‘The Enemy Among Us’: Detecting Cyber Hate Speech with Threats-based Othering Language Embeddings

WAFA ALORAINY, Cardiff University, UK
PETE BURNAP, Cardiff University, UK
HAN LIU, Cardiff University, UK
MATTHEW L. WILLIAMS, Cardiff University, UK

Offensive or antagonistic language targeted at individuals and social groups based on their personal characteristics (also known as cyber hate speech or cyberhate) has been frequently posted and widely circulated via the World Wide Web. This can be considered as a key risk factor for individual and societal tension surrounding regional instability. Automated Web-based cyberhate detection is important for observing and understanding community and regional societal tension - especially in online social networks where posts can be rapidly and widely viewed and disseminated. While previous work has involved using lexicons, bags-of-words or probabilistic language parsing approaches, they often suffer from a similar issue which is that cyberhate can be subtle and indirect - thus depending on the occurrence of individual words or phrases can lead to a significant number of false negatives, providing inaccurate representation of the trends in cyberhate. This problem motivated us to challenge thinking around the representation of subtle language use, such as references to perceived threats from ‘the other’ including immigration or job prosperity in a hateful context. We propose a novel ‘othering’ feature set that utilises language use around the concept of ‘othering’ and intergroup threat theory to identify these subtleties, and we implement a wide range of classification methods using embedding learning to compute semantic distances between parts of speech considered to be part of an ‘othering’ narrative. To validate our approach we conducted two sets of experiments. The first involved comparing the results of our novel method with state of the art baseline models from the literature. Our approach outperformed all existing methods. The second tested the best performing models from the first phase on unseen datasets for different types of cyberhate, namely religion, disability, race and sexual orientation. The results showed F-measure scores for classifying hateful instances obtained through applying our model of 0.81, 0.71, 0.89 and 0.72 respectively, demonstrating the ability of the ‘othering’ narrative to be an important part of model generalisation.

CCS Concepts: • **Information systems** → *World Wide Web*; • **Social and professional topics**;

Additional Key Words and Phrases: Hate speech, Cyber Hate, Twitter, Social Media, Language modelling, Typed Dependency, Embedding Learning, Text Classification, Machine Learning

ACM Reference Format:
Wafa Alorainy, Pete Burnap, Han Liu, and Matthew L. Williams. 2018. ‘The Enemy Among Us’: Detecting Cyber Hate Speech with Threats-based Othering Language Embeddings. *ACM Trans. Web* 9, 4, Article 39 (March 2018), 26 pages. https://doi.org/0000001.0000001

Authors’ addresses: Wafa Alorainy, Cardiff University, School of Computer Science and Informatics , Cardiff, Wales, CF24 3AA, UK; Pete Burnap, Cardiff University, School of Computer Science and Informatics , Cardiff, Wales, CF24 3AA, UK; Han Liu, Cardiff University, School of Computer Science and Informatics , Cardiff, CF24 3AA, UK; Matthew L. Williams, Cardiff University, School of Social Sciences, Cardiff, Wales, CF10 3WT, UK.

1559-1131/2018/3-ART39 $15.00
https://doi.org/0000001.0000001

1 INTRODUCTION

While the benefits of online social media are enabling distributed societies to be connected, one disadvantage of the technology is the ability for hateful and antagonistic content, or cyberhate, to be published and propagated [64]. As people increasingly communicate through Web-enabled applications, the need for high-accuracy automated cyberhate detection methods has become much higher. Several studies have shown how individuals with biased or negative views towards a range of minority groups are taking to the Web to spread such hateful messages [38, 50]. Instances of cyberhate and racist tension on social media have also been shown to be triggered by antecedent events, such as terrorist acts [9, 64].

Expressing discriminative opinions employs different language uses. For example, words might be used to convey intense dislike such as *'hate them'*; moreover, to encourage violence, an inflammatory verb could be used such as 'kill'. While these examples contain directly threatening or offensive words *(kill, hate)*, some examples contain words that, on their own, would not constitute discriminative opinions *(e.g. send them home)*. Although they do not contain explicitly hateful words, they are conveying the desire to distance different groups, within which there is an inherent promotion of discrimination and division within society, fostering widespread societal tensions[1], [65] [32]. Context in which distancing terms are used are of course important here. For instance, if two boxers are fighting and an audience member shouts "kill them", then this is a case of distancing, and discrimination lies in it, but it is in a non-societal context. There have been a number of attempts to automatically identify and quantify cyberhate by using different approaches, such as lexicons [25], syntactic [26] and semantic [35] [4] features - yet the limitation lies in classifying text that does not contain clearly hateful words and would have an impact on classification accuracy, *(e.g. send them home)*. While previous studies highlight the utility of methods capable of measuring semantic distances between words, such as embedding learning using individual words [19], and n-grams [47], this example requires an additional layer of qualitative context that sits above combinations of individual words.

Recent studies have begun to interpret the effective features for machine classification of abusive language by focusing on how language is used to convey hateful or antagonistic sentiment. 'Othering' - the use of language to express divisive opinions between the in-group ('us') and the out-group ('them') - has been identified as an effective feature [11]. The concept of 'othering' offers a potential candidate framework for the aforementioned qualitative layer capable of capturing the more subtle expressions of cyberhate such as the 'send them home' example. Anti-Hispanic speech might make a reference to border crossing or crime, anti-African American speech often references unemployment or single parent upbringing, and anti-Semitic language often refers to money, banking and the media. The use of stereotypes also means that certain types of language may be regarded as hateful even if no single word in the passage is hateful by itself [20]. In this study we develop a novel method for cyberhate classification based around (i) the use of two-sided pronouns that combine the in-group and out-group (e.g. your/our, you/us, they/we), and (ii) the use of pronoun patterns, such as verb-pronoun combinations, which capture the context in which two-sided pronouns are used (e.g. send/them, protect/us). Our hypothesis is that considering these linguistic features will provide an additional set of qualitative features that will improve the classification performance. We use these to build a feature set that we refer to as an 'othering feature set', which we use to enrich the representation of text examples of cyberhate. These features are subsequently used in combination with a paragraph embedding algorithm that infers semantic similarity between features to create a model that represents 'othering' language which is used for cyberhate classification.

[1] https://www.article19.org/wp-content/uploads/2018/06/UK-hate-speech_March-2018.pdf

Paragraph embedding algorithms aim at learning the semantic similarity of our proposed contextual features jointly with the rest of the text in the corpus. Samples that contain two-sided pronouns or pronoun patterns (e.g. verb-pronoun combinations) in a hateful or antagonisitc context are aligned in similar feature 'spaces'. This increases the probability of the machine classification method labelling any new samples exhibiting these features as cyberhate. For example, the following sentence: (***We*** *want to hang* ***them*** *all*) contains the verb *"hang"* and pronoun *them*, as well as the two-sided pronouns *we, them.* If our hypothesis is correct, and such features do indeed improve the context of the automated learning method, we would expect the sentence:***We*** *need to get* ***them*** *out* to be classified as cyberhate. This is not a sentence that would immediately flag as hateful by using existing classification methods, but is an example of where human annotators identified a threat to individual groups and communities, and therefore needs to be considered when 'taking the social mood' following trigger events. To benchmark our approach, we present results of different models that use state of the art classification algorithms and features sets from the existing literature, and compare these to our proposed method. The results show that our novel 'othering feature set' approach outperforms all existing methods. We tested the best performing models on four unseen data sets based around cyberhate targeted at religion, disability, race and sexual orientation - with F-measure results of: 0.81, 0.60, 0.89 and 0.71 respectively.

The paper is structured as follows: in Section 2 we review related works that are relevant to our technical proposal. In Section 3, we present our methods and explain the experimental steps. In Section 4, the classification results are presented and discussed. Finally, in Section 5, the contributions of this paper are summarized and some future directions are suggested for advancing this research area.

2 LITERATURE REVIEW

In this section, we review research work on cyberhate detection. In particular, we review related works that aimed to detect hate speech, as well as those works focused on sentiment or opinions that are deemed abusive [16]. [60] proposed a typology that synthesizes the sub-tasks involved in hate speech detection. This includes directed hateful abuse or general abuse, and further refines this into explicit or implicit abuse. In our study we seek to detect both direct and indirect hateful abuse, which we refer to under the broad term cyberhate. This section is composed of four sub-sections: othering language narrative surrounding cyberhate, feature set enrichment, linguistic features for cyberhate detection, and the use of embedding learning.

2.1 Othering language

The hypothesis of this study is that we can leverage linguistic features of text posted to the Web to improve the classification of cyberhate. In particular we build a theoretical framework based on leveraging the theories of *othering* and *Intergroup Threat Theory (ITT).* ITT posits that prejudice is a product of perceived realistic and symbolic threats. Realistic threats can be conceptualized in economic, physical and political terms. Such threats refer to competition over material economic group interests, including scarce resources such as jobs, houses, benefits and healthcare. Symbolic threats are based on perceived group differences in values, norms and beliefs. Out-groups that have a different viewpoint can be seen as threatening the cultural identity of the in-group [55]. Studies show that perceived threats to in-group values by immigrants and minorities are related to more negative attitudes towards these groups, unless countered by other in-group members [46]. For instance, research using ITT has recently focused on the perception of threat from Muslims in Europe [15]. This can result in 'othering' language, such as 'get them out', which represents a speech act that aims to protect resources for the in-group. The core concept is that these resources and values are threatened by the out-group, leading to anxiety and uncertainty in the in-group

[56]. The desire to protect the in-group is considered the underlying motivation responsible for negative attitudes and discriminatory behavior. *Othering* is an established construct in rhetorical narrative surrounding hate speech [42], and the 'we-they' dichotomy has previously been identified in racist discourse [69]. Othering has been used as a framework for analysing racist discourse from a qualitative perspective in previous work. For instance, [68] argued that while the 'self' or the concept of 'us' is constructed as an in-group identity, the 'other' or the concept of 'them' is constructed as an out-group identity [59]. Therefore, polarization and opposition are created by emphasizing the differences between 'us' and 'them'. This may occur, for example, through the use of language to convey positive self-representation and negative representation of the 'other' as an out-group that is undesirable [67]. In machine learning research the principle of othering has been identified by [9] as a useful feature for classifying cyberhate based on religious beliefs, specifically for identifying anti-Muslim sentiment. However, this was post-classification in the form of an effort to interpret some of the statistically effective linguistic features. It is yet to be used as a theoretical foundation of feature engineering and tested with machine classification algorithms. Therefore the literature review focuses on lexicon-based and linguistic features, as well as machine classification methods, to provide a baseline for comparison to our proposed innovation.

2.2 Lexicon-based Methods

While dictionary-based approaches generally suffer from an inability to find offensive words with domain and context-specific orientations [16], corpus-based approaches use a domain corpus to capture opinion words with preferred syntactic or co-occurrence patterns. Focusing on a theme-related lexicon, [25] generated a lexicon of sentiment expressions using semantic and subjectivity features with an orientation towards hate speech, and then used these features to create a classifier for cyberhate detection. However, their work depends on the existence of specific feature co-occurrence to decide the polarity of a specific tweet which might omit indirect/implicit hate speech. Going down to a phrase level, some work has combined sentence-level features with dictionary-based features for automatic cyberhate detection using two steps: first, the author used word features (tokens), sentence/structure features (dependency relations) and document features (document topic). A binary feature was applied that captured whether the word was being locally negated. Its value was true if a negation word or phrase was found within the four preceding words, or in any of the word's children in the dependency tree. If there was no negation word in a phrase, this intensified the hateful rating. Secondly, the author combined sentence-level features with dictionary-based features and achieved significant improvements in cyberhate polarity prediction by 4.3% compared to the baselines [66]. Similar to the previous study, their work depends on detecting the polarity of a sentence depending on the existence of specific features, which raises the same limitations of other dictionary based approaches - can the model generalise if these features are not present? [54] detected cyberhate using sentence structure - specifically patterns starting with the word 'I'. They assume that the word 'I' means that the user is talking about the emotions that he or she is feeling. Their work introduces the direction of the sentence structure rather than depending on specific words to recognize the sentence polarity. They suffered from a high false positive rate due to the model classifying sentences such as '*I hate following people*' as hateful. Indeed, phrase-level sentiment analysis is different from lexicon-based analysis in that the former focuses on the polarity of the contextual content (e.g. whether there is negation in the sentence or not), whereas the latter predicts the polarity depending on the occurrence of the words that exist in the dictionary. Another direction of building lexicons is to follow a sentiment scoring method which uses emoticons, modifiers, negations and domain specific words [3]. Despite the scoring method outperforming the baseline methods, it needs a manual scoring of words. Lexicon methods could involve the use of offensive words and slurs or negative/positive related words (e.g. emotions and negation words) as features, which might help to distinguish hate speech from other posts, yet still has a weakness in detecting the hate stereotypes when the text contains no single hateful words - an scenario we see in text containing ‘othering’ language.

2.3 Linguistic Features

One of the most basic forms of natural language processing is the Bag of Words (BoW) feature extraction approach. BoW has been successfully applied as a feature extraction method for automated detection of hate speech, relying largely on keywords relating to offence and antagonism [10, 48, 63]. However, the method suffers from a high rate of false positives, since the presence of hateful words can lead to the misclassification of tweets being hateful when they are used in a different context (e.g. ‘black’) [26]. For instance, [16] demonstrated how non-hateful content might be misclassified due to the fact that it contains words used in racist text. In contrast, they also showed that hateful instances were misclassified because they did not contain any of the terms most strongly associated with cyberhate. N-grams are features that capture consecutive words of varying sizes (from *1...n*) and have been used to improve the performance of hate speech classification by capturing context within a sentence that is lost in the BoW model [7, 12, 24, 47, 61]. Character n-grams have been shown to be appropriate for abusive language tasks due to their ability to capture variations of words associated with hate [61]. In addition, using a character n-gram based approach outperforms word n-grams due to character n-gram matrices being far less sparse than the word n-gram matrices [61], [43]. Character n-grams have been shown to be more effective if joined with additional linguistic features including gender and location [61]. Furthermore, character n-grams have been shown to improve cyberhate detection and distinguishing it from general profanity by building surface n-grams, word skip-grams, and Brown clusters features [41]. In their work, the features were used for discriminating between hate speech and profanity using an annotated data set with three labels: (1) hate speech; (2) offensive language but no hate speech ; and (3) no offensive content. The study found that a character 4-gram model was able to distinguish between hate speech and offensive language while other features were more frequently confused for offensive content [41]. Moreover, n-grams were shown to outperform the use of keyword-based methods when using self-identified hateful communities as training data for hateful speech classifiers using a sparse representation of unigrams with tf-idf weights as a feature set. The study provides evidence that using self-identified hateful comments outperformed the use of keyword-based methods by 10%-20% [52]. However, the fact that not all content in a hateful community is hateful (i.e. it is used as a form of in-group affection) shows some weakness in this approach that would lead to false positives.

Another feature is the use of a keywords dictionary, a study by [22] showed the strength of leveraging the insult key word for capturing both explicit and implicit hate speech from an unbiased corpus. They trained two weakly supervised bootstrapping (slur term learner and LSTM classifier) models simultaneously to identify hateful tweets. They found that training two models jointly identified many more hate speech texts with a significantly higher (albeit low compared to other literature) F-score (slur learner = 0.19, LSTM = 0.26, both = 0.49). This is an interesting approach to splitting the problem of explicit and implicit hate, but needs improvement to increase accuracy. Another aspect of linguistic features was studied by [14]. They focused on grammatical features (e.g. markers, place and time adverbs, questions etc). They demonstrated the effectiveness of a wide range of grammatical features to identify the main dimensions of functional linguistic variation that occur in racist and sexist Tweets. They identified 3 dimensions of linguistic variation in racist and sexist Tweets: interactive, antagonistic, and attitudinal. By applying different linguistic features, they demonstrated that there is a significant functional difference between racist and sexist Tweets, with sexists Tweets tending to be more interactive and attitudinal than racist Tweets. The study did not however focus on indirect hate using linguistic features (e.g. *'send them back'*).

[6] took a paraphrasing approach - identifying tokens that were clear indicators for hateful content by retrieving words that are most strongly related with cyberhate. They did this using nouns, named entities, and hashtags. The study examined a relatively small dataset and saw some improvements in classification accuracy. Our approach builds on this work by using the core focus on specific types of language via the othering feature set. A study conducted by [62] showed that unigram features as well as the pattern features present the highest accuracy with values respectively equal to 82.1% and 70%, whereas the semantic and sentiment features did not produce a good classification accuracy. The pattern feature set was prepared using PoS tagging as follows: when the tweet contained sentiment, a specific PoS tagged word (e.g. coward) was replaced by *Negative_ADJECTIVE*, otherwise when the tweet did not contain any sentiment words they were replaced by a simplified PoS tag. The combination of all previous features achieved an accuracy of 87% for binary classification. However, this method depends on replacing a hateful content with a specific pattern and unigram features which again leads to the possibility over overlooking implicit hateful content with no clearly hateful pattern.

In general, while previous studies addressed the difficulty of the definition of hateful language, their experiments led to better results when combining a large set of features. They showed that BoW, n-grams, part-of-speech tagging, and data preprocessing (stop word/punctuation removal) provided a significant improvement in sentiment classification among different data sets (blogs and movies) when applied as a sophisticated combination of feature sets. They also speculated that engineering features based on deeper linguistic representations (e.g. dependencies and parse tree) may improve classification results for contents on social media. Typed dependencies have been widely used for extracting the functional role of context words for sentiment classification [28, 33] and document polarity [57]. Applying typed dependencies for classifying cyberhate showed that typed dependencies consistently improved the performance of machine classification for different types of cyberhate by reducing the false negative rate by 7%, beyond the use of BoW and known hateful terms [10, 11]. Our work is different from the previous works in that we introduce a new feature set that uses the othering language patterns to detect implicit/explicit hateful content.

2.4 Text Embedding

Embedding learning is aimed at training a model that can automatically transform a sentence/word into a vector that encodes its semantic meaning. It has been shown that embedding representation is very capable of semantic learning when word vectors are mapped into a vector space, such that distributed representations of sentences and documents with semantically similar words have similar vector representations [44] [45]. Based on the distributional representation of the text, several methods of deriving word representations that are related to cyberhate and offensive language detection are explored in the following works. In general, neural network applications were shown to be capable of capturing specific semantic features from complex natural language (e.g. location [49], entity [53] and images feature [2]). For hate speech detection purposes, [19] solved the problem of high dimensionality and sparsity by applying sentence embedding (paragraph2vec). In their study, paragraph2vec, which is an extended version of Word2Vec for sentences, has been shown to outperform the BoW representation for cyberhate classification models by around 3% to 4% in F1 score. However, they limited their study to comparing the classification results with TF-BoW and TF-IDF-BoW for the same comments. Similarly, [4] compared the classification accuracy of the combination of different baselines and classifiers (Char n-gram, TF-IDF, BoW and LSTM) and found that learning embedding with gradient-boosted decision trees led to the best classification performance by 18% over state-of-the-art char/word n-gram methods. For German language processing, [35] examined different types of features (BOW, 2-grams, 3-grams, linguistics, Word2Vec, Paragraph2vec, extended 2-grams and extended 3-grams) for training logistic regression LR classifiers. The experimental results, obtained on a 75/25 split between training and test data, showed that the best performing types of features are Word2Vec and Extended 2-grams.

Word vector extraction was also applied to tweets for cyberhate classification by [21], who built a Convolutional Neural Network (CNN) model which was trained on four feature sets: character 4-grams, word vectors based on semantic information built using Word2Vec, randomly generated word vectors, and word vectors combined with character n-grams. They used a data set with the sample size of 6655 that contains a small set of abusive samples (91 for racism, 946 for sexism, 18 for both). They showed that while adding character n-grams slightly increased the precision but resulted in lower recall and F-measure; the second feature set (Word2Vec alone) performed best overall, with an F-measure of 0.78. Another use of word vector extraction was introduced by [31]who investigated the less pronounced form of sexism demonstrated online using three types of approaches: SVM, seq2seq and FastText which was set with 100 dimension. They created a dataset of tweets that exhibit benevolent sexism and classified the tweets into 'Hostile', 'Benevolent' or 'Others' class depending on the kind of sexism they exhibit. They worked with three data sets of sizes: (712 for Hostile set, 2254 for Benevolent data set and 7129 for other). They mentioned that the small sample size of their study is considered as a study limitation because of the method of gathering benevolently sexist tweets which was biased towards the initial search terms and likely missed many forms of benevolent sexism. However, the use of the FastText approach, which allows update of word vectors through back-propagation during training, even with the small data set, outperformed the others with an F1-score of 0.87.

Recently, [72] introduced a deep neural network model combining CNN and gated recurrent unit (GRU) layers, which were used to train on Word2Vec features with 300 dimensions. The results show that the classification accuracy was improved by between 1 and 13% on 7 data sets (sample size between 2435 and 24783), when compared to baselines methods. Another use of deep learning methods was introduced by [51], they explored employed an ensemble of LSTM-based classifiers to improve classification performance using mechanisms for aggregating the classifications - namely Voting and Confidence. Their deep learning architecture involves using word frequency vectorisation for implementing a series of features associated with users' behavioural characteristics, such as the tendency to post abusive messages, as input to the classifier. The word vector dimension was set to 30 to encode every word in the vocabulary used. They experimented using a dataset of approximately 16k short messages from Twitter which contain 1943 tweets labeled as Racism, 3166 tweets labeled as Sexism and 10889 tweets labeled as Neutral (i.e., tweets that neither contain sexism nor racism). They reported that the small number of tweets and the class imbalance in the dataset made the task more challenging. However, they reported that their approach outperformed the current state-of-the-art approaches.

[71] introduced the problem of unbalanced nature and the lack of discriminative features of hateful content in the typical datasets and they propose a new DNN model which is designed to capture implicit features that are potentially useful for classification. They modified the typical CNN architecture by adding the operation that involves using 'gapped window' to extract features from word embedding inputs of 300 dimensions. A skipped CNN component was functioned which was expected to extract the dependent sequences of input n-grams, and performed much better over the baselines when only hateful tweets are considered suggesting that the skipped CNNs may be more effective feature extractors for hate speech detection in very short texts such as tweets. They evaluated their approach on a group of datasets from previous studies which contain hateful samples that fall in the range between 413 and 5773. [18] also applied an LSTM classifier for classifying hate speech in Italian texts. They applied morpho-syntactical features, sentiment polarity

and word embedding lexicons as feature extraction methods leading to a 262-dimensional vector being transformed from each word. The LSTM classifier was compared with the SVM classifier using two datasets: the three-class dataset, composed by 3,356 documents - divided into 2,816 non-hate, 410 weak hate and 130 strong hate documents; and the two-class dataset, composed of 3,575 documents - divided into 2,789 non hate and 786 hate. The second dataset shows much higher result for hate recognition than the first dataset. We note that emeddings typically requires large corpora to develop meaningful results but in the context of hate speech detection, smaller datasets of the order of between 400 and 6000 hateful samples has yielded promising results.

Several works have merged typed dependencies with embedding learning and clarified the different levels of embedding learning when using the dependency context rather than the bag of words linear text. [39] showed that dependency context embeddings can provide valuable syntactic information for sentence classification tasks, which is a motivation for implementing classification tasks in respect of dependency embedding text, and [36] showed that dependency-based embeddings are less topical and exhibit more functional similarity than bag of words linear embeddings . In addition, [74] defined the differences between flat text, which they called neighbour words, and the dependency context, and clarified through examples the drawbacks of learning embedding from flat text. While these studies introduced the effectiveness of the word distances of the dependency context, which capture the semantic relations, their works targeted other areas of research, not cyberhate. One study that *has* combined typed dependencies with embedding learning in the context of cyberhate was reported by [47]. They developed a machine learning approach to cyberhate based on different syntactic features as well as different types of embedding features, and reported its effectiveness when combined with some standard NLP features (n-grams, dependencies) in detecting hate speech in online user comments. They showed how applying each feature set alone resulted in different classification performance, and found that character n-grams alone are useful in noisy datasets. While using n-grams boosted the learning performance, n-grams result in high dimensionality and thus render the models susceptible to overfitting. They examined different syntactic features as well as different types of embedding features and showed that this combination outperforms basic embedding learning. While they examined a combination of different syntactic features and embedding learning, some syntactic features led to confusion in the embedding learning process (e.g. some PoS and dependency modifiers). Therefore, this presents an open research question on how to better refine the framing of cyberhate from a computational feature processing perspective to improve the classification performance. Our study is different from the previous study in that we are yet to see evidence that the complex and nuanced ‘us and them’ narrative emerging on social media can be captured using a combination of typed dependencies and embeddings.

3 THEORY AND DATA

In this section we introduce our research hypothesis, the datasets that are used in our study, and a statistical analysis on the use of othering language in our datasets.

3.1 Research Hypothesis

Based on the analysis of existing literature on automated approaches to cyberhate and the theories of Integrated Threat Theory and ‘othering’ as a form of discriminatory language use, we propose that existing machine classification performance could be improved by including a layer of linguistic features representing ‘othering’ within short informal text, such as posts that are published to online social networks. We assume that othering language could be identified through specific uses of English language parts of speech - particularly verbs (action-driven language) and pronouns (referring to ‘others’). We hypothesize that the use of pronouns that refer to an ingroup (e.g. we, us) *co-occurring* with pronouns that refer to an outgroup (e.g. them, they) in the same post, will be indicative of divisive or antagonistic attitudes and therefore will improve machine classification of cyberhate. In this study, we refer to the co-occurrence of ingroup/outgroup pronouns as a *two-sided pronoun.*

Figure 1 presents an overview of linguistic features that can be used between different groups to distinguish themselves (the in group) from others (the out group).

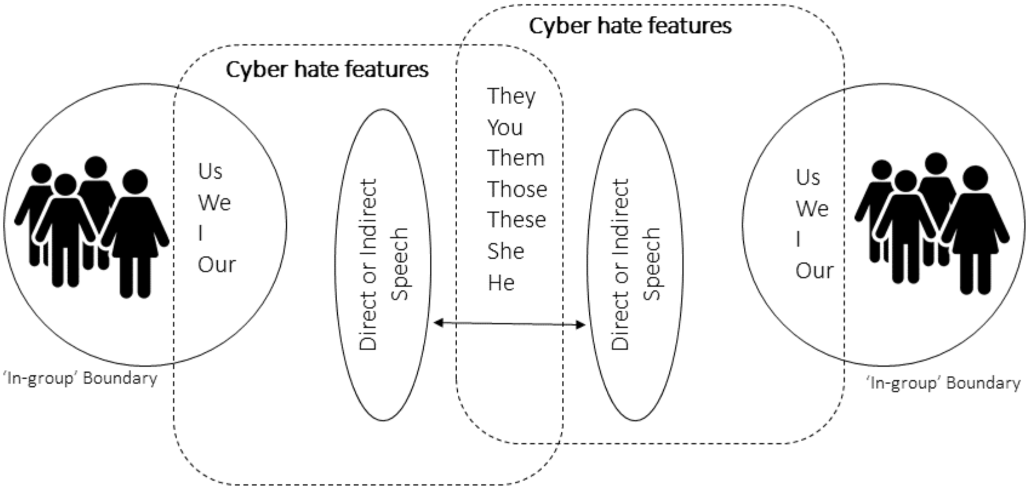


Fig. 1. The graph shows the boundary defined between two groups using the othering terms, and the space between the boundary shows how the negative text could be defined

The figure shows how pronoun terms from one side (us, we, our, etc.) draw the boundary between the in group by referring to the out group (we, they etc).

3.2 Datasets

We used two datasets for our experimentation. Our main contribution to the literature is to enhance existing cyberhate machine classification performance by developing a novel othering feature set to provide additional features that capture more nuanced forms of cyberhate based around a fear of 'the other' - for instance, around immigration and terrorism. To develop the othering feature set we used the dataset provided by [16]. They collected tweets containing different types of hate and used crowd-sourcing to further divide the sample into three categories: those containing hate speech, those with only offensive language, and those with neither. Annotators were asked to think not just about the words appearing in a given tweet but also about the context in which they were used. They were instructed that the presence of a particular word, however offensive, did not necessarily indicate a tweet is hate speech. Each tweet was coded by three or more annotators. The inter coder-agreement score was 92%. They used the majority decision for assigning a label to each tweet. This resulted in a sample of 24,802 labeled tweets and only 5% of the tweets were coded as hate speech by the majority of annotators. We refer to this as *Training Dataset* which contains 3161 non-malicious samples and 5323 hateful samples - which is at the higher end of samples size in previously published literature that used embeddings to classifying cyberhate (e.g [18] 540 hateful samples, [21] 1037 hateful samples, [31] 712 hateful samples, [72] 413-5773 hateful samples, [51] 5109 hateful samples). To compare our work to the state-of-the-art in cyberhate classification we used a second dataset for testing purposes, which was used in previous work [11]. The dataset was collected from Twitter and the method of data collection was random to ensure that the produced dataset is free from bias. The dataset was annotated using the CrowdFlower human intelligence task service with a single question: 'Is this text antagonistic or hateful based on a protected characteristic?'. The dataset comprises cyberhate directed at four different protected characteristics, as follows: sexual orientation - 1803 tweets, with 183 instances of offensive or antagonistic content (10.15% of the annotated sample); race 1876 tweets, with 73 instances of offensive or antagonistic content (3.73% of the annotated sample); disability 1914 tweets, with 51 instances of offensive or antagonistic content (2.66% of the annotated sample); and religion 1901 tweets, with 222 instances of offensive or antagonistic content (11.68% of the annotated sample). The authors conducted all of the necessary tests so as to ensure agreement between annotators for the gold-standard samples [11]. The amount of abusive or hateful instances is small relative to the size of the sample. However, these are random instances of the full datasets for each event and they are considered representative of the overall levels of cyberhate within the corpus of tweets [10]. We evaluate the relative improvement in classification performance using this dataset, which we refer to as *Testing Dataset.*

3.3 Summary Statistics for Othering Language in the Datasets

To provide an initial justification for our research hypothesis on the use of two sided pronouns to improve cyberhate classification, we conducted a corpus analysis of our datasets - calculating the percentage of tweets from Testing Dataset that included at least two pronouns. We found these features present in only 0.9% of non-malicious instances within the data and 17.6% of cyberhate instances.

In Figure 2 we show the comparative occurrence of two-sided othering terms in both hateful samples and non-hateful samples among different types of cyberhate datasets. We can see the anti-Muslim dataset contains the most frequent usage of the two-sided othering language, followed by anti-semitism. This follows from the findings of [9] who identified 'othering' language as a useful feature for classifying cyberhate based on religious beliefs. We can also note that the percentage of two sided othering was higher in the hateful annotated samples than non-hateful samples for all other type of hate speech, and to around the same level. In addition to that, in Figure 3 we show the same comparison based on the results obtained using Training Dataset which confirms again the much higher use of two-sided pronouns in the hateful samples.

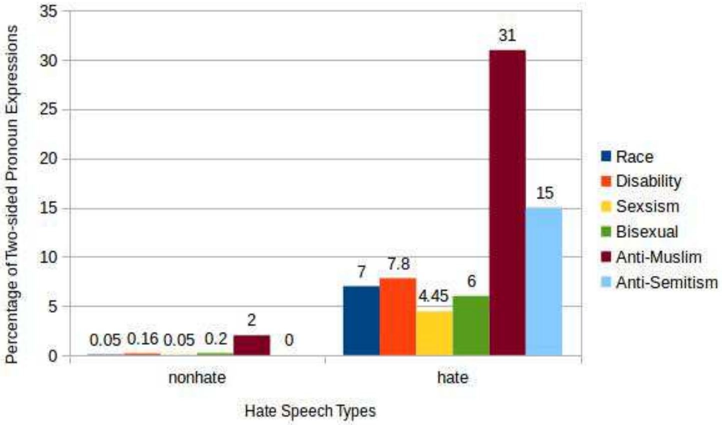


Fig. 2. The graph shows the use of two-sided othering for each hate speech type in both hateful and non-hateful samples in the testing dataset

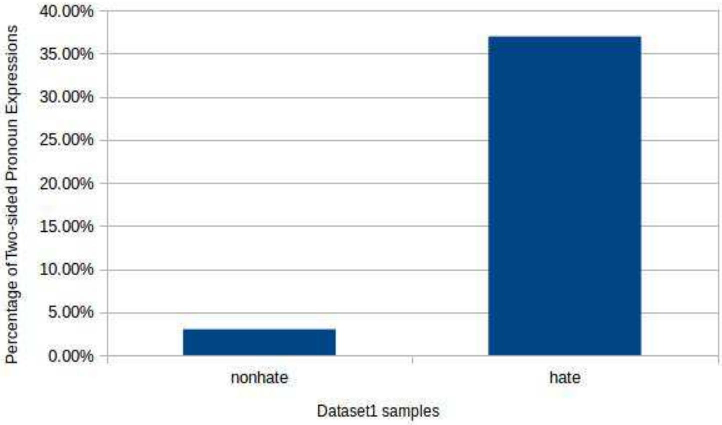


Fig. 3. The graph shows the use of two-sided othering for each hate speech type in both hateful and non-hateful samples in the training dataset

4 METHODS

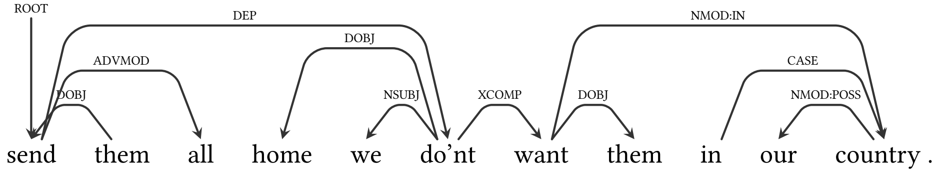
To achieve our novel othering layer within the machine classification framework we developed an othering feature set containing three components: (i) a constrained subset of dependency relationship labels extracted using probabilistic parse trees that we hypothesized would be representative of othering, (ii) more general parts of speech associated with othering including verbs (VB), nouns (NN) and adjectives (JJ) (e.g. send them home), and (iii) a list of English pronouns. Together these capture as much as possible linguistically to provide a focused set of othering features. This section details the process used to extract these features and how they were used in our automated machine classification approach.

4.1 Extracting Othering Terms

The first phase involved using the Training Dataset and analysing only the hateful samples. We identified all samples where two-sided othering was present - i.e. where at least two pronouns were used. We discarded any samples without at least two pronouns and then we discard the repeated samples. Using this subsample, we used the Stanford Typed Dependency Parser to transform the text to provide co-occurring words with a probabilistically derived linguistic label. The Stanford typed dependencies representation was designed to provide a simple description of the grammatical relationships in a sentence[17].

Figure 4 shows the linguistic labels associated with each word in a sample sentence. Word order within a sentence is preserved in the type dependency and provides a feature for classification as well as the syntactic relationship between words.

Fig. 4. Dependency Relationships



The Stanford Parser returns 51 different linguistic labels [2], and in the previous example, the parser produced seven dependency relationships, which are distributed over ten instances. To provide a specific focus on othering language, we retained only 6 types of dependency relationships: *nsubj*, *dobj*,*nmod*,*det*, *advmod* and *compound*. The remaining dependency modifiers were discarded. The rationale for preserving these modifiers is as follows. The *nsubj* label captures the syntactic subject or proto-agent in a sentence (i.e. the active agent). Examples include 'muslims caused' and 'they inflicted'. *dobj* concerns the direct object of a verb phrase and has a high probabilistic likelihood for capturing relationships between verbs and nouns, pronouns and determiners in the same phrase (e.g. send and them). *nmod* is likely to identify nominal modifiers for nouns, for instance 'all gays' or 'womens place'. The *det* captures the relationship between nominals and their determiner (e.g. 'these terrorists'). *advmod* captures adverb modifiers (e.g. where we see 'home' we may also see 'send'). The *compound* will identify compound verb phrases including verb and adjective compounds such as 'send back' or 'kill black'.

As a worked example of how this method is expected to capture othering, the translation of the text in Figure 4 becomes *nsubj(want-7, we-5), dobj(send-1, them-2) det(home-3, all-4), nmod:poss(country-11, our-10)* and the remaining relationships would be discarded. We are now capturing distinctive othering features that co-occur in the same sentence. Despite none of these words being clearly antagonistic or offensive of their own - together they provide a greater contextual feature for machine classification to detect these unseen samples using similar phrasing.

4.2 Building the Othering Feature Set

To complement the dependency relationship features we also applied part-of-speech (POS) tagging to the hateful samples that included at least two pronouns. We once again refined the set of labels to include those most likely to represent othering and retained only words tagged as nouns (NN), adjectives (JJ), verbs (VB) and adverbs (RB). The POS labels themselves were removed to leave only

[2] https://nlp.stanford.edu/software/dependencies_manual.pdf

words. These POS words, the dependency relationship features and a list of all English pronouns were then concatenated into a triple that formed the basis of an othering feature set - a novel concatenation of a range of grammatical and linguistic features extracted from a human annotated data set of hateful and antagonistic texts. To reduce noise, we removed all tweets that contained a single word. This process resulted in a dataset of 975 rows. As the othering feature set was built using annotated hateful samples, we expect that all the entries could contribute to the learning process. Figure 5 shows the process of extracting the ‘othering feature’ from each tweets that contain two sided pronouns. Algorithm 1 illustrates the steps of building the othering vectors feature set. Figure 6 shows the process of training our model and figure 7 illustrates the testing phase.

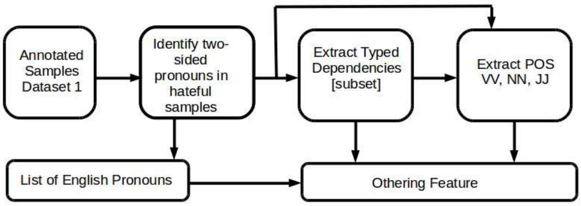

Annotated Samples Dataset 1
Identify two-sided pronouns in hateful samples
Extract Typed Dependencies [subset]
Extract POS VV, NN, JJ
List of English Pronouns
Othering Feature

Fig. 5. Othering Feature Extraction

We examined three types of inputs: (1) Training Dataset, which contains all the non-hateful and hateful instances that contain two sided pronouns in an unprocessed form - i.e. raw text; (2) Training Dataset with the non-hateful instances transformed into Typed Dependency representation, plus the othering feature set alone to represent hateful instances. We call this **proposed feature set 1**; and (3) we merge (1) and (2), which we call **proposed feature set 2**. Examples of each input are as follows: (1) *‘Send them all home we don’t want them in our country’*; (2) *‘Row0: [(them,we,our) + nsubj(want-7, we-5), dobj(send-1, them-2) det(home-3, all-4), nmod:poss(country-11, our-10) + send,want,home]’*. These features then become the input to the Paragraph2Vec algorithm.

Algorithm 1 Othering feature set

```
INPUT:Annotated Training Dataset
OUTPUT: Othering Feature set
for  each samples in Training Dataset do
    Identify tweets containing two sided pronoun
end for
for each sample containing two sided pronoun do
    Extract all the pronoun ($P$)
    Extract Typed Dependency ($TD$)
    Extract POS ($POS$)
    Append ($P$,$TD$,$POS$)
end for
```

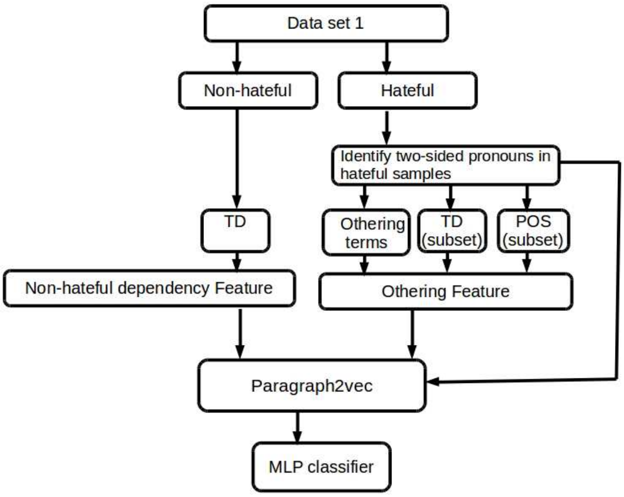


Fig. 6. Model Training Workflow

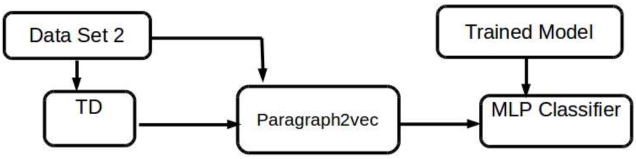


Fig. 7. Model Testing Workflow

4.3 Feature Extraction

At this stage, we needed to identify a suitable method for utilising the features extracted through our othering feature set. We could have used these as raw features for classification but to provide further refinement we employed embedding learning to capture the relative 'distance' between these features in a cyberhate context. Learning vector representations allows us to plot each feature in such a way that we can calculate numeric distances between features based on their use in a context. A common example of this is that the distances between 'man' and 'king' would be similar to that of 'woman' and 'queen'. Therefore we can identify relationships between words (i.e. 'man' and 'king'), and context (i.e. 'king' is to 'queen' as 'man' is to 'woman'). With two-sided pronouns we assume that our method would benefit from this approach to extract the semantic 'meaning' of othering features and learn these jointly across the hateful and non-hateful texts to provide context for term use - ultimately with the aim of these features being able to better support machine classification of both.

Various methods have been proposed to learn vector embedding representations. Word2Vec and Paragraph2vec have been proposed for building word/paragraph representations in low-dimensional vector space [44]. In the Word2Vec model, *words* are represented in continuous space where semantically similar words have a high similarity measure in that space. In the Paragraph2vec model, *paragraphs* are represented as low-dimensional vectors and are jointly learned with distributed vector representations of tokens using a distributed memory model (for further detail see [45]). Every sentence is mapped to a unique vector, and every word included in the sentence is also mapped to a unique vector. In our context this means each tweet is fed into the embedding learning methods as if it were a sentence, and each feature of the tweet derived in the othering feature extraction phase becomes a part of the sentence embedding.

Both data sets 1 and 2 were transformed into paragraph embeddings using Paragraph2Vec for training and testing purposes. Note we also implemented Word2Vec for sentence classification, which resulted in poor classification results so the decision was made to discard the use of Word2Vec and use only Paragraph2Vec.

We learned the Distributed Memory (PV-DM) vectors using the Gensim [3] implementation of distributed representations of the sentence (tweet) [37]. In the Distributed Memory component (PV-DM), the sentence acts as a memory that remembers the missed word in the current context of the sentence. According to [45], the distributed memory model is consistently better than PV-DBOW. To find the best implementation for our data, we experimented with both and found that distributed memory performed better in learning feature vectors from our data set. We used small window sizes because, according to [39], a window of size 5 is commonly used to capture broad topical contents, whereas smaller windows (e.g. *k=2* windows) contain more focused information regarding the target word.

For example, for k = 2, the context around the target word w comprises w- 2, w- 1, w + 1, w + 2. These become the features used for learning distances between the target word and its surrounding context. The larger the window, the broader the context. The final output is a vectorised data set that is used as a feature set for feeding in to a machine classification approach. We were expecting the othering layer to assist in improving the performance of machine classification of cyberhate, so a more focused approach seemed logical given it will be these nuanced othering terms in a smaller window that will likely lead to improvements in classification. We experimented with various window sizes including 100, 300, 600, 800 and 1000 dimensions and k = 2, 3, 5, 6 and 10. We recorded the performance of each and report the best performing configuration - which was for 600 dimensions and *windows = 2*. We used fixed embedding because our datasets are not associated with time, which would require grouping the data into time bins and training the embeddings separately on these bins [34]. Once we learned vector representations, we joined the vector with its original human-assigned label, assigning the label 0 to the non-hateful samples and 1 to the hateful samples, and then used these to train and test the machine classifier.

4.4 Machine Classification: Comparing the 'Othering' Classifier to Baselines

We examined several classification approaches, drawing on state of the art related cyberhate research to determine the overall improvement when using our novel othering feature set. The candidate methods included: **(Baseline 1)** - Support Vector Machines (SVM) and Random Forests

[3] https://radimrehurek.com/gensim/models/Doc2Vec.html

(RF) combined with Bag of Words (BoW), n-gram, and Typed Dependency features, as used in [10][11]. The SVM parameters were set to normalize data, use a gamma of 0.1 and C of 1.0 and we employed radial basis function (RBF) kernel and the Random Forest (RF) iteratively selects a random sub-sample of features in the training stage and trains multiple decision trees before predicting the outputs and averaging the results which maximize the reduction in classification error [8]. The Random Forest algorithm was trained with 100 trees; **(Baseline 2)** used a Logistic Regression (LR) classifier with Paragraph2Vec feature extraction for joint modeling of comments and words, as used in [19]; **(Baseline 3)** used Vowpal Wabbit's regression model and different NLP features with Paragraph2Vec and Word2Vec used for feature extraction, as applied by [47]; **(Baseline 4)** used Gradient Boosted Decision Trees (GBDTs) in combination with Long Short-Term Memory (LSTMs) for feature extraction (not as a classifier), and random embeddings, as published by [4]. They used the LSTM model to capture sequence-based features. The LSTM model has a single layer of LSTM units and all of the words in the corpora were initialised with random values. The output dimension size of the LSTM layer was 100. A sigmoid layer was built on the top of the LSTM layer to generate predictions. The input dropout rate and recurrent state dropout rate were both set to 0.2. In each iteration of the bootstrapping process, the training of the features and classifier runs for 15 epochs; **(Baseline 5)** included a CNN model in combination with Word2Vec embedding. We performed training in batches of size 128 for CNN as introduced in [21]. We use the 'adam' optimizer for CNN. We configured the model with three max-pooling layers, as introduced in [21]. A max-pooling layer captures the most important latent semantic factors from the Tweets. The output layer used softmax to calculate the class probability distributions for each Tweet and assigns each Tweet the class that obtains the maximum probability value; **(Baseline 6)** was a modified CNN classifier with Gated Recurrent Units (GRU) layer which applied on learned Word2Vec embeddings as introduced by [71]. They integrated a GRU layer with a CNN classifier to capture long range dependencies in Tweets, which may play a role in hate speech detection. GRU layer takes input from the max pooling layer. This treats the features as time steps and outputs 100 hidden units per time step. Compared to LSTM, the key difference in a GRU is that it has two gates (reset and update gates) whereas an LSTM has three gates (namely input, output and forget gates). Thus GRU is a simpler structure with fewer parameters to train. In theory, this makes it faster to train and generalise better on small data; while empirically it is shown to achieve comparable results to LSTM [13]; (**Baseline 7**) used an LSTM classifier with random embedding which was introduced in [4] but did not produce an improvement on the use of GBDTs in their results. However, in our study we aimed to reveal the effectiveness of using an LSTM model as a standalone classifier on our proposed feature set, and the parameters were set the same as [4]. For CNN, LSTM and CNN+GRU models, the feature extraction phase all resulted in paragraph level vectors being extracted for each sentence (Tweet). This means that all of the baselines (except baseline 1) have paragraph vectors for classifier inputs, which makes them comparable to the Paragraph2Vec as used in our othering feature set.

In addition to the state of the art classification methods, we propose the use of our othering feature set (pre-processed with Paragraph2Vec) along with a Multilayer Perceptron (MLP) classifier [70]. Multilayer feed-forward networks can provide competitive results on sentiment classification and factoid question answering [30]. MLP is a feed-forward artificial neural network model which maps input data sets on an appropriate set of outputs. MLP consists of multiple layers of nodes in a directed graph, with each layer being fully connected to the next layer [23]. To the best of our knowledge no work has previously used MLP classifiers in combination with Paragraph2Vec for cyber hate detection. MLP parameters were set experimentally by setting the initial number of the hidden layers to 1, and increasing the number of the hidden layers to improve performance through trial and error [27]. In our case, two hidden layers with five hidden connected units achieved the best performance for our vectors with 200 iterations.

We implemented all these approaches on the raw data set, on our proposed othering feature set, and on the combination of both data sets for comparisons. Then, to determine the effectiveness of each individual model in classifying cyber hate, we cross-validated across all input feature sets on an individual basis. For each cross-validation fold, the paragraph embeddings were re-trained. Throughout the results section we refer to the correct classification of non-hateful samples as true negatives, and correct classification of hateful samples as true positives.

5 RESULTS AND DISCUSSION

5.1 Quantitative Results

The first set of experiments (Section 5.1.1) included applying a wide range of models from previous studies, which we summarised as baselines in the previous section. This allows us to compare the best performing models for cyber hate classification with our proposed 'othering' feature set. The second set of experiments (Section 5.1.2) involved testing the best performing models from the first phase on completely unseen data to test model generality over four types of hate speech.

5.1.1 ***Testing against the state of the art*** . The results are shown in Table 1, in which the first column represents a summary of the baseline models ([11], [19], [47], [4], [21] and [71]). The last row in the table contains the result of our novel approach. We trained our othering feature set as well as the baseline methods using ten-fold cross validation. This method has previously been used for experimentally testing machine classifiers for short text [58] [11]. It functions by iteratively training the classifier on feature vectors from 90 percent of the annotated data set, and classifying the remaining 10 percent as 'unseen' data, based on the features evident in the cases it has encountered in the training data. It then determines the accuracy of the classification process and moves on to the next iteration, finally calculating the overall accuracy. The results presented in Tables 1 and 2 are for the cyberhate class only. The classification performance for non-hateful text was consistently above 0.90-0.99 and are omitted to reduce complexity in presenting the results. Our main interest is with the improvement of cyberhate classification. In training the classifiers matrix, we use F-measure as our main comparison metric, given it controls for false positives and false negatives and a lack of balance in the dataset. At classifier testing phase, we used F-measure and AUC for error analysis.

Our experiments were conducted on three input datasets: (i) an unprocessed dataset (Training Dataset) which reflected the (actual) implementation of the applied models; (ii) the othering feature dataset; and (iii) the unprocessed Training Dataset enriched with the othering feature set. This follows the method proposed in Sections 4.1 to 4.3. From Table 1 we firstly notice that from baseline 1 to baseline 2 there was a large reduction of FPs and FNs among the three feature sets. We suggest this is likely due to the use of semantic learning for features extraction. This also confirms that there is no bias in our data sets as noticed by [5]. From baseline 2 to baseline 3, the reduction of FPs (but not necessarily FNs) is clear. We suggest this is likely a result of using linguistic features (n-gram, dependency relations, etc.). The combination of baseline 3 with our third feature set produced the lowest number of FPs, detecting 99% of the cyberhate samples. This suggests that the extra features (n-gram and linguistics), which were applied by the baseline 3 model, improved the process of hate speech classification, compared with the baseline models, which already use the semantic features. However, applying neural network models proposed in baseline 4 [21], baseline 5 [4], baseline 6 [73] and baseline 7 [4] did not show a significant increase in the detection of cyberhate within the three datasets. The CNN [21] and LSTM models [4] were unable to improve on this using the original feature set nor our feature set. This is likely due to the length and sparsity of short texts. Furthermore, RNNs models including LSTM features extraction, which showed the best performance in [4] and the GRU layer, which was added to the CNN model to achieve best performance by [73]) showed weak performance compared to other classifiers. Another possible explanation for this is their use of Word2Vec feature extraction which, empirically, has not produced better semantic learning than Paragraph2Vec for our datasets (see Section 4.3).

The last row in the table shows the results of training our proposed feature sets using Paragraph2Vec for feature extraction and the MLP classifier. Despite two feature sets producing higher FP than baseline 3 on our feature set by 1 and 3 extra FP samples, they show a reduction of FN by 5 and 4 non-hateful samples. The result shows that the othering features set is working on a par with baseline 3 that uses a range of text pre-processing methods. To evaluate the generality of these two approaches (ours and baseline 3) - and stress the range of linguistic features used in baseline 3, as well as the othering method, we tested them using unseen datasets. The three best performing classifiers (shown in bold font in table 1) were the candidate models for training our 'othering' classifiers, because they resulted in the lowest number of both FPs and FNs (ranging between 0 and 10). We named the three best models as 'Comprehensive-classifier', 'Othering-classifier', and 'Othering+raw-classifier', respectively. Comprehensive-classifier refers to applying baseline 3 model on our proposed feature set 2; we name it as a *'comprehensive'* because there is a wide range of features were applied. Othering-classifier which refers to our proposed application of Paragraph2Vec feature extraction and MLP classifier on the proposed feature 1 (othering feature set) and Othering+raw-classifier which refers to our proposed application of Paragraph2Vec feature extraction and MLP classifier on the proposed feature 2 (othering feature set and raw data set).

Table 1. Machine classification performance for the cyberhate classifiers based on training dataset

			Unprocessed Training Dataset				**Proposed Feature set 1 (Othering Feature set)**				**Proposed Feature set 2 (Dataset1 + othering feature set)**			
Classifiers	**cl**	**sample**	**P**	**R**	**F**	**AUC**	**P**	**R**	**F**	**AUC**	**P**	**R**	**F**	**AUC**
Baseline model 1: n-Gram words (1-5) with 2,000 features +n-Gram typed dependencies +hateful term [11]	SVM+ RF	Hateful	0.58 FP=407	0.48 FN=610	0.52	0.67	0.68 FP=311	0.30 FN=1560	0.41	0.56	0.78 FP=209	0.28 FN=1891	0.42	0.57
		Neutral	0.80	0.86	0.83		0.50	0.83	0.63		0.40	0.85	0.54	
Baseline model 2: Comment Embedding [19]	LR	Hateful	0.88 FP=118	0.93 FN=61	0.91	0.94	0.74 FP=253	0.87 FN=110	0.78	0.89	0.98 FP=10	0.96 FN=40	0.97	0.97
		Neutral	0.98	0.96	0.97		0.95	0.92	0.94		0.98	0.99	0.99	
Baseline model 3: N-grams+ linguistic+ dependencies+ word and comment Embedding [47]	VR	Hateful	0.94 FP=54	0.93 FN=66	0.92	0.95	0.97 FP=24	0.98 FN=9	0.98	0.99	**0.98 FN=4**	**0.99 FN=9**	**0.99**	**0.99**
		Neutral	0.97	0.98	0.98		0.99	0.99	0.99		1.00	1.00	1.00	
Baseline Model 4: Word2vec [?])	CNN	Hateful	0.97 FP=27	0.88 FN=131	0.92	0.93	0.98 FP=15	0.83 FN=196	0.90	0.91	0.98 FP=11	0.90 FN=101	0.94	0.95
		Neutral	0.95	0.99	0.97		0.93	0.99	0.90		0.96	0.99	0.98	
Baseline Model 5: Word2vec +LSTM [4]	GBDT	Hateful	0.73 FP=258	0.77 FN=215	0.75	0.84	0.98 FN=17	0.89 FN=102	0.94	0.95	0.96 FP=34	0.95 FN=47	0.97	0.97
		Neutral	0.93	0.91	0.92		0.96	0.99	0.98		0.98	0.99	0.98	
Baseline Model 6: Word2vec [73]	CNN+ GRU	Hateful	0.92 FP=75	0.87 FN=133	0.90	0.92	0.97 FP=25	0.92 FN=30	0.95	0.98	0.99 FP=10	0.89 FP=115	0.94	0.94
		Neutral	0.95	0.97	0.96		0.99	0.99	0.99		0.96	0.99	0.97	
Baseline 7:Word2vec [13]	LSTM	Hateful	0.00 FP=975	0.00 FN=0	0.00	0.00	0.59 FP=365	0.36 FN=1003	0.45	0.60	0.83 FP=158	0.45 FN=974	0.59	0.69
		Neutral	1.00	0.76	0.86		0.68	0.84	0.75		0.69	0.93	0.79	
Paragraph2vec	MLP	Hateful	0.98 FP=21	0.95 FN=51	0.96	0.97	**0.99 FP=5**	**0.99 FN=3**	**1.00**	**0.99**	**0.99 FP=7**	**0.99 FN=5**	**0.99**	**0.99**
		Neutral	0.98	0.99	0.98		1.00	1.00	1.00		1.00	1.00	1.00	

5.1.2 ***Testing the Othering Classifier.*** Our second set of experiments involved testing our model on unseen datasets. The training phase (the previous experiment) produced evidence to suggest that including othering language features in predictive models of cyberhate speech (in short informal text, such as Twitter posts) will improve the classification performance. The second phase, which is testing the trained classifier, aimed to determine the possibility of developing a more generalised model of cyberhate detection. A key finding from previous research is that, compared with using hateful terms alone, the inclusion of features capable of detecting othering language in the classification of religious cyberhate reduced false negatives by 7%. Additionally, [47] found that computing feature embeddings when combined with the standard NLP features showed the effectiveness for improving the performance of cyberhate classification.

Table 2. Machine classification performance for cyberhate classifiers based on unseen testing datasets

	Tested Data Sets															
	Religion				**Disability**				**Race**				**Sexual-orientation**			
Trained Models	**P**	**R**	**F**	**AUC**	**P**	**R**	**F**	**AUC**	**P**	**R**	**F**	**AUC**	**P**	**R**	**F**	**AUC**
Baseline Model 3 + Proposed Feature set 2 (Comprehensive-classifier)	**0.95** **FP=12**	**0.71** **FN=84**	**0.81**	**0.94**	**0.68** **FP=16**	**0.74** **FN=12**	**0.71**	**0.86**	0.89 FP=8	0.30 FN=163	0.43	0.64	**0.98** **FP=3**	0.31 FN=398	0.47	0.65
Paragraph2vec+ Proposed Feature set 1 (Othering-classifier)	0.95 FP=12	0.44 FN=260	0.60	0.71	0.31 FP=35	0.94 FN=1	0.47	0.96	0.95 FP=4	0.79 FN=18	0.86	0.89	0.78 FP=40	0.46 FN=168	0.58	0.71
Paragraph2vec+ Proposed Feature set 2 (Othering+raw-classifier)	0.99 FP=2	0.54 FN=187	0.69	0.76	0.55 FP=23	1.00 FN=0	0.71	0.99	**0.95** **FP=3**	**0.84** **FN=13**	**0.89**	**0.92**	**0.86** **FP=24**	**0.61** **FN=99**	**0.72**	**0.80**

Our results show how the use of othering features alongside embeddings for training the classifier enables a new level of hate speech detection, in the form of othering-level feature embeddings. We tested the best three classifiers from previous experiments on four unseen cyberhate datasets. As shown in Table 2, for religion, the third classifier was able to detect 99% of the religious hate but only 88% of non-hateful samples. However, the lowest number of incorrect non-hateful samples was achieved by the Comprehensive-classifier which detected 94%. The Comprehensive-classifier detected 95% of the hateful samples, and is considered to be a more balanced classifier. The very low number of missed true positives in the Othering+raw-classifier indicates the effectiveness of the 'othering feature set' in detecting hate speech, whilst using n-grams and linguistic features for training the Comprehensive-classifier had a positive effect on non-hateful detection. For the disability dataset, the Comprehensive-classifier detected 68% of the hateful samples, which was the lowest among the three classifiers, whereas the Othering+raw-classifier detected all non-hateful samples but approximately half of the hateful samples were missed. This could be interpreted as othering language not being commonly used in disability hate speech. Additionally, because the Comprehensive-classifier used linguistic features (n-grams) in addition to our feature set, the disability context was enriched. For the race dataset, the Othering+raw-classifier improved the detection of hateful and non-hateful instances compared to the Comprehensive-classifier and the Othering-classifier by 6% and 1%, respectively. For the sexual orientation dataset, the Othering+raw-classifier detected the lowest number of hateful instances (3 were missed) but had the highest false detection for non-hateful instances (missing 33%).

In summary, the Othering+raw-classifier detected the highest number of hateful and non-hateful instances for racism hate speech, whereas the Comprehensive-classifier performed best at detecting hateful samples in the religion, disability, and sexual orientation datasets. These results show that there is strong competition between the Comprehensive-classifier and the Othering+raw classifier (with the Comprehensive-classifier using additional n-gram and linguistic features) in detecting hateful and non-hateful speech for religion and sexual orientation. The use of n-grams and linguistic features did not, therefore, positively contribute to the detection of religious hate speech, whereas the opposite was true for sexual orientation. The best performance was achieved by the Othering+raw-classifier for racial hate speech and non-hateful speech, which suggests that the use of 'othering' features is more important than additional n-grams or linguistics features. However,the Othering-classifier did not record prominent results despite interesting results at the training stage.

We have validated the utility of the learned vectors in the classification of cyber hate by reporting the area under the curve (AUC). AUC provides an aggregate measure of performance across all possible classification thresholds, ranging from 0 to 1; a model with predictions that are 100% incorrect has an AUC of 0.0, and one with predictions that are 100% correct has an AUC of 1.0. We found that other approaches for classifying religious and disability datasets were outperformed by training the classifier using baseline model 3 and our proposed 'othering' feature set 1, having an AUC of 0.94 and 0.71, respectively. For race and sexual orientation datasets, combining our proposed 'othering' feature set 2 with the MLP classifier achieved a higher AUC than combining it with baseline 3, having an AUC of 0.92 and 0.80 respectively. High AUC and high F-scores indicate that the classifier performs well at all thresholds [40], and this was seen in the case of religion, race and sexual orientation classifiers (see bold font in table 2). In comparison, low AUC indicates a weak classifier possibly due to an imbalanced dataset [29]. Obtaining F-measures over 0.80 by testing a model on unseen data sets which were collected in different circumstances shows the effectiveness. Additionally, table 2 contains results that were not compared with previous studies baselines for the following reasons: the data set used for training and testing the classifiers are entirely different so we have not followed the splitting method which split the same data set for training and testing the model. In addition, we have not followed the 10-cross validation to validate the generality (testing) of our trained classifier. We avoid using the two previous methods for testing our classifiers, which were commonly used for validating the trained model in the previous studies, because of the probability of text repetition which might happen during the data collection (e.g. retweets).

5.2 Qualitative Results

Given our improvements over the state of the art using the othering feature, we conducted our own qualitative analysis to identify any insights into the features captured using feature embedding on the othering features - i.e. the two-sided pronouns. Given the improvement, we can assume that the embedding method has effectively assigned othering features to similar vectors spaces in such a way as to better distinguish hateful from non-hateful content using the Paragraph2Vec embedding algorithm. We have visualized two data sets - unprocessed Training Dataset using embeddings only (see Fig 9), and second using the othering feature set with the embedding model (see Fig 8) - which reflects the representation of the 'us and them' narrative that produced the third experiment in last row of Table 1). We visualise our model using TensorBoard which has a built-in visualizer (we perform 2D principal component analysis (PCA)), called the Embedding Projector, for interactive visualization and analysis of high-dimensional data like embeddings [4]. The distances between words are relative based on their computed similarity to other words in the hateful sample. The two graphs are focused and enlarged to show the 300 most similar words. The colors indicate the distances from the key word 'us', the purple dots indicate the smallest distances (0.008-0.09), next smallest are the pink dots (0.093-0.2), then orange dots (0.21-0.39), then dark yellow dots (0.4-0.55), and finally the light yellow dots are furthest from 'us' (0.56-1). In distance functions, smaller values imply greater similarity between words [1]. Ideally we want the classifier to be able to use these small distances to make effective use of them as features for distinguishing hate from non-hate.

We can see from Figure 8 that the words with the smallest relative distance from 'us' (the purple, pink dots) include pronouns from the ingroup (*us, we*), pronouns from the outgroup (*these, them, they etc.*), and different reaction verbs (*send, kill, shoot, hang etc.*), which captures their co-occurrence in more nuanced othering aspects of cyberhate language. From an Integrated Threat Theory perspective we can also see symbolic and realistic anxiety present (e.g. the words 'attack', 'state', 'jihadist', 'animals') and intergroup anxiety (e.g. 'Arab', 'Israel'). Furthermore we can see symbolic threats which are focused on cultural differences (e.g. Muslims, Jews, Arab, Pakistanian, Africano, American). We can also see the obvious derogatory terms in the same Figure (e.g. *suck, fuck, discuggg, niger, niggero, savages*). Thus, this model is picking up both the obvious and non-obvious cyberhate using our othering features. Whereas in Figure 9 none of the words are particularly

[4] https://www.tensorflow.org/versions/r0.12/how$_{t}os/embedding_{v}iz/$

close to 'us' in terms of distance, meaning the classifier is unable to make effective use of the othering narrative. Thus, the classifier based on these features will become more dependent on the hateful words and miss the less obvious narrative. We posture that this ability to capture the more nuanced text is the core reason behind our successful improvement over the state of the art from previous research. In Table 3, we have summarized the visualization by showing the top 10 similar words through the two models: embedding based on the raw training data set only and the othering feature embedding. The similarity was measured by using the cosine similarity function. The similarity table shows the most similar words to the word *'us'* using our model which reflects the definition of different groups(e.g. *'Muslims, Jews, jihadist'*) while the word *'us'* in the raw embedding refer to concepts (e.g.*'fans, safe, devil, wife'*). The othering feature set succeeds in defining different groups and different attitudes which are important aspects in the field of hate speech recognition.

Table 3. Target words and their 10 most similar words as induced by raw training data set embedding and our proposed 'othering feature embedding'

Target word	**Raw training Data Set Embedding**	**Othering Feature Set Embedding**
us	fans, safe, devil, wife, gisuz, whips main, they, tinge, armi	ni**as, we, Arab, those, iran group, Jews, Muslim, send, headiest,animals
them	sort, stop, close, ask, differ, speak busi, tweeting, ni**ers, we	these, Jew, those, pakistanian someon hang, Muslims, rednecks, country, p**s, niggu
arab	shit, hot, like, bleiv, thing hat, went, thing, die, lol	iraq, we, ni**a, fu**er, israel, getout nig******as, outta, chop, animals
Muslim	smh, office, appear, backlash, nonmuslim dye, agenda, rasicm, christian, asian	us, nonmuslim, iraq, lable, Jews, racism high, arab, yellow, doe, country
send	only, what, neireian, fu**ing, I bet, saying, realy, reason, pick	Israel, suck, Africano, ni**er, getout, home, these paki, ni**o, kill, burn
out	have, try, see, never, some, home bastards, fail, wrong, give	gotta, outta, Islam, America, chop, attack, mosque, send home, manchest, blacks, these
scum	hear, speak, dumb, edltrobinson, exact, sort, breedingwoolwich, seem, kind, threaten	savages,islamic, yell, muslims, beat, pi**, tuesday wogs, against, burn, qouet, leage
shoot	divis, behind, pub, bloodi, scence hospital, condemen, plane	fu**ing, shot, nobody, ni**o, disable nutter, muslims, burn, nonmuslim, condemen
kill	stabb, death, pari, involve, year between, brutal, cheldren, charge, three	stabb, live, them, innocent, pakistanian soldir, who, place, krazi, suspect
burn	local, critic, swedish, bbcnews, islamabad faggots, saudi, antiterror, bangladish, letter	church, non-whites, chop, mosques, themfu**ing,shoot chop, ni**as, commit, nigeria

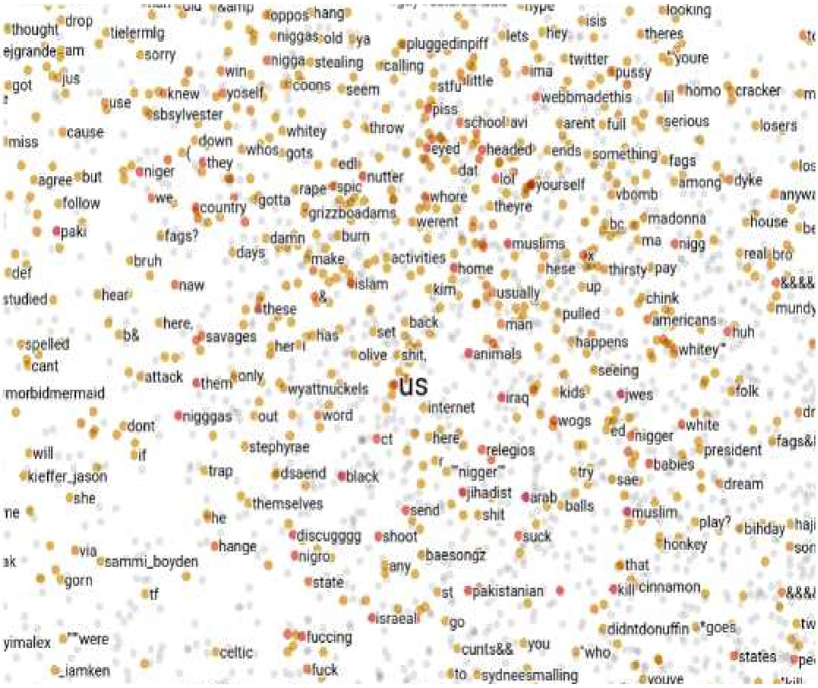


Fig. 8. Embedding and our Othering feature set vectors

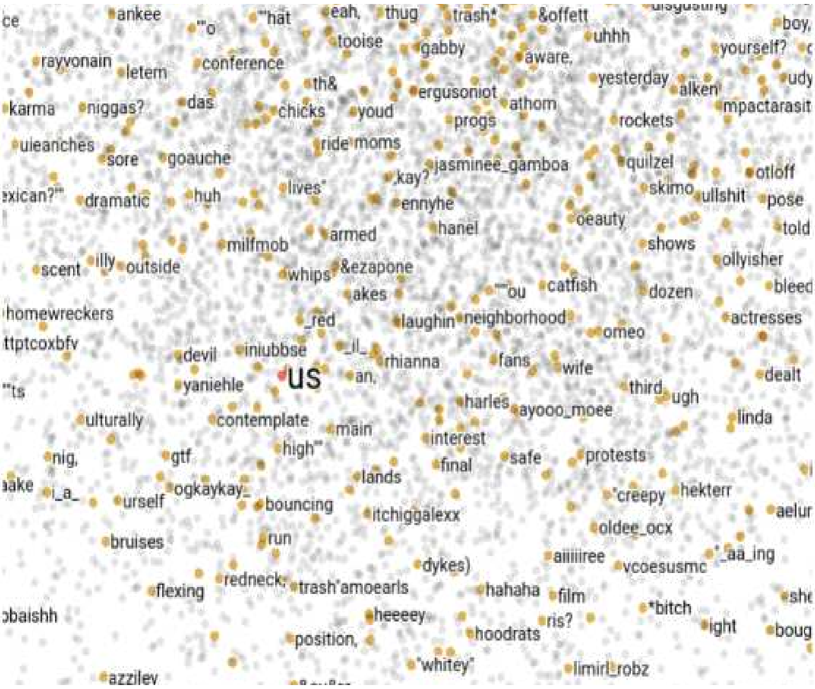


Fig. 9. Embedding Visualisation on original training data set

6 CONCLUSION

In this paper we aimed to improve the machine classification performance for different types of hateful and antagonistic language posted to Twitter - known as cyberhate. Our study was inspired by the concepts presented by Integrated Threat Theory (ITT) and ‘othering’ theory. We investigated the effectiveness of developing an abstract layer of linguistic features based around the use of ‘othering’ language, such as terms and phrases that separate the ingroup (e.g ‘we’, ‘us’) from the outgroup (e.g. ‘them’, ‘these’), and suggested action (e.g. send, kill, shoot, hang, etc.) or separation based on perceived symbolic and realistic threats (e.g. ‘attack’, ‘state’, ‘jihadist’). We also expected to detect symbolic threats which focused on the types of cultural differences (e.g. Muslims, Jews, Arab, Pakistanian, Africano). Our hypothesis was that the use of an ‘othering feature set’ would provide better context for the classifier beyond using words alone. We used vector embedding and the Paragraph2Vec algorithm to cluster these features, thereby re-framing the linguistic features from individual terms and phrases to numeric distances representing a form of semantic similarity of these terms - learned in the context of hateful or non-hateful texts. We then experimented with machine classification methods to determine the improvement of our novel ‘othering feature’ over the state-of-the-art research, and the most effective machine classifier to use with our embedding-transformed feature set.

Generally, the results show the effectiveness of including our proposed feature set for classifier training over the baselines model, and produced 0.99 F-measure for three models when trained using 10 fold cross validation. When tested on unseen data using four different types of cyberhate, namely religion, disability, race and sexual orientation, we obtained F-measures of 0.81, 0.60, 0.89 and 0.89 respectively. These results outperform the state of the art in the cyberhate literature and show the ability of the ‘othering’ narrative to be an important part of hate speech detection. However, our models perform well on some but not all categories of the unseen data (types of hate speech), which indicates that different types of hate speech have different language characteristics and the use of othering terms can be effective for some but not all contexts of hate speech.

We performed a qualitative inspection of the embedding representation of our ‘othering feature set’ and were clearly able to see the vector space similarity between ingroup/outgroup terms (us, we, they, them), action terms (get, send, shoot), terms related to anxiety of the threat from ‘the other’ (jihadist, fight, hate), and terms related to intergroup anxity (e.g. Arab, Jews, Africano) when pre-processed using our othering feature. Such features were not present using feature extraction methods in the existing literature, which applies embeddings without the context of othering. These additional contextual features provide the key novelty in our approach - and the Paragraph2Vec embedding method has made good use of these additional features to improve on the state of the art. Our approach allows the machine classifier to use a more broad range of features beyond individual words and ngrams, providing greater context for the classifier. In future we aim to develop larger datasets on which to test these classifiers, and use them to study rising and falling cyberhate levels on a range of online social media platforms, with the intention of collecting these narratives to better understand the topics and touch-points that are being discussed in this context at an aggregate level during times of civil unrest or following trigger events.

ACKNOWLEDGMENTS

This work was supported by the Economic and Social Research Council Research Grant: ’Centre for Cyberhate Research Policy: Real-Time Scalable Methods Infrastructure for Modelling the Spread of Cyberhate on Social Media’ [ES/P010695/1]. We would also like to acknowledge the scholarship from Shaqra university that supports the collaboration between the co-authors.

REFERENCES

[1] Charu C Aggarwal. 2015. Similarity and Distances. In *Data Mining*. Springer, 63–91.

[2] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. 2016. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence* 38, 7 (2016), 1425–1438.

[3] Muhammad Zubair Asghar, Aurangzeb Khan, Shakeel Ahmad, Maria Qasim, and Imran Ali Khan. 2017. Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. *PloS one* 12, 2 (2017), e0171649.

[4] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 759–760.

[5] Mario Guajardo-CÃĺspedes Margaret Mitchell Ben Packer, Yoni Halpern. 2018. Text Embedding Models Contain Bias. Here's Why That Matters. (2018). https://developers.googleblog.com/2018/04/text-embedding-models-contain-bias.html

[6] Darina Benikova, Michael Wojatzki, and Torsten Zesch. 2017. What Does This Imply? Examining the Impact of Implicitness on the Perception of Hate Speech. In *International Conference of the German Society for Computational Linguistics and Language Technology*. Springer, 171–179.

[7] Adam Bermingham and Alan F Smeaton. 2010. Classifying sentiment in microblogs: is brevity an advantage?. In *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 1833–1836.

[8] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[9] Peter Burnap and Matthew Leighton Williams. 2014. Hate speech, machine classification and statistical modelling of information flows on Twitter: Interpretation and communication for policy decision making. (2014).

[10] Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet* 7, 2 (2015), 223–242.

[11] Pete Burnap and Matthew L Williams. 2016. Us and them: identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data Science* 5, 1 (2016), 11.

[12] Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. IEEE, 71–80.

[13] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[14] Isobelle Clarke and Jack Grieve. 2017. Dimensions of Abusive Language on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*. 1–10.

[15] Stephen M. Croucher. 2013. Integrated Threat Theory and Acceptance of Immigrant Assimilation: An Analysis of Muslim Immigration in Western Europe. *Communication Monographs* 80, 1 (2013), 46–62. DOI: http://dx.doi.org/10.1080/03637751.2012.739704

[16] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. *arXiv preprint arXiv:1703.04009* (2017).

[17] Marie-Catherine De Marneffe and Christopher D Manning. 2008. *Stanford typed dependencies manual*. Technical Report. Technical report, Stanford University.

[18] Fabio Del Vigna12, Andrea Cimino23, Felice DellâĂŹOrletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on Facebook. (2017).

[19] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 29–30.

[20] Iginio Gagliardone, Danit Gal, Thiago Alves, and Gabriela Martinez. 2015. *Countering online hate speech*. UNESCO Publishing.

[21] Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-Speech. In *Proceedings of the First Workshop on Abusive Language Online*. 85–90.

[22] Lei Gao, Alexis Kuppersmith, and Ruihong Huang. 2017. Recognizing Explicit and Implicit Hate Speech Using a Weakly Supervised Two-path Bootstrapping Approach. *arXiv preprint arXiv:1710.07394* (2017).

[23] Sahar Ghannay, Yannick Esteve, and Nathalie Camelin. 2015. Word embeddings combination and neural networks for robustness in asr error detection. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 1671–1675.

[24] Manoochehr Ghiassi, James Skinner, and David Zimbra. 2013. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications* 40, 16 (2013), 6266–6282.

[25] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering* 10, 4 (2015), 215–230.

[26] Edel Greevy and Alan F Smeaton. 2004. Classifying racist texts using a support vector machine. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 468–469.

[27] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[28] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.

[29] Jin Huang and Charles X Ling. 2005. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering* 17, 3 (2005), 299–310.

[30] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 1681–1691.

[31] Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the Second Workshop on NLP and Computational Social Science*. 7–16.

[32] Christopher S Josey. 2010. Hate speech and identity: An analysis of neo racism and the indexing of identity. *Discourse & Society* 21, 1 (2010), 27–39.

[33] Eunice Kim, Yongjun Sung, and Hamsu Kang. 2014b. Brand followersâĂŹ retweeting behavior on Twitter: How brand relationships influence brand electronic word-of-mouth. *Computers in Human Behavior* 37 (2014), 18–25.

[34] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014a. Temporal analysis of language through neural language models. *arXiv preprint arXiv:1405.3515* (2014).

[35] Sebastian Köffer, Dennis M Riehle, Steffen Höhenberger, and Jörg Becker. 2018. Discussing the value of automatic hate speech detection in online debates. *Multikonferenz Wirtschaftsinformatik (MKWI 2018): Data Driven X-Turning Data in Value, Leuphana, Germany* (2018).

[36] Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of NAACL-HLT*. 1490–1500.

[37] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML*, Vol. 14. 1188–1196.

[38] Laura Leets. 2001. Responses to Internet hate sites: is speech too free in cyberspace? *Communication Law & Policy* 6, 2 (2001), 287–317.

[39] Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings.. In *ACL (2)*. 302–308.

[40] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. 2009. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39, 2 (2009), 539–550.

[41] Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. *arXiv preprint arXiv:1712.06427* (2017).

[42] Priscilla Marie Meddaugh and Jack Kay. 2009. Hate speech or 'reasonable racism?' the other in stormfront. *Journal of Mass Media Ethics* 24, 4 (2009), 251–268.

[43] Yashar Mehdad and Joel Tetreault. 2016. Do Characters Abuse More Than Words?. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. 299–303.

[44] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[45] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[46] Kevin Munger. 2017. Tweetment effects on the tweeted: Experimentally reducing racist harassment. *Political Behavior* 39, 3 (2017), 629–649.

[47] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 145–153.

[48] Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining.. In *LREc*, Vol. 10.

[49] Aasish Pappu and Amanda Stent. 2015. Location-Based Recommendations Using Nearest Neighbors in a Locality Sensitive Hashing (LSH) Index. (Nov. 20 2015). US Patent App. 14/948,213.

[50] Barbara Perry and Patrik Olsson. 2009. Cyberhate: the globalization of hate. *Information & Communications Technology Law* 18, 2 (2009), 185–199.

[51] Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence* (2018), 1–13.

[52] Haji Mohammad Saleem, Kelly P Dillon, Susan Benesch, and Derek Ruths. 2017. A web of hate: Tackling hateful speech in online social spaces. *arXiv preprint arXiv:1709.10159* (2017).

[53] Scharolta Katharina Sienčnik. 2015. Adapting word2vec to named entity recognition. In *Proceedings of the 20th Nordic*

Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania. Linköping University Electronic Press, 239–243.

[54] Leandro Araújo Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the Targets of Hate in Online Social Media.. In *ICWSM*. 687–690.

[55] Walter G Stephan and Cookie White Stephan. 2009. Intergroup threat theory. *The International Encyclopedia of Intercultural Communication* (2009).

[56] Walter G Stephan, Cookie White Stephan, and William B Gudykunst. 1999. Anxiety in intergroup relations: A comparison of anxiety/uncertainty management theory and integrated threat theory. *International Journal of Intercultural Relations* 23, 4 (1999), 613–628.

[57] Luke Kien-Weng Tan, Jin-Cheon Na, Yin-Leng Theng, and Kuiyu Chang. 2012. Phrase-level sentiment polarity classification using rule-based typed dependencies and additional complex phrases consideration. *Journal of Computer Science and Technology* 27, 3 (2012), 650–666.

[58] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 61, 12 (2010), 2544–2558.

[59] Teun A Van Dijk. 1993. *Elite discourse and racism.* Vol. 6. Sage.

[60] Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: a typology of abusive language detection subtasks. *arXiv preprint arXiv:1705.09899* (2017).

[61] Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of NAACL-HLT*. 88–93.

[62] Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. 2018. Hate Speech on Twitter A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. *IEEE Access* (2018).

[63] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management.* ACM, 625–631.

[64] Matthew L Williams and Pete Burnap. 2016. Cyberhate on social media in the aftermath of Woolwich: A case study in computational criminology and big data. *British Journal of Criminology* 56, 2 (2016), 211–238.

[65] Matthew L Williams and Jasmin Tregidga. 2014. Hate crime victimization in Wales: Psychological and physical impacts across seven hate crime victim types. *British Journal of Criminology* 54, 5 (2014), 946–967.

[66] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing.* Association for Computational Linguistics, 347–354.

[67] Ruth Wodak. 2009. *Discursive construction of national identity.* Edinburgh University Press.

[68] Ruth Wodak and Norman Fairclough. 2013. Critical discourse analysis. (2013).

[69] Ruth Wodak and Martin Reisigl. 1999. Discourse and racism: European perspectives. *Annual Review of Anthropology* 28, 1 (1999), 175–199.

[70] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee, 1391–1399.

[71] Ziqi Zhang and Lei Luo. 2018. Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. *arXiv preprint arXiv:1803.03662* (2018).

[72] Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018a. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *European Semantic Web Conference.* Heraklion, Crete.

[73] Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018b. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *European Semantic Web Conference.* Springer, 745–760.

[74] Yinggong Zhao, Shujian Huang, Xinyu Dai, Jianbing Zhang, and Jiajun Chen. 2014. Learning word embeddings from dependency relations. In *Asian Language Processing (IALP), 2014 International Conference on.* IEEE, 123–127.

Received ; revised ; accepted