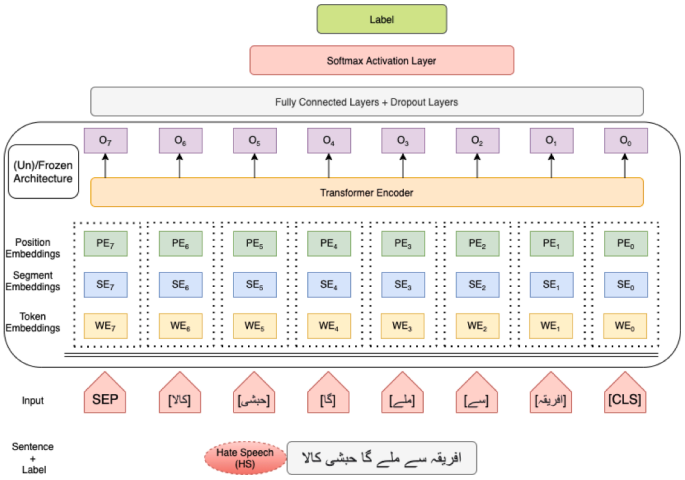

Fig. 13. Frozen/Unfrozen BERT Architecture with Fully Connected layers.

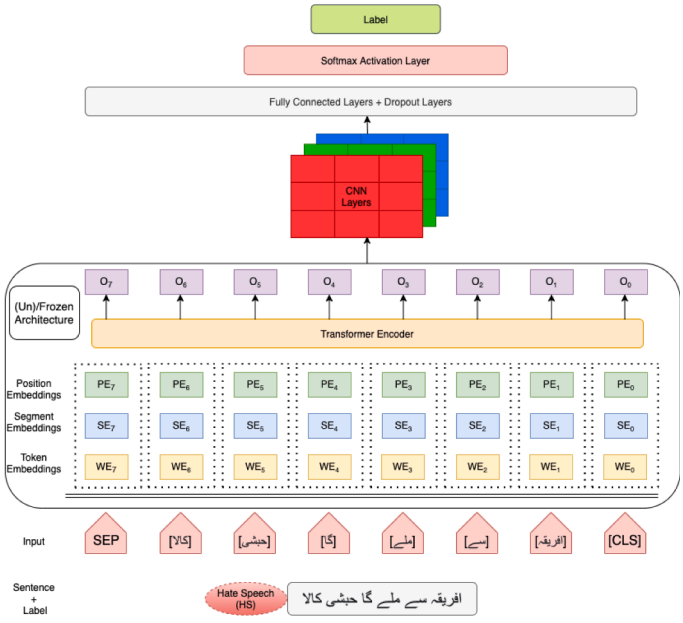

Fig. 14. Frozen/Unfrozen BERT Architecture with CNN Layers and Fully Connected layers.

- Second, we unfreeze BERT architecture and add feed-forward network with a softmax activation at the end as shown in Fig. 13. In this case, the loss is propagated to the entire architecture and entire BERT architecture plus the feed- forward network are fine-tuned on our dataset.
- Third, we again freeze BERT architecture and add a few convolution layers and fully connected layers to the original BERT. In this case, only the additional CNN layers and fully connected layers get updated.
- Lastly, we un-freeze BERT and use the entire architecture for fine-tuning by adding additional CNN and fully connected layers as shown in Fig. 14. Here again the entire BERT architecture and the additional layers are fine-tuned .

8. Error Analysis of Various Models

In this section, we will dive deep into the correct predictions and mis- classifications of our models. Our analysis of results show that xlm-roberta and distil-bert models were able to correctly classify offensive and neutral tagged tweets considering the context of the tweets. In instances, institution re- lated offensive speech and gender related neutral context were learnt by the model. However, it is debatable that institution related tweet might be categorised as hate speech, but it might be label bias that it was tagged as such by

Table 6
Hate Speech and offensive speech tweets with their corresponding ground truth labels, XLM-Roberta and distil-BERT predictions.

Tweet (English)	Tweet (Urdu)	True Label	XLM_Rob	distilBERT
The watchman's watch is in danger, now he will show his watch	چوکیدار کی چوکیداری کو خطرہ لاحق ہوگیا ہے، اب وہ اپنی چوکیداری کا دکھائے گا۔	hate	hate	neutral
These poor are envelope journalists	یہ لفافہ صحافی ہیں بےچارے	offensive	hate	offensive
And our envelope journalists are spreading this statement now.	اور ہمارے لفافہ صحافی اس بیان کو اب پھیلا رہے ہیں	hate	offensive	offensive

three different annotators. In addition, as shown in Table 6 if we analyse the first tweet we see that true label is hate while distil-bert could not learn this tweet accurately, multi-lingual xlm-roberta was able to learn accurately. Conversely, offensive labelled tweet, as shown in second example, was learnt by distil-bert model, but xlm-roberta model categorised it as hate speech. In our research opinion it should be classified as hate speech given that it targets journalist community specifically.

Moreover, we also experiment with Ktrain's in-built function to analyse the top losses while model training. Beginning with BERT as it the best performing technique on our dataset, we observe top 20 losses and found a mixture of mis- classification. In most of the cases neutral tweets were mis-classified as hate speech as they contained a strong word that is commonly used to express hatred. In these tweets, BERT was unable to learn and understand the whole context which was neutral. Interestingly, we also see that in some cases offensive tagged tweets were classified as hate speech, when in fact looking at these examples we can clearly see that it is a hate-related tweet. This is in part due to human-level error, but is encouraging to see that the model was able to correctly identify it as hate speech. One such example is مرضی میری میری ہیں سکتے کر ادا کیا اور آپ سے خواتین گ (What else can you expect from my body my choice pros****tes.)

Similarly, in the case of FastText model, we see that again neutral tweets were mis-classified as hate speech. One example in the case of FastText is الحمد و للہ میری قوم میں ماشاء اللہ اب بھی ایسی خواتین ہیں جن پر فخر کر کہ سکتا ہوں کہ وہ اقدار پر، کر ار پر اور پر سمجھوتہ نہیں کرتیں چاہے جتنی ہی لبرل آپ کو سلام ہے (Fortunately, there are still women in my nation who are proud to say that they do not compromise on values, character and so on, no matter how liberal you are.)

9. Conclusion

In conclusion, we can say that BERT, multi-lingual models such as xlm- roberta and distil-Bert are largely able to learn the contextual information in the tweets and accurately classify hate and offensive speech. DistilBert with tansfer learning gave the maximum F1-score of 0.69 while both BERT and XLM- Roberta also performed similarly, albeit with a small difference of F1-score of 0.67. We do acknowledge that when it comes to neutral tweets containing com- monly used hate words, various models struggle to capture the whole context. However, we have shown that in the case of neutral tweets multi-lingual models perform well as discussed in the success and error analysis section. Broadly speaking, the results are encouraging and can be further improved upon with the help of better understanding of hate speech and its variant that flood social media platforms. In future, this work may be extended to include the fine- grained classification of hate speech tweets. In addition, we may also improve results with the help of a pre-trained trans-former model trained specifically on a corpus of local Urdu language dataset so that better contextual information is learnt.

Declaration of Competing 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

Akhter, M.P., Jiangbin, Z., Naqvi, I.R., Abdelmajeed, M., Sadiq, M.T., 2020a. Automatic detection of offensive language for urdu and roman urdu. IEEE Access 8, 91213–91226. https://doi.org/10.1109/ACCESS.2020.2994950.

Akhter, P., Jiangbin, Z., Naqvi, I.R., Abdelmajeed, M., Sadiq, M.T., 2020b. Automatic detection of offensive language for urdu and roman urdu. IEEE Access 8, 91213–91226. https://doi.org/10.1109/ACCESS.2020.2994950.

Aljarah, M.Habib, Hijazi, N., Faris, H., Qaddoura, R., Hammo, B., Abushariah, M., Alfawareh, M., 2020. Intelligent detection of hate speech in ara- bic social network: a machine learning approach. J. Inf. Sci., 0165551520917651

P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: Proceedings of the 26th International Conference On World Wide Web Companion, 2017, pp. 759–760.

Baruah, A., Barbhuiya, F., Dey, K., 2019. ABARUAH at SemEval-2019 task 5 : bi-directional LSTM for hate speech detection. Proceedings of the 13th International Workshop On Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 371–376. https://doi.org/10.18653/v1/S19-2065. URL. https://www.aclweb.org/anthology/S19-2065.

Benito, D., Araque, O., Iglesias, C.A., 2019. GSI-UPM at SemEval-2019 task 5: semantic similarity and word embeddings for multilingual detection of hate speech against immigrants and women on Twitter. Proceedings of the 13th International Workshop On Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 396–403. https://doi.org/10.18653/v1/S19-2070. URL. https://www.aclweb.org/anthology/S19-2070.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. Enriching word vectors with sub word information. Trans. Assoc. Comput. Linguistics 5, 135–146. https://doi.org/10.1162/tacl_a_00051. URL. https://www.aclweb.org/anthology/Q17-1010.

Bounabi, M., El Moutaouakil, K., Satori, K., 2018. A probabilistic vector repre- sentation and neural network for text classification. International Con- ference on Big Data, Cloud and Applications. Springer, pp. 343–355.

Breiman, 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Cho, B.van Merriënboer, Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: encoder–decoder approaches. Proceed- ings of SSST-8, Eighth Workshop On Syntax, Semantics and Structure in Statistical Translation. Association for Computational Linguistics, Doha, Qatar, pp. 103–111. https://doi.org/10.3115/v1/W14-4012. URL. https://www.aclweb.org/anthology/W14-4012.

Chowdhury, Ghosh, Didolkar, A., Sawhney, R., Shah, R.R., 2019. ARHNet - leveraging community interaction for detection of religious hate speech in Arabic. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. Association for Computational Linguistics, Florence, Italy, pp. 273–280. https://doi.org/10.18653/v1/P19-2038. URL. https://www.aclweb.org/anthology/P19-2038.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V., 2020. Unsuper- vised cross-lingual representation learning at scale. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747. OnlineURL. https://www.aclweb.org/anthology/2020.acl-main.747.

Cristianini, J.Shawe-Taylor, et al., 2000. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge university press.

T. Davidson, D. Warmsley, M. Macy, I. Weber, (2021) Automated hate speech detection and the problem of offensive language.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Min- neapolis, Minnesota, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423. URL. https://www.aclweb.org/anthology/N19-1423.

Ferreira, P.J., Cardoso, J.M., Mendes-Moreira, J., 2020. Knn prototyping schemes for embedded human activity recognition with online learning. Computers 9 (4), 96.

H. Fok, J.A. Jimenez, D. Guest, J. Houghton, S. Debloudts, (2021) Text classification, a general approach.

J. Gaoa, H. Ninga, Z. Han, L. Kongb, H. Qib, (2021) Legal text classification model based on text statistical features and deep semantic features.

Ge, L., Moh, T.-S., 2017. Improving text classification with word embedding. 2017 IEEE International Conference On Big Data (Big Data). IEEE, pp. 1796–1805.

Gertner, A., Henderson, J., Merkhofer, E., Marsh, A., Wellner, B., Zarrella, G., 2019. MITRE at SemEval-2019 task 5: transfer learning for multilingual hate speech detection. Proceedings of the 13th International Workshop On Se- mantic Evaluation. Association for Computational LinguisticsMinneapo- lis, Minnesota, USA, pp. 453–459. https://doi.org/10.18653/v1/S19-2080. URL. https://www.aclweb.org/anthology/S19-2080.

HaCohen-Kerner, Y., Shayovitz, E., Rochman, S., Cahn, E., Didi, G., Ben- David, Z., 2019. JCTDHS at SemEval-2019 task 5: detection of hate speech in tweets using deep learning methods, character n-gram features, and preprocessing methods. Proceedings of the 13th International Workshop On Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 426–430. https://doi.org/10.18653/v1/S19-2075. URL. https://www.aclweb.org/anthology/S19-2075.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735. URL 10.1162/neco.1997.9.8.1735.

Ibrohim, M.O., Budi, I., 2018. A dataset and preliminaries study for abusive language detection in indonesian social media. Procedia Comput. Sci. 135, 222–229.

Joulin, E.Grave, Bojanowski, P., Mikolov, T., 2017. Bag of tricks for efficient text classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. Association for Computational Linguistics, Valencia, Spain, pp. 427–431. URL. https://www.aclweb.org/anthology/E17-2068.

Kadhim, A.I., 2019. Survey on supervised machine learning techniques for au- tomatic text classification. Artif. Intell. Rev. 52 (1), 273–292.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90. https://doi.org/10.1145/3065386. URL 10.1145/3065386.

Ma, X.Du, Cao, L., 2020. Improved knn algorithm for fine-grained classifica- tion of encrypted network flow. Electronics (Basel) 9 (2), 324.

A.S. Maiya, (2021) ktrain: a low-code library for augmented machine learning, arXiv preprint arXiv:2004.10703.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (Eds.), 1st International Conference on Learning Representations. ICLR, Scottsdale, Arizona, USA. May 2-4, 2013, Workshop Track Proceedings, 2013. URL. http://arxiv.org/abs/1301.3781.

Mustafa, R.U., Nawaz, M.S., Farzund, J., Lali, M., Shahzad, B., Viger, P., 2017. Early detection of controversial urdu speeches from social media. Data Sci. Pattern Recognit. 1 (2), 26–42.

Pennington, R.Socher, Manning, C., 2014. GloVe: global vectors for word representation. Proceedings of the 2014 Conference on Empirical Meth- ods in Natural Language Processing (EMNLP). Association for Computa- tional Linguistics, Doha, Qatar, pp. 1532–1543. https://doi.org/10.3115/v1/D14-1162. URL. https://www.aclweb.org/anthology/D14-1162.

Perelló, C., Tomás, D., Garcia-Garcia, A., Garcia-Rodriguez, J., Camacho- Collados, J., 2019. UA at SemEval-2019 task 5: setting a strong linear baseline for hate speech detection. Proceedings of the 13th International Workshop On Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 508–513. https://doi.org/10.18653/v1/S19-2091. URL. https://www.aclweb.org/anthology/S19-2091.

L.A. Qadi, H.E. Rifai, S. Obaid, A. Elnagar, Arabic text classification of news articles using classical supervised classifiers, in: 2019 2nd Interna- tional Conference On New Trends in Computing Sciences (ICTCS), 2019, pp. 1–6. doi:10.1109/ICTCS.2019.8923073.

Rezaeian, G.Novikova, 2020. Persian text classification using naive bayes algorithms and support vector machine algorithm. Indones. J. Electr. Eng. Inform. (IJEEI) 8 (1), 178–188.

Rizwan, H., Shakeel, M.H., Karim, A., 2020. Hate-speech and offensive lan- guage detection in Roman Urdu. Proceedings of the 2020 Confer- ence on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, pp. 2512–2522. https://doi.org/10.18653/v1/2020.emnlp-main.197. OnlineURL. https://www.aclweb.org/anthology/2020.emnlp-main.197.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning Internal Repre- sentations by Error Propagation. MIT Press, Cambridge, MA, USA, pp. 318–362.

Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., Choi, G.S., 2019. Tweets classifi- cation on the base of sentiments for us airline companies. Entropy 21 (11), 1078.

M. Sajjad, F. Zulifqar, M.U.G. Khan, M. Azeem, Hate speech detection using fusion approach, in: 2019 International Conference on Applied and Engineering Mathematics (ICAEM), 2019, pp. 251–255. doi:10.1109/ICAEM.2019.8853762.

V. Sanh, L. Debut, J. Chaumond, T. Wolf, (2021) Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108.

Schapire, R.E., 1999. A brief introduction to boosting. Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI'99. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1401–1406.

Schmidt, A., Wiegand, M., 2017. A survey on hate speech detection using natural language processing. Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Compu- tational Linguistics, Valencia, Spain, pp. 1–10. https://doi.org/10.18653/v1/W17-1101. URL. https://www.aclweb.org/anthology/W17-1101.

Schuster, M., Paliwal, K., 1997. Bidirectional recurrent neural networks. Trans. Sig. Proc. 45 (11), 2673–2681. https://doi.org/10.1109/78.650093. URL 10.1109/78.650093.

M. Singh, Nepali multi-class text classification (2018).

Sun, Y., Li, Y., Zeng, Q., Bian, Y., 2020. Application research of text classification based on random forest algorithm. 2020 3rd International Conference On Advanced Electronic Materials, Computers and Software Engineering (AEMCSE). IEEE, pp. 370–374.

Tiun, S., 2017. Experiments on malay short text classification. 2017 6th In- ternational Conference On Electrical Engineering and Informatics (ICEEI). IEEE, pp. 1–4.

Truong, Q.-.D., Dkaki, T., Truong, Q.-.B., 2016. Graph methods for social network analysis 168, 276–286. https://doi.org/10.1007/978-3-319-46909-6_25.

Wang, Y., Zhou, Z., Jin, S., Liu, D., Lu, M., 2017. Comparisons and selections of features and classifiers for short text classification. IOP Conference Series: Materials Science and Engineering, Vol. 261. IOP Publishing, 012018.

Waseem, Z., Hovy, D., 2016. Hateful symbols or hateful people? predictive fea- tures for hate speech detection on Twitter. Proceedings of the NAACL Student Research Workshop. Association for Computational Linguistics, San Diego, California, pp. 88–93. https://doi.org/10.18653/v1/N16-2013. URL. https://www.aclweb.org/anthology/N16-2013.

Winter, K., Kern, R., 2019. Know-center at SemEval-2019 task 5: multilingual hate speech detection on Twitter using CNNs. Proceedings of the 13th International Workshop On Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 431–435. https://doi.org/10.18653/v1/S19-2076. URL. https://www.aclweb.org/anthology/S19-2076.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A., 2020. Transformers: state-of-the-art natural language pro- cessing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, pp. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6. OnlineURL. https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R., 2019. SemEval-2019 task 6: identifying and categorizing offensive lan- guage in social media (OffensEval). Proceedings of the 13th Inter- national Workshop On Semantic Evaluation. Association for Computational Linguistics, Minneapolis, Minnesota, USA, pp. 75–86. https://doi.org/10.18653/v1/S19-2010. URL. https://www.aclweb.org/anthology/S19-2010.