Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model

Hind Saleh, Areej Alhothali & Kawthar Moria

Published online: 02 Feb 2023.

Submit your article to this journal ↗

Article views: 11674

View related articles ↗

View Crossmark data ↗

Citing articles: 57 View citing articles ↗

Taylor & Francis
Taylor & Francis Group

Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model

Hind Saleh [ID][a,b], Areej Alhothali[b], and Kawthar Moria[b]

[a]Computer Science Department, University of Tabuk, Tabuk, The Kingdom of Saudi Arabia; [b]Computer Science Department, King Abdulaziz University Jeddah, Makkah, The Kingdom of Saudi Arabia

ABSTRACT

There is an increased demand for detecting online hate speech, especially with the recent changing policies of hate content and free-of-speech right of online social media platforms. Detecting hate speech will reduce its negative impact on social media users. A lot of effort in the Natural Language Processing (NLP) field aimed to detect hate speech in general or detect specific hate speech such as religion, race, gender, or sexual orientation. Hate communities tend to use abbreviations, intentional spelling mistakes, and coded words in their communication to evade detection, which adds more challenges to hate speech detection tasks. Word representation from its domain will play an increasingly pivotal role in detecting hate speech. This paper investigates the feasibility of leveraging domain-specific word embedding as features and a bidirectional LSTM-based deep model as a classifier to automatically detect hate speech. This approach guarantees that the word is assigned its negative meaning, which is a very helpful technique to detect coded words. Furthermore, we investigate the use of the transfer learning language model (BERT) on the hate speech problem as a binary classification task as it provides high-performance results for many NLP tasks. The experiments showed that domain-specific word embedding with the bidirectional LSTM-based deep model achieved a 93% f1-score, while BERT achieved 96% f1-score on a combined balanced dataset from available hate speech datasets. The results proved that the performance of pre-trained models is influenced by the size of the trained data. Although there is a huge variation in the corpus size, the first approach achieved a very close result compared to BERT, which is trained on a huge data corpus, this is because it is trained on data related to the same domain. The first approach was very helpful to detect coded words while the second approach achieved better performance because it is trained on much larger data. To conclude, it is very helpful to build large pre-trained models from rich domains specific content in current social media platforms.

CONTACT Hind Saleh ✉ h.alatwi@ut.edu.sa 🖃 Computer Science Department, King Abdulaziz University, Tabuk, Kingdom of Saudi Arabia

Introduction

Social media has been used extensively for various purposes, such as advertising, business, news, etc. The idea of allowing users to post anything at any time on social media contributed to the existence of inappropriate content on social media. As a result, these platforms become a fertile environment for this type of content. Hate speech is the most common form of destructive content on social media, and it can come in the form of text, photographs, or video. It is defined as an insult directed at a person or group based on characteristics such as color, gender, race, sexual orientation, origin, nationality, religion, or other characteristics (Weber 2009). Hate speech poses a significant threat to communities, either by instilling hatred in young people against others or by instigating criminal activity or violence against others.

Hate speech on the internet is on the rise around the world, with approximately 60% of the global population (4.54 billion) using social media to communicate (Ltd, 2020). According to studies, approximately 53% of Americans have encountered online harassment and hatred (League 2019). This score is 12 points higher than the findings of a similar survey performed in 2017 (Duggan 2017). According to (Clement 2019), 21% of students frequently encounter hate speech on social media. Detection of hate content on social media is an essential and necessary requirement for social media platforms. Social media providers work hard to get rid of this content for a safer social environment, which motivates us to work on this problem. Automatic detection of hateful content is considered one of the challenging NLP tasks as the content might target/attack individuals or groups based on various characteristics using different hate terms and phrases (Badjatiya et al. 2017). Through this work, we seek to provide an experimental-based solution to automatically detect all hate speech terms using real-world data from social media.

Social media users often employ abbreviations and ordinary words (not hateful) to express their hate intent implicitly that known as code words to evade being detected (e.g., using Google to refer to dark-skinned people), which adds extra difficulties in detecting hate speech. Many studies have proposed machine learning models to handle this problem by utilizing a wide range of feature sets and machine learning algorithms for classification (Agarwal and Sureka 2015; Hartung et al. 2017; Jaki and De Smedt 2019; Magu, Joshi, and Luo 2017). These methods often utilize features that require considerable effort and time to be extracted, such as text-based, profile-based, and community-based features. Other studies have worked on linguistic-based features (e.g., word frequency) and deep learning for classification (Gibert et al. 2018), or distributional-based features (e.g., word embedding) and machine learning classifier (Badjatiya et al. 2017; Djuric et al. 2015; Gupta and Waseem 2017; Nobata et al. 2016).

Several research studies have attempted to solve the problem of detecting hate speech in general by differentiating hate and non-hate speech (Djuric et al. 2015; Ribeiro et al. 2017). Others have tackled the issue of recognizing certain types of hate speech, such as anti-religious hate speech (Albadi, Kurdi, and Mishra 2018; Zhang, Robinson, and Tepper 2018), jihadist (Ferrara et al. 2016; Gialampoukidis et al. 2017; Smedt, Tom, and Van Ostaeyen 2018; Wei, Singh, and Martin 2016), sexist, and racist (Badjatiya et al. 2017; Gambäck and Kumar Sikdar 2017; Pitsilis, Ramampiaro, and Langseth 2018). The problem has been addressed from different points of view seeking to achieve a state-of-the-art result which is not yet been achieved. This work also aims to achieve better results for hate speech problems.

Studies show that distributional features provide a promising result in NLP tasks such as sentiment analysis (Gupta and Waseem 2017). Recently, deep learning methods also show that it performs well on various NLP problems (Socher, Bengio, and Manning 2012). Accordingly, our proposed solution investigates the performance of employing domain-specific word embedding/distributional representation features as it is one of the distributional-based learning methods and deep learning classifiers which is bidirectional Long Short-Term Memory (BiLSTM) to detect hate speech. The word embedding in this research is built upon a hate speech corpus of $1,048,563$ sentences to reach the closest meaningful representation vector of hate words. Then, compare it with the domain-agnostic embedding model such as Google Word2Vec and GloVe under the same classifier. We also assess the performance of detecting hate speech using Google's pre-trained BERT model, which has generally achieved state-of-the-art for many NLP tasks. The contributions of this research are highlighted as follows:

• An unsupervised domain-specific word embedding model was developed to extract the meaning of commonly used terminology, acronyms, and purposefully misspelled hate words.

• A comparison between the domain-specific and domain-agnostic embedding was provided. The findings show that domain-agnostic embedding performs slightly better (about 1%), despite the huge difference in the trained corpus size.

• The evaluation of a BiLSTM-based deep model with domain-specific embeddings shows an improvement ranging from 5 to 6 points on available datasets over the state-of-the-art techniques.

• The evaluation of the BERT language model on the hate speech binary classification task shows an improvement of about 2 points compared to the domain-specific word embedding model.

This study focuses on the detection of English language hate speech including all its types (e.g., race, sex, gender, etc.), and its levels (e.g., offensive and hate) as a binary classification task (hate or not hate).

The remaining of this paper is constructed as follows: the background section, which explains information about the applied methodologies; the review of literature section summarizes the most recent related studies; the methodology section provides detailed descriptions of proposed solution methods; the experiment and result section includes datasets, embedding models, and results of the experiments; the discussion section encompasses analysis and observation from the results, and finally the conclusion section summarizes all the findings.

Background

This section gives an overview of the used methodologies for both features and classifiers.

Word Embedding

Word embedding (Bengio et al. 2003) is a prominent natural language processing (NLP) technique that seeks to convey the semantic meaning of a word. It provides a useful numerical description of the term based on its context. The words are represented by an N-dimensional dense vector that can be used in estimating the similarities between the words in a specific language (Liu 2018; Mikolov et al. 2013). The word embedding has been widely used in many recent NLP tasks due to its efficiency such as text classification (Gambäck and Kumar Sikdar 2017; Lilleberg, Zhu, and Zhang 2015), document clustering (Ailem, Salah, and Nadif 2017), part of speech tagging (P. Wang et al. 2015) named entity recognition (Sienčnik 2015), sentiment analysis (Al-Azani and El-Alfy 2017; Tang et al. 2014; J. Wang, Liang-Chih Yu, and Zhang 2016), and many other problems. The most common pretrained word embedding models are Google Word2Vec, and Stanford GloVe, which are described in the following subsections.

Word2vec

Word2Vec is one of the most-used word embedding models. It is provided by the Google research team (Mikolov et al. 2013). Word2Vec associates each word with a vector based on its surrounding context from a large corpus. The training process for extracting the word vector has two types, the continuous bag of word model (CBOW), which predicts the target word from its context, and the Skip-Gram model (SG), which predicts the target context from a given word. The feature vector of the word is manipulated and updated according to each context the word appears in the corpus. Google has released a vector model called Google Word2Vec that has been trained on a massive corpus of over 100 billion words.

GloVe

GloVe (Global Vectors for Word Representation) is another popular word embedding model (Pennington, Socher, and Manning 2014). GloVe learns embeddings using an unsupervised learning algorithm that is trained on a corpus to create the distributional feature vectors. During the learning process, a statistics-based matrix is built to represent the word-to-word co-occurrence of the corpus. The main difference between GloVe and Word2Vec is in the learning process, Word2Vec is a prediction-based model, while GloVe is a count-based model. The GloVe is learned from Wikipedia, web data, and Twitter and it has models with different vector dimensions.

Bidirectional Long Short-Term Memory (BiLstm)

LSTM (Hochreiter and Schmidhuber 1997) is an enhanced version of the recurrent neural network, which is one of the deep learning models that is designed to capture information from a sequence of information. LSTM saves data for long sequences only from left to right. However, to save sequence data from both directions, a bidirectional LSTM (BiLSTM) is used. BiLSTM consists of two LSTMs, one processes the data from left to right and the other in opposite direction then concatenates and flattens both forward and backward LSTM to improve the knowledge of the surrounding context.

BERT Pre-Trained Language Model

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018) is a language model trained on very huge data based on contextual representations. BERT consists of feature extraction layers, which consist of word embedding and layers for the model (e.g., Classification, Question Answering, and Named Entity Recognition). BERT is the most recent language model and provides state-of-the-art results in comparison to other language models for various NLP tasks. BERT differs from other word embedding models in the training procedure of word embedding as it creates a bidirectional representation of words that may be learned from both left and right directions. Word embedding approaches like Word2Vec and GloVe only examine one direction (either left to right or right to left), resulting in static word representation that does not change with context. BERT is also different from previous language models (e.g., ELMo stands for Embeddings from Language Models (Peters et al. 2018)) in that it manipulates the context in all layers in both directions (left and right). Instead of shallow combining processes such as concatenating, it uses cooperative conditioning to combine both the left and right contexts. BERT is trained on Books Corpus (800 M words) and English Wikipedia (2,500 M words) devlin2018bert.

Review of Literature

Word embedding is an effective approach for different NLP issues. It has been used to extract bio-events from the scientific literature (Chen et al. 2015). They used multiple sets of features such as word embedding, BOW + n-gram joint model, and word embedding BOW joint model with SVM classifier, and the overall performance of word embedding BOW is better than other models on different events, which achieved about 77.37% f1-score. The small dataset size influences negatively the performance of the word embedding model. Word embedding was employed in (Yonghui et al. 2015) study for distinguishing clinical abbreviations as a special case of word sense disambiguation (WSD). The performance of SVM utilizing word embedding features increased with an average accuracy of 93%.

Recently, researchers have been interested in detecting hate speech on social media more accurately. The study of (Liu 2018) used a domain-specific word embedding model trained on the articles from hate speech websites and high centrality users' tweets to reach to the semantics of code words used in hate speech. They experimented on CNN, and LSTM models and concluded that CNN performed better than LSTM on tweets due to the length of tweets. They achieved 78% f1-score but they experimented on the previous tweet length, which was limited to 180 characters. The performance of using the hate Word2Vec (i.e., domain-specific) model was also examined by (Gupta and Waseem 2017) experiments with Logistic Regression (LR) classifier on three different datasets. They achieved up to 91% f1-score and concluded that domain-specific word embedding has an acceptable performance and it is suitable for unbalanced datasets.

Nobata et al. (2016) aimed to detect abusive language using pre-trained word embeddings on two domains (finance and news) and regression model classifier, they achieved 60.2% and 64.9% f1-score, respectively. The results showed that Google Word2Vec provides 5% better performance on both domains. While deep learning techniques were employed in (Badjatiya et al. 2017) to extract embedding features from hate speech text and then used a decision tree model for classification. They accomplished 93% f1-score using random embedding initialization that is fed to LSTM to construct word embedding features. The results proved that domain-specific embedding can provide a better representation of hate words such as "racist" or "sexist" words because it can extract the meaning of frequently used terms by the hate community.

A systematic review of the up-to-date studies related to hate speech detection and fake news of Ethiopian languages summarized related research (Demilie and Olalekan Salau 2022). The authors make a comparative analysis of the contributions and methodologies. The used method varies for both feature extraction and classification stages. They concluded that deep learning

outperforms machine learning classifiers. Furthermore, using a combination of deep learning and machine learning approaches provide a better result on a balanced dataset.

Previously mentioned studies confirmed that domain-specific-based detection is a promising feature extraction method in different domains. The hate speech domain is one of the domains that need deep studies and more effort to reach satisfactory results, which is the main goal of this study compared to the state-of-the-art solution. Furthermore, our proposed solution exploits the confirmed result of Demilie and Olalekan Salau (2022) examined the performance of using a deep learning model classifier with domain-specific word embedding features, which are not yet been explored in the literature on hate speech problems.

BERT language model is employed in different fields as it provides state-of-the-art solutions. The authors of (Devlin et al. 2018) looked into the BERT model's performance on a variety of NLP tasks. On 11 of these tasks, the model accomplished state-of-the-art results. It improved the performance by 7.7 points in the General Language Understanding Evaluation (GLUE) benchmark (Wang et al. 2018), 4.6 points in Multi-Genre Natural Language Inference (MultiNLI) (Williams, Nangia, and Bowman 2017), and 1.5 to 5.1 points in the SQuAD various versions of question answering tests (Rajpurkar et al. 2016).

BERT is also used in a shared task to detect offensive language (Pelicon, Martinc, and Kralj Novak 2019; Zhu, Tian, and Kübler 2019, 2019). Zhu, Tian, and Kübler (2019) fine-tuned BERT model for this task and came in third place among competitors. They used $13, 240$ tweets to train the algorithm and achieved 83.88% f1-score in classifying each tweet as offensive or not. While Mozafari, Farahbakhsh, and Crespi (2020) investigated the performance of using the BERT language model on a multi-class hate speech problem. They utilized a BERT basis model and a variety of classifiers, including CNN, which provided the highest f1-score, which is 92%.

From the previous review, it has been clarified that the problem of hate speech has been addressed using different methodologies. The domain-specific-based features are not explored enough, and the literature still has this gap. BERT is also one of the recent techniques that is not experminted yet on the hate speech problem as a binary classification task.

Methodologies

This section describes the proposed methodologies to handle the detection of hate speech. Mainly, there are two approaches used in this study seeking to find the best classification performance and to compare the results with current solutions. Figure 1 shows the block diagram of the steps of the experiment, which describes the flow of the proposed solutions and how we finally got their results.
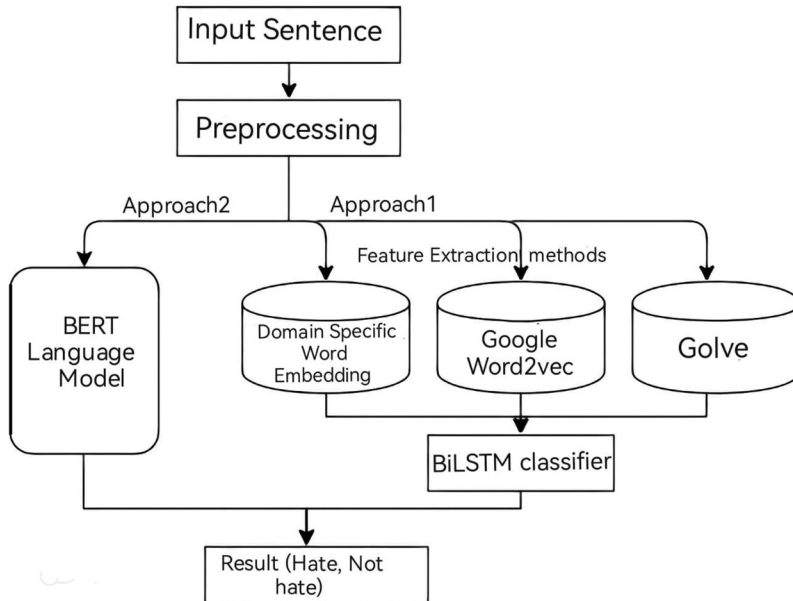
Figure 1. Block diagram of the experiments.

Approach 1: Domain-Specific Embedding Features with BiLstm Based Deep Model Classifier

The reason for using domain-specific (hate domain) word embedding is to construct the vector that represents a closer meaning of hate terms and abbreviations (e.g., black). The classifier used in this approach is a deep model constructed from bidirectional LSTM to preserve long dependencies of the input text from both directions. The following steps describe the first approach in detail:

Data Collection: The goal of this step is to build a large data corpus to be utilized for embedding extraction. The data collected consists of 1,048,563 sentences from available hate speech datasets from (Davidson et al. 2017; Founta et al. 2018; Golbeck et al. 2017; Waseem and Hovy 2016), and (Waseem 2016), in addition to a dataset we collected in this study from Twitter using commonly known hate keywords (e.g., n*gga, f*ck, and s*ave) from a pre-defined lexicon includes hate speech words and expressions, called (Inc, HateBase 2020), also from accounts who have an explicit hate content in their tweets or usernames (e.g., @Na***e_and_Race), or share hate words or phrases in hate hashtags.

Pre-processing: The pre-processing stage was performed to remove any non-meaningful words and symbols such as (stop words and punctuation), stop words are excluded because the presence or absence of these words is not important to extract the meaning of the word in case we are looking for the

hate meaning as the stop words used in both contexts (hate, non-hate) and it never used by the hate community as a hate word. We normalized the textual data by lower casing the words, and handle negation by converting "can't" to "cannot." However, the misspelling is excluded from the data cleaning phase because most of the hate words (e.g., f*k) are misspelled or abbreviated in purpose to avoid detection. Stemming removes prefixes and suffixes of the word. The normalization phase is excluded from to comprise some hate code words such as "blacks" which mostly refer to a race while the word "black" refers to a color.

Feature Extraction: from the literature, distributional-based features (e.g., word embedding) are recommended features of the words. A domain-specific embedding vector that represents the co-occurrence statistics of the word with its surrounding context is used to extract the word's meaning. Genism provides a word2vec library for this purpose, the parameters of the training model are the type of the model, which is Continuous Bag of Word (CBOW) because it performs slightly better than the Skip-Gram model (SG), with a window size = 5 that represents the number of surrounding words, and vector size = 300. The output of this stage is the embeddings for each word in the vocabulary to be used as a feature in the classification model. We build our own domain-specific hate speech word2vec model from hate speech domains named HSW2 V by collecting about 1 Million corpus from hate domains on Twitter and compare it with General purpose models (e.g., Google Word2vec and Glove).

Classification: A deep sequential model structured by three layers is used for classification. The first layer is the Bidirectional CuDNNLSTM, which is a fast LSTM implementation backed by a GPU-accelerated library of primitives for deep neural networks (CuDNN) (Abadi et al. 2016; CUDA® 2020). Using BiLSTM maintains both forward and backward data of the input sentence. This makes the bidirectional model more familiar in context, but it consumes more computation time, but we used CuDNNLSTM for accelerating the process on GPU. The second layer is a dense layer with a linear activation function chosen using grid search among other activation functions (relu, sigmoid, and none). The third layer is also a dense layer with a sigmoid activation function. The model compilation was performed with Binary Cross-Entropy Loss and optimization and Adam optimizer. The data is split into three categories: 60% training, 20% testing, and 20% validation. The batch size is 256, and the training was done across 10 epochs.

Approach 2: BERT Language Model

The second experiment was carried out using BERT, a pre-trained language model that had been fine-tuned for our objective. After a pre-trained model has been trained on a large generic text, fine-tuning is the process of training

its on application-specific content. With the use of its embedding vectors, BERT encodes the input text. We used BERT for sequence classification model, which comprises a neural-network layer for classification.

The initial stages of the BERT model convert the input sentence to tokens. The token embedding vector is created by adding the token, segment, and position embeddings together. For sentence classification, BERT uses [CLS] short for classification, which is a unique token placed at the beginning of the sentence tokens to indicate the starting position of the classification task; in other words, the starting position of the fully connected layer to the last encoder layer, and finally to the softmax layer.

BERT released different versions that have different properties based on the used language (e.g., Chinese, English, and Multilingual), the alphabet (i.e., Cased and Uncased), and the size of the layer structure (i.e., BERT-Base and BERT-Large). The BERT-Base model has 12 Transformer layers, each with 12 self-attention heads, and a total of 768 hidden states. The BERT-Large model has 24 transformer layers, 16 self-attention heads, and a total of 1024 hidden layers. For training testing, the model parameters are LEARNING RATE $= 2e - 5$, NUM TRAIN EPOCHS = 3.0, and BATCH SIZE = 16, 8 which are the parameter values recommended by the literature for sequence classification tasks.

Experiment and Results

This section includes a detailed description of the used datasets and technical details of each step in both approaches.

Datasets

We tested both approaches on three available datasets: Davidson-ICWSM (Davidson et al. 2017) dataset, Waseem-EMNLP (Waseem 2016), and Waseem-NAACL (Waseem and Hovy 2016) datasets and compare it with (Gupta and Waseem 2017) results who used Hate Speech Word2Vec trained on 1 billion corpus size and LR classifier. The details of the datasets are shown

Table 1. Datasets description.

Dataset	Original labels	Number of hate	Number of non-hate	Total
Davidson-ICWSM (Davidson et al. 2017)	Hate speech, offensive, and neither	20620	4163	24783
Waseem-EMNLP (Waseem 2016)	Racism, sexism, both, and neither	1059	5850	6909
Waseem-NAACL (Waseem and Hovy 2016)	Sexist, racist, and neither	5406	11501	16910
Balanced Combined	Racism, sexism, and neither, both	16260	16260	32520

in Table 1. Waseem and Hovy (2016) listed a number of criteria for identifying hate speech, including using a sexist or racial slur, attacking a minority, or promoting but not explicitly using hate speech or violent crime, among others, while (Davidson et al. 2017)'s study defined it as a language that is used to express hatred to a specific group or is meant to be derogatory, to humiliate, or to insult an individual of the group. The authors excluded the offensive language from their definition, the reason attributed is that offensive language is less hateful and more frequent use by the users, thus it should not be considered as hate.

However, according to (Fortuna and Nunes 2018), hate speech could use offensive language but not necessary, which we agree with because frequent use of hate words does not mean that it should be socially acceptable, and no need to detect them and for this reason, we combined hate and offensive classes in their dataset to be hate class. Both of (Waseem and Hovy 2016) and (Davidson et al. 2017) definitions do not conflict with each other, and they agreed on the general hate speech definition. Thus, collapsing the labels and combining the datasets does not conflict with the general hate speech definition. Waseem's datasets have different classes, mainly racism, sexism, and neither and since we are handling the hate speech detection from neutral as to the best of our knowledge, there is no yet the state-of-the-art solution for this problem as a binary classification task, and to compare it with (Gupta and Waseem 2017), we collapsed the classes into two classes because both of race and sex are types of hate speech as follows: nolistsep

- Racism and sexism as a hate class.
- Neither as a non-hate class.

For Davidson's dataset, both offensive and hate are considered as different levels of hate so we collapsed them as offensive and hate as a hate class and neither as a non-hate class. We also combined all the previously mentioned datasets in one dataset to assess the model performance on the largest possible diverse dataset that is balanced according to the lowest class number by randomly selecting a similar number of examples for each class and the number of classes specified according to the lowest class in the combined dataset. The dataset combining offers us a hugely diverse set of data to assess the deep model performance as it is known that deep models perform better on large training data.

HSW2 V and BiLstm Based Deep Model

For the first experiment, we investigated the performance of using Hate Speech Word2Vec (HSW2 V) as features, and the bidirectional LSTM-based deep model as a classifier. We compared our domain-specific embedding features (HSW2 V) performance with domain-agnostic embedding models, which are GloVe, and Google Word2Vec word embeddings. We also compare

Table 2. Details description of embedding models.

Methods	Dimension	Trained on data of size	Pretrained on Platform
GoogleNews-vectors-negative[a] (Mikolov et al. 2013)	300	3 billion words	Google News
glove.6B. 300d[b] (Pennington, Socher, and Manning 2014)	300	6B tokens, 400K vocab	Wikipedia 2014
glove.twitter.27B.200d [b] (Pennington, Socher, and Manning 2014)	200	2B tweets, 27B tokens, 1.2M vocab	Twitter
W2V-Hate (Gupta and Waseem 2017)	300	1 billion documents	Twitter
Hate speech Word2Vec (HSW2V)	300	1,048,563 sentences, 116955 vocab, uncased	Twitter, hate websites

a. https://code.google.com/archive/p/word2vec/
b. https://nlp.stanford.edu/projects/glove/

results with domain-specific Hate Word2Vec (W2 V-Hate) by (Gupta and Waseem 2017) study. The details of each word embedding model are mentioned in Table 2.

The model performance is reported using weighted precision, recall, AUC, and f1-score to consider the class imbalance. The f1-score is also reported for each class separately to have a clear insight into the classifier performance on each class. The results are shown in Table 3.

As shown in Table 3, for each dataset, the maximum attained performance across different features is underlined, while the best performance among different classifiers is in **BOLD**. From a feature standpoint, we compare all of the embeddings for both domain-agnostic and domain-specific embeddings using the same classifier (BiLSTM deep model). HSW2 V, as shown in the table, outperforms all domain agnostic embedding models (Google Word2vec, GLoVe), considering the large range of corpus sizes (our corpus is 1 M, other models are at least 2B), HSW2 V could slightly outperforms domain agnostic approaches. The other variable that we consider in our comparison is the classification approach (LR by (Gupta and Waseem 2017), and BiLSTM-based deep model) and assess their performance with embedding features. The results show that the deep model surpasses the LR classifier. To show the deep model performance on a balanced dataset and to overcome the class imbalance influence on the deep model, we evaluate the proposed model on the combined dataset. This gives a decent sense of the performance, as well as the best result that our proposed approach can produce.

BERT Language Model

The BERT language model was used in the second experiment because it performs well across the board in NLP applications. BERT for sequence classification was implemented and fine-tuned using datasets. Table 4 summarizes the findings of the testing evaluation.

Table 3. Results of bidirectional LSTM-based deep model on the datasets.

Source	Machine Learning Approach	Methods	Dataset	P	R	f1-score (hate)	f1-score (non-hate)	f1-score	AUC
Gupta and Waseem (2017)	LR	Hate W2V(300)	Davidson ICWSM	0.91	0.91	-	-	0.9120	0.8400
			Waseem EMNLP	0.84	0.86	-	-	0.8440	0.6380
			Waseem NAACL	0.76	0.77	-	-	0.7500	0.6790
Our proposed deep model	BiLSTM Deep model	GoogleNews-vectors- negative300	Davidson ICWSM	0.94	0.94	0.9679	0.8457	0.9473	0.9132
			Waseem EMNLP	0.91	0.91	0.6737	0.9484	0.9033	0.7697
			Waseem NAACL	0.80	0.80	0.6805	0.8542	0.7990	0.7654
			Combined balanced	0.94	0.94	0.9365	0.9376	0.9371	0.9370
		GloVe.6B.300d	Davidson ICWSM	0.94	0.94	0.9646	0.8310	0.9421	0.9073
			Waseem EMNLP	0.90	0.91	0.6631	0.9463	0.9028	0.7792
			Waseem NAACL	0.80	0.80	0.6794	0.8569	0.8002	0.7634
			Combined balanced	0.94	0.94	0.9414	0.9413	0.9414	0.9414
		GloVe.Twitter. 27B.200d	Davidson ICWSM	0.95	0.95	0.9676	0.8347	0.9453	0.8927
			Waseem EMNLP	0.91	0.90	0.6917	0.9399	0.9018	0.8324
			Waseem NAACL	0.80	0.80	0.6911	0.8503	0.7994	0.7738
			Combined balanced	0.94	0.94	0.9381	0.9394	0.9388	0.9388
		HSW2V(300)	Davidson ICWSM	0.95	0.95	0.9703	0.8551	**0.9509**	0.9162
			Waseem EMNLP	0.91	0.91	0.6928	0.9469	**0.9079**	0.8094
			Waseem NAACL	0.81	0.81	0.7055	0.8634	**0.8129**	0.7831
			Combined balanced	0.94	0.94	0.9428	0.9439	**0.9434**	0.9434

Table 4. BERT for sequence classification hate speech experiment results (base-large).

Methods	Datasets	P	R	f1-score (Hate)	f1-score (non- Hate)	f1-score	AUC
BERT Base	Davidson- ICWSM	0.96	0.96	0.98	0.89	0.962	0.9309
	Waseem- EMNLP	0.92	0.92	0.7654	0.9541	**0.9216**	0.8455
	Waseem- NAACL	0.85	0.85	0.7612	0.8881	0.8472	0.8227
	Combined Balanced	0.95	0.95	0.9543	0.9552	0.9547	0.9547
BERT Large	Davidson- ICWSM	0.96	0.96	0.9788	0.8924	**0.9646**	0.9345
	Waseem- EMNLP	0.91	0.91	0.6939	0.9458	0.9103	0.8371
	Waseem -NAACL	0.85	0.85	0.7643	0.8937	**0.8521**	0.823
	Combined Balanced	0.96	0.96	0.962	0.9625	**0.9623**	0.9623

The performance of the BERT classifier was highly acceptable and desirable. In addition, for datasets of greater size, BERT Large surpasses BERT Base with a pretty similar result to that of BERT Base. Due to the computational requirements for BERT large, most current research avoids utilizing it. However, the overall performance of BERT overcomes all of the embedding models with the deep model that we proposed as the first approach due to its large training corpus size during the training of the model.

Discussion

For the feature extraction stage, we evaluate the influence of using domain-specific word embedding of hate speech, which helps the model expose the most used terms, abbreviations, and intentional spelling mistakes by the community who post in a given domain (Badjatiya et al. 2017). This is the main reason to investigate domain-specific embedding with deep models. We applied word similarity, which finds the closest word to the input word according to the cosine distance between them. Table 5 shows the result of applying word similarity to an intentionally misspelled word that is commonly used by the hate community (fc*). As shown in the table, domain-agnostic embedding models failed to retrieve similar words, while our HSW2 V was able to retrieve other intentionally misspelled words that were close in meaning to the input word. Finally, this study was limited by the data size during the training stage, it is obvious that the corpus size was not enough for the first approach experiment to get a deeper sentiment for the word.

The results of the first experiment of using the first approach (Table 3) showed that using a domain-specific embedding model (HSW2 V) was very competitive to the domain agnostic embedding models (Word2Vec, GloVe) although there is a huge difference between the corpus size, it is 1 M for

Table 5. Word similarity of misspelled hate word fc*.

The word	Word2Vec	Glove	HSW2V
fc* (misspelled)	hahah,lmfao, questlove_@, u_ppl, hahahahahahaha, JeremyShockey_@, ummmmm, ROTFLMAO, Awright, freaken	h?ndbold, nordsj?lland, k? benhavn,br?ndby, hik, parken, s?nderjyske, genêlerbirli?i, 1972/73, iwr, al-hilal	fu*, lmfaoo, shii, fucc, dese, lmaooooo, dats, noooo, lmfaoooo, ay

▾ Misspellings

```
[31] 'ni♦ger' in tokenizer.vocab # Common Hate Word
     False

[32] 'f♦k' in tokenizer.vocab # Another common Hate Word
     True

[33] 'f♦' in tokenizer.vocab # misspelled hate word
     False
```

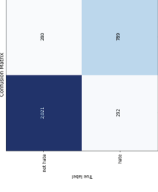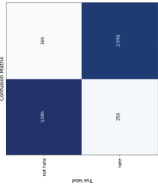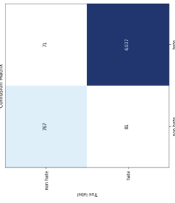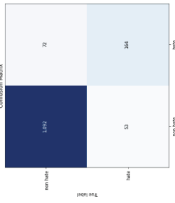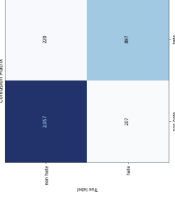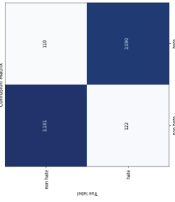**Figure 2.** BERT Base vocabulary search for misspelling and hate term.
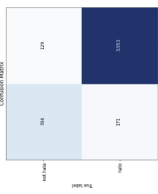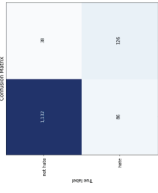
HSW2 V and at least 2B for domain agnostic embedding models taking into consideration that the classifier is the same (BiLSTM deep model), which confirmed that domain-specific word embeddings outperform domain-agnostic word embedding models, because it is more knowledgeable about the hate domain, while domain-agnostic are trained on books and Wikipedia, which rarely have hate community context.

From the classifier perspective, we compared our BiLSTM deep model with LR experiments by (Gupta and Waseem 2017), the results showed that the BiLSTM-based deep model outperforms the LR classifier, and the BiLSTM-based deep model improves the performance with at least 5%.

In the second approach, we used BERT model on the hate speech binary classification task. Table 4, reports the result of experiments on both Base and Large models. Because the BERT model is deeply bidirectional and trained on huge data sets, it outperforms all other distributional-based embeddings, including domain-agnostic (e.g., Google Word2Vec and GloVe) and domain-specific (e.g., HSW2 V). BERT also has an intuitive training procedure for its vocabulary as it includes sub-words instead of complete words. Although the simple training procedure of domain-specific embeddings, the performance was not too low in comparison with BERT. Domain-specific embeddings overcome BERT model embedding in that it includes intentional misspellings and commonly hate words that BERT fails to retrieve when we searched about specific words in Bert Base vocabulary as shown in Figure 2, and this is because BERT also trained on books and Wikipedia, which rarely includes these words.

Finally, Table 6 shows the confusion matrix of HSW2 V(300), the Bidirectional LSTM deep model, which correctly classifies the most prevalent labels in the datasets. It also shows BERT language model performance graphically for each of the FP, FN, TP, and TN on the same datasets.

Table 6. Confusion matrix of HSW2 V and BERT.

Datasets/model	Davidson-ICWSM	Waseem-EMNLP	Waseem-NAACL	Balanced Dataset
HSW2V(300), Bidirectional LSTM deep model				
BERT Large				

Figure 3. Result of applying LIME on document actual label=1 and predicted=1.

BERT Interpretation Using LIME

To gain more knowledge about BERT model performance in the classification task of this study, we applied LIME (Ribeiro, Singh, and Guestrin 2016). LIME stands for Local Interpretable Model-agnostic Explanations, a strategy for understanding the model by modifying data samples and seeing how the predictions change by looking at internal properties and how they connect to specific predictions. We employed LIME on the Waseem-EMNLP dataset and BERT-based model. We reported different cases as follows:

- **Case 1: True Positive**

LIME highlights the words that contributed more in classifying the sentence with a different color for each class, Figure 3 shows the case in which the sentence's actual and predicted class is hate, and which words contribute in classifying this sentence as hate. The color intensity increased according to the word more contributed to the predictions such as n*gga.
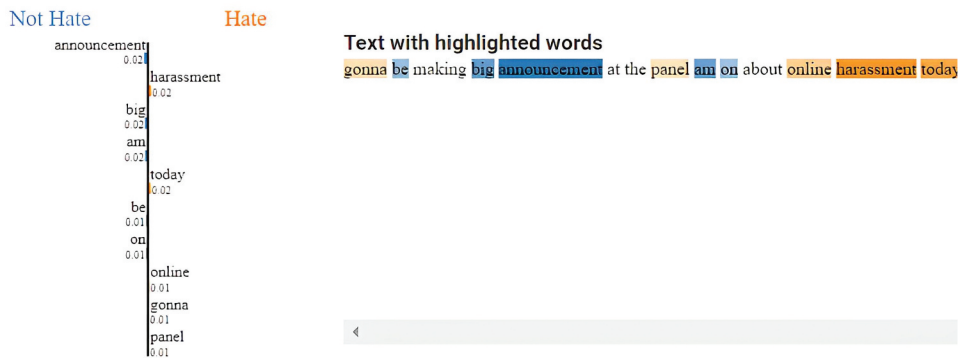

Figure 4. Result of applying LIME on document actual label=0 and predicted=0.

• **Case 2: True Negative**

Figure 4 shows the case in which the sentence actual and predicted class is not hate, and which words contribute to classify this sentence as not hate with darker blue colors such as announcement, while the words with orange color contributed more to classify the sentence to be hate; however, in this case, the weight of the blue-colored words is more than the orange-colored words, thus the classifier predict this sentence as not hate which agree with human sense.

• **Case 3: False Positive**

We also apply LIME to analyze error classifications. Figure 5 shows the case in which the sentence actual class is not hate but the predicted class is hate, and which words contribute in classifying this sentence as hate, which is a feminazi word that influences on the classifier to predict it as hate.

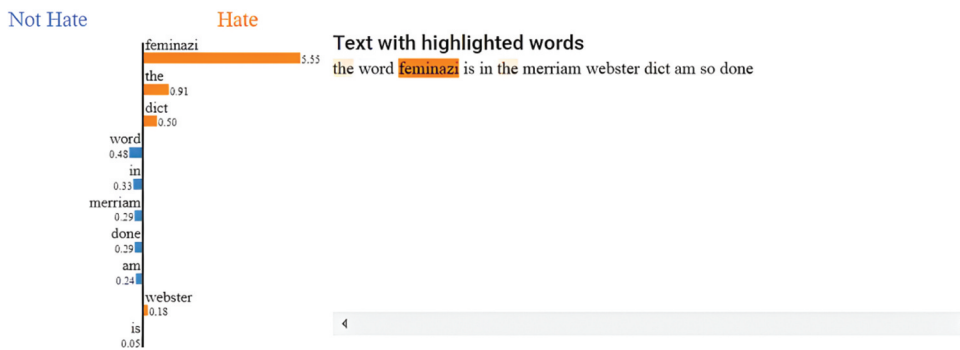**Figure 5.** Result of applying LIME on document actual label=0 and predicted=1.

• **Case 4: False Negative**

Figure 6 shows the case in which the sentence actual class is hate but predicted class is not hate, and which words contribute in classifying this sentence as not hate. It seems that in this situation, the classifier is more
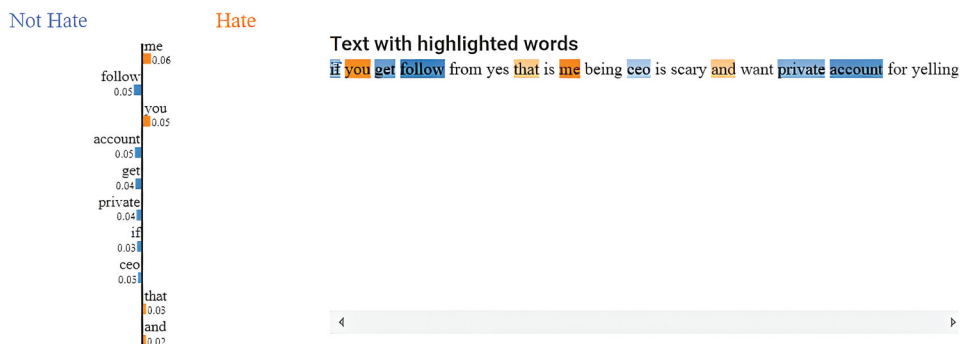

Figure 6. Result of applying LIME on document actual label=1 and predicted=0.

accurate than the human annotator as this sentence is not a hate sentence as it is obvious from the observed intent of the context writer.

Conclusion and Recommendation

To conclude, BERT design provides an appropriate feature extraction and classification procedure for hate speech detection. BERT combines the benefits of domain-agnostic and domain-specific word embedding by training the model on vast data and then adding an extra layer to train on domain-specific data (fine-tuning). BERT also saves effort and time in building an embedding model from scratch. However, domain-specific word embedding overcomes the BERT model in that it can detect hate terms and abbreviations and intentionally misspell meanings. One of the challenges that we faced during the experiments is that identifying hate speech is a highly subjective task in which the machine may also encounter some difficulties in the detection. The impact of this paper is mainly proven by experiments that BERT Model provides acceptable performance in dealing with a hate speech problem. This paper could be used for comparison for future work studies, especially after changing hate speech policies on Twitter. This study can be extended to detect multi-class hate speech for future work.

Disclosure statement

ORCID

Hind Saleh 🆔 http://orcid.org/0000-0002-8208-3510

References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. 2016. TensorFlow: Large-scale machine learning on heterogeneous systems. *arXiv preprint arXiv:1603.04467*. Accessed 2015. https://www.tensorflow.org/api_docs/python/tf/compat/v1/keras/layers/CuDNNLST.

Agarwal, S., and A. Sureka. 2015. Using knn and svm based one-class classifier for detecting online radicalization on twitter. In *International Conference on Distributed Computing and Internet Technology*, Cham, 431–42. Springer.

Ailem, M., A. Salah, and M. Nadif. 2017. Non-negative matrix factorization meets word embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Shinjuku, Tokyo, Japan, 1081–84.

Al-Azani, S., and E.S.M. El-Alfy. 2017. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short Arabic text. *ANT/SEIT* 359–66. doi:10.1016/j.procs.2017.05.365.

Albadi, N., M. Kurdi, and S. Mishra. 2018. Are they our brothers? Analysis and detection of religious hate speech in the Arabic Twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Los Alamitos, CA, USA, 69–76. IEEE.

Badjatiya, P., S. Gupta, M. Gupta, and V. Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, Perth, Australia, 759–60.

Bengio, Y., R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3 (Feb):1137–55.

Chen, L., R. Song, M. Liakata, A. Vlachos, S. Seneff, and X. Zhang. 2015. Using word embedding for bio-event extraction. In *Proceedings of BioNLP 15*, 121–26. Beijing, China: Association for Computational Linguistics, July. https://doi.org/10.18653/v1/W15-3814.

Clement, J. 2019. *U.S. teens hate speech social media by type 2018 l Statistic*, October. https://www.statista.com/statistics/945392/teenagers-who-encounter-hate-speech-online-social-media-usa/.

CUDA®, N. V. I. D. I. A. 2020. *NVIDIA cuDNN | NVIDIA Developer*, Accessed November 2019. https://developer.nvidia.com/cudnn.

Davidson, T., D. Warmsley, M. Macy, and I. Weber. 2017. "Automated hate speech detection and the problem of offensive language." In *Eleventh international aaai conference on web and social media*, Canada.

Demilie, W. B., and A. Olalekan Salau. 2022. Detection of fake news and hate speech for Ethiopian languages: A systematic review of the approaches. *Journal of Big Data* 9 (1):1–17.

Devlin, J., M.W. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:181004805* 11:512–515.

Djuric, N., J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, Florence, Italy, 29–30.

Duggan, M. 2017. *Online harassment 2017*. Pew Research Center: Internet, Science & Tech. Accessed January 25, 2023. https://policycommons.net/artifacts/617798/online-harassment-2017/1598654/.

Ferrara, E., W.Q. Wang, O. Varol, A. Flammini, and A. Galstyan. 2016. Predicting online extremism, content adopters, and interaction reciprocity. In *International conference on social informatics*, Cham, 22–39. Springer.

Fortuna, P., and S. Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)* 51 (4):1–30.

Founta, A. M., C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*, Stanford, California, USA.

Gambäck, B., and U. Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, Vancouver, BC, Canada, 85–90.

Gialampoukidis, I., G. Kalpakis, T. Tsikrika, S. Papadopoulos, S. Vrochidis, and I. Kompatsiaris. 2017. Detection of terrorism-related twitter communities using centrality scores. In *Proceedings of the 2nd International Workshop on Multimedia Forensics and Security*, Bucharest, Romania, 21–25.

Gibert, O. D., N. Perez, A. Garcıa-Pablos, and M. Cuadros. 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:180904444* 11–20.

Golbeck, J., Z. Ashktorab, R. O. Banjo, A. Berlinger, S. Bhagwan, C. Buntain, P. Cheakalos, A. A. Geller, R. Kumar Gnanasekaran, R. Rajan Gunasekaran, et al. 2017. A large labeled

corpus for online harassment research. In *Proceedings of the 2017 ACM on web science conference*, Troy, New York, USA, 229–33.

Gupta, S., and Z. Waseem. 2017. *A comparative study of embeddings methods for hate speech detection from tweets*. Association for Computing Machinery.

Hartung, M., R. Klinger, F. Schmidtke, and L. Vogel. 2017. Identifying right-wing extremism in German Twitter profiles: A classification approach. In *International conference on applications of natural language to information systems*, Cham, 320–25. Springer.

Hochreiter, S., and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* 9 (8):1735–80.

Inc, HateBase. 2020. *HateBase*. https://hatebase.org/.

Jaki, S., and T. De Smedt. 2019. Right-wing German hate speech on Twitter: Analysis and automatic detection. *arXiv preprint arXiv:191007518*, 31.

League, A.D. 2019. *Online hate and harassment: The American experience*, Pew Research Center: Internet, Science & Tech. Accessed December 25, 2022. https://policycommons. net/artifacts/617798/online-harassment-2017/1598662/.

Lilleberg, J., Y. Zhu, and Y. Zhang. 2015. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, Beijing, China, 136–40. IEEE. doi:10.1109/ICCI-CC.2015.7259377.

Liu, A. 2018. Neural network models for hate speech classification in tweets. PhD diss., Dept. Arts Sci., Harvard, Cambridge, MA, USA. Retrieved from https://dash.harvard.edu/handle/ 1/38811552.

Ltd, We Are Social. 2020. *Digital 2020*. Accessed November 2008. https://wearesocial.com/ digital-2020.

Magu, R., K. Joshi, and J. Luo. 2017. Detecting the hate code on social media. In *Eleventh International AAAI Conference on Web and Social Media,* Canada, 11:608–611. doi:10.1609/ icwsm.v11i1.14921.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:13013781* 11:608–611.

Mozafari, M., R. Farahbakhsh, and N. Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on BERT model. *Plos One* 15 (8):e0237861.

Nobata, C., J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, Montreal, Quebec, Canada, 145–53.

Pelicon, A., M. Martinc, and P. Kralj Novak. 2019. Embeddia at SemEval- 2019 Task 6: Detecting hate with neural network and transfer learning approaches. In *Proceedings of the 13th international workshop on semantic evaluation*, Minneapolis, Minnesota, USA, 604–10.

Pennington, J., R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, 1532–43.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:180205365*.

Pitsilis, G. K., H. Ramampiaro, and H. Langseth. 2018. Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence* 48 (12):4730–42.

Rajpurkar, P., J. Zhang, K. Lopyrev, and P. Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:160605250* 2383–2392.

Ribeiro, M. H., P. H. Calais, Y. A. Santos, V. A. Almeida, and W. Meira Jr. 2017. "Like sheep among wolves": Characterizing hateful users on twitter. *arXiv preprint arXiv:180100317*.

Ribeiro, M. T., S. Singh, and C. Guestrin. 2016. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, California, USA, 1135–44.

Sienčnik, S. K. 2015. Adapting word2vec to named entity recognition. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODAL- IDA 2015)*, Vilnius, Lithuania, 239–43.

Smedt, D., G. D. P. Tom, and P. Van Ostaeyen. 2018. Automatic detection of online jihadist hate speech. *arXiv preprint arXiv:1803.04596*, 31. Computational Linguistics & Psycholinguistics.

Socher, R., Y. Bengio, and C. D. Manning. 2012. Deep learning for NLP (without magic). In *Tutorial Abstracts of ACL 2012*, 5. Jeju Island, Korea: Association for Computational Linguistics.

Tang, D., F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin. 2014. Learning sentiment specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland, USA, 1555–65.

Wang, J., K. R. L. Liang-Chih Yu, and X. Zhang. 2016. Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Berlin, Germany, 225–30.

Wang, A. S., J. Michael, F. Hill, O. Levy, and S. R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:180407461* 353–355.

Wang, P., Y. Qian, F. K. Soong, H. Lei, and H. Zhao. 2015. Part-of-speech tagging with bidirectional long short-term memory recurrent neural network. *arXiv preprint arXiv:151006168*.

Waseem, Z. 2016. 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*, Austin, 138–42.

Waseem, Z., and D. Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, San Diego, California, 88–93.

Weber, A. 2009. *Manual on hate speech*. Strasbourg, France: Council Of Europe.

Wei, Y., L. Singh, and S. Martin. 2016. "Identification of extremism on Twitter." In *2016 IEEE/ ACM international conference on advances in social networks analysis and mining (ASONAM)*, Davis, California, 1251–55. IEEE.

Williams, A., N. Nangia, and S. R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:170405426* 1: 1112–1122.

Yonghui, W., X. Jun, Y. Zhang, and X. Hua 2015. Clinical abbreviation disambiguation using neural word embeddings. In *Proceedings of BioNLP*, Beijing, China, *15*, 171–76. doi:10.18653/v1/W15-3822.

Zhang, Z., D. Robinson, and J. Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, 745–60. Cham: Springer.

Zhu, J., Z. Tian, and S. Kübler. 2019. Um-iu@ ling at semeval-2019 task 6: Identifying offensive tweets using BERT and svms. *Proceedings of the 13th International Workshop on Semantic Evaluation*, 788–795. Minneapolis, Minnesota, USA: Association for Computational Linguistics.