The Use of Deep Learning Distributed Representations in the Identification of Abusive Text

Hao Chen
Applied Intelligence Research Center
Technological University Dublin
hao.chen@mydit.ie

Susan McKeever
Applied Intelligence Research Center
Technological University Dublin
susan.mckeever@dit.ie

Sarah Jane Delany
Applied Intelligence Research Center
Technological University Dublin
sarahjane.delany@dit.ie

Abstract

The selection of optimal feature representations is a critical step in the use of machine learning in text classification. Traditional features (e.g. bag of words and n-grams) have dominated for decades, but in the past five years, the use of learned distributed representations has become increasingly common. In this paper, we summarise and present a categorisation of the state-of-the-art distributed representation techniques, including word and sentence embedding models. We carry out an empirical analysis of the performance of the various feature representations using the scenario of detecting abusive comments. We compare classification accuracies across a range of off-the-shelf embedding models using 10 labelled datasets gathered from different social media platforms. Our results show that multi-task sentence embedding models perform best with consistently highest classification results in comparison to other embedding models. We hope our work can be a guideline for practitioners in selecting appropriate features in text classification task, particularly in the domain of abuse detection.

Introduction

When using supervised machine learning approaches to tackle the task of text classification, the text must first be transformed into an interpretable and compact representation of its content prior to its input to an algorithm. For many years, the dominant approach to feature representation for text has been based upon bag of words or n-grams. In this traditional approach, a term document matrix is used, where each text document is represented as a numeric vector of feature occurrence (denoted by 0 or 1) or feature frequency. However, there are several downsides of using this traditional feature representation. The vector is typically sparse as each dimension represents a specific term from the training corpus. In addition, the ordering of words is lost, which reduces the ability to capture semantic or syntactic aspects of the content. To alleviate these issues, previous studies used feature selection techniques to reduce the feature space, and feature engineering to supplement missing language information. For example, Chen et al. (2017b) applied document frequency to remove 50% of features without damaging the performance of the classification algorithm; Agarwal et al. (2011) exploited part of speech (POS) information, and the occurrence of negation (e.g. 'not') to classify the sentiment polarity; For the task of identifying abusive comments, Dadvar et al. (2013) incorporated expert domain knowledge into feature engineering where the features were designed by several experts who have a strong background in social studies and psychological science.

However, hand-crafting of features requires domain specific knowledge, which limits the generalisation ability of the classifier. Recent research has explored the use of distributed representations where the text content is mapped into a fixed-length vector by a pre-trained embedding model. Given that the embedding model is trained on a general language corpus, it preserves intrinsic language information, providing richer input to the downstream text classification algorithm. Word level distributed representations, known as word embeddings, have been a success story in natural language processing (NLP) since Mikolov et al. (2013a) proposed word2vec. This word embedding model captures semantic and syntactic aspects of words through a neural network architecture, and has been widely used as an input for many downstream NLP tasks such as language modeling (Kiros et al. 2015), text classification (Wang et al. 2015), and multilingual translation (Mikolov, Le, and Sutskever 2013) etc. By contrast, sentence level distributed representations, known as sentence embeddings, have been relatively underdeveloped thus far. A few studies have started to explore general sentence embeddings in recent years. For example, Hill et al. (2016) proposed two language models that can be used to generate sentence embeddings; Cer et al. (2018) presented a model to encode sentences into a dense vector. However, there is little work to systematically evaluate sentence embedding models due to the lack of gold-standard corpora. Unlike word embeddings which can be evaluated by the use of a dictionary, the quality of sentence embeddings can only be evaluated through comparative results from downstream NLP tasks, such as text classification.

In this work, we assess and compare distributed feature representations, including word embeddings and sentence embeddings, using a classification task - the identification of abusive comments written in English on social media websites. Our contribution is twofold: First, we summarise the cutting-edge distributed representation techniques and categorise them based on how they are generated; Second, we carry out an empirical analysis of the performance of these distributed features using the scenario of abusive comments detection. To generalise our results, we carry out experiments on multiple datasets across a variety of data sources.

The remainder of this paper is framed as follows. In Section 2, we present a categorisation of general distributed feature representations. Then, we review the state of the art in feature representation for detecting abusive content; Section 3 describes the datasets and methodology that we have used to assess the performance of different feature representations; In Section 4, the experiments used to compare the different representations are explained and results analysed; Section 5 is our conclusion and future work.

Literature Review

Distributed representations have existed in natural language processing for years (Hinton et al. 1984) but have become more widespread with the availability of deep learning models to facilitate the training of representations. There are two types of distributed representation used for text: word and sentence. Word distributed representation maps a single word to a vector. Sentence distributed representation maps blocks of text to a vector. For this paper, we standardise the terminologies, using 'sentence embedding' to cover 'sentence embedding', 'paragraph embedding', and 'document embedding', all of which refer to a distributed representation for a chunk of text content. Fig. 1 shows our categorisation of distributed representations. There are two types of word embedding models, predictive and co-occurrence matrix. For distributed representations at sentence level, the straightforward approach is the aggregation of word vectors that occur in the sentence, which we term 'simple calculation model'. Beyond this approach, the most recent approach is pre-training of sentence embedding on a large language corpus. The generated sentence vector can be directly used in the downstream classification task. As shown in Fig. 1, we categorise pre-trained sentence level representations based on their requirement or not for supervised learning.

Word Level Distributed Representation

Word embedding is now an integral part of text classification. It has been widely used since Tomas Mikolov et al. (2013a) proposed Word2vec, a predictive word embedding model. This model is based on a three-layer neural network that leverages the surrounding information to predict the central word (known as CBOW) or uses the central word to predict the surrounding information (known as Skip-gram). However, it has no ability to capture global language information due to the lack of local context. Glove (Pennington, Socher, and Manning 2014), which is a co-occurrence matrix model, provides better word representation by using global matrix factorisation.

While both Word2vec and Glove are the predominant models for word representation, alternatives have been proposed in recent years. Bojanowski et al. (2016) introduced a modification to Skip-gram named 'fastText' which takes into account the word component information via the integration of character n-grams. This model addresses the lack of word morphology knowledge, and tackles the problem of out-of-vocabulary (OOV) words where the words are unknown in the test dataset. Another criticism of the original Word2vec is that it is unable to capture word polysemy where a word has several meanings. To alleviate this issue, Peters et al. (2018) proposed an ELMo framework (Embeddings from Language Models) that can generate word vectors dynamically based on context in a downstream dataset, rather than a word statically represented by one vector. In addition, Lai et al. (2016) proposed that using a suitable domain corpus to train word embeddings benefits the downstream task.

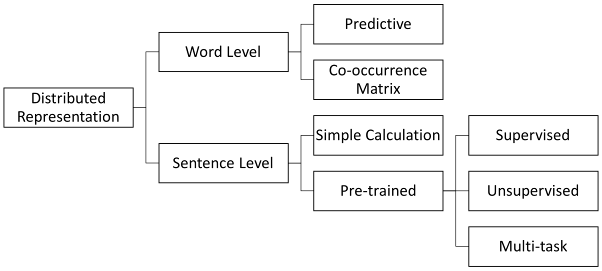

Figure 1: The Category of Distributed Representations

Sentence Level Distributed Representation

The success of word embedding has motivated the generation of 'universal' embedding for groupings of text content. The straightforward approach to generating sentence representation from pre-trained word embedding is based on the simple model of word vector aggregation. Both Wieting et al. (2015) and Arora et al. (2016) show the performance of word vector averaging is comparable to the more complex approaches using neural networks such as recurrent neural network with long short-term memory units (LSTM) for the task of sentiment analysis. Ruckle et al (2018) generalised the concept of word embedding averaging to use power mean, where the average value can be replaced by either maximum or minimum values. The concatenation of various power mean word embeddings for inducing the sentence representation outperforms the simple averaging approach in several text classification tasks.

Beyond the simple calculation model, the development of pre-trained sentence embeddings has emerged only recently. We categorise them into three types: unsupervised, supervised and multi-task model based on the requirement or not for supervised learning in the pre-trained models.

Unsupervised Learning Unsupervised learning models learn sentence representations as a by-product of language modelling where word sequences are used to make probabilistic predictions of surrounding text. The common unsupervised models include:

- **Paragraph2vec** Le et al. (2014) modified the word2vec algorithm adding a paragraph token which can learn a representation for blocks of text.
- **Skip-Thought** Inspired by the Skip-gram, Kiros et al. (2015) proposed Skip-Thought which uses the central sentence to reconstruct the surrounding sentences.
- **Quick-Thoughts** A modification of Skip-Thought was proposed by Logeswaran et al. (2018). Quick-Thoughts replaces the prediction of surrounding sentences by a classifier which aims to choose the target sentence amongst a set of candidate sentences.
- **SDAE** The Sequential Denoising Autoencoder is a neural network composed of the encoder (process the input to the feature map) and the decoder (process the feature map to the output). The encoder of SDAE (Hill, Cho, and Korhonen 2016) is impaired by using some noise functions. The objective of the model is to recover the original data from this impaired version.
- **FastSent** FastSent (Hill, Cho, and Korhonen 2016) is an efficient extension of Skip-Thought where sentences are encoded using a simple calculation model (sum of word embeddings) instead of a sequential model (e.g. RNN).

A variety of data sources have been used for training the various sentence embedding models. For example, Hill et al. (2015) used the definitions of vocabulary in a dictionary as a training corpus; Wieting et al. (2015) proposed a general sentence embedding model that is trained on the Paraphrase Database (a large volume of pairs of phrases); The Book Corpus, which consists of 70 million sentences from over 7000 books, has been used in several studies (Hill, Cho, and Korhonen 2016; Pagliardini, Gupta, and Jaggi 2017; Gan et al. 2016).

Supervised Learning Although unsupervised learning is the prevailing method used to generate sentence representations, recent work has shown that supervised learning can also achieve high-quality sentence representations. Conneau et al. (2017) introduced **InferSent** where the sentence embedding model is pre-trained on a supervised dataset. It uses the Stanford Natural Language Inference (SNLI) corpus (Bowman et al. 2015) which consists of pairs of sentences with 3 manual labels: entailment, contradiction, and neutral. The generated embeddings show consistently better performance than the representation generated by the unsupervised learning models (e.g. Skip-Thought) on a wide range of downstream NLP tasks such as binary classification and semantic textual similarity.

Multi-Task Learning One difficulty of conducting supervised learning is the decision on which supervised data source will generate optimal sentence representations. To prevent overfitting of supervised learning embeddings to a specific domain, several studies have used multi-task learning. In brief, multi-task is a combinational approach that uses both unsupervised and supervised learning. For example, Yang et al. (2018) presented a framework based on two neural networks. The first neural network is a language model that is trained on an unsupervised dataset. The second neural network is a classifier that is trained on a supervised dataset. The resultant sentence embedding achieves state-of-the-art performance on the Semantic Textual Similarity (STS) benchmark. Likewise, Subramanian et al. (2018) presented a framework that combines various training objectives (e.g. neural machine translation, natural language inference, language parsing) in a single model. According to their observation, the syntax information can be learned from the task of natural language inference, and the semantic information can be learned from the task of neural machine translation. In addition, Cer et al. (2018) proposed a universal sentence encoder where the sentence embedding model is trained using a variety of data sources including unsupervised (e.g. Wikipedia) and supervised (e.g. SNLI).

Features for Abusive Detection

At present, automatic detection of abusive user comments on social media sites relies heavily on supervised text classification techniques. Most existing work has focused on feature engineering which aims to identify high quality features that can be of benefit to the classifier. Traditional content-based features (e.g. bag of words or n-grams) (Chen, McKeever, and Delany 2017b; Xu et al. 2012; Mangaonkar, Hayrapetian, and Raje 2015; Sood, Churchill, and Antin 2012) have been widely used. In addition, classifier accuracy has been enhanced by adding linguistic information (Zhang, Robinson, and Tepper 2018) such as the usage of capitalisation letters, hashtags, emoji and punctuation. Likewise, Nobata et al. (2016) combined n-grams with syntactic part-of-speech (POS). Apart from language-based features, Dadvar et al. (2012) demonstrated that taking gender-specific features into account boosts the discrimination capacity of an SVM classifier. Chen et al. (2012) proposed an effective framework that includes the user's conversation history based on user's identification. A significant number of studies (Van Hee et al. 2018; Reynolds, Kontostathis, and Edwards 2011; Sood, Antin, and Churchill 2012) relied on predefined profane words to improve the classifier performance. However, Hosseinmardi et al. (2015) illustrated a high proportion of negative words do not in fact constitute abuse. The critical weaknesses of feature engineering are the requirement for expert domain knowledge and time. The fact that the designed features are dataset dependent and typically cannot be generalised across different sources. For example, Chatzakou et al. (2017) suggested a promising approach to the task of detecting abusive language is combining user profile (e.g. the age of user account, the number of tweets the user has made), but this information can only be extracted from a Twitter dataset.

Recent research has focused on distributed feature representations. Most distributed representations are at word level where the pre-trained word embeddings are the input of a deep learning classifier. The classic deep learning architectures, CNN and RNN, were used by Gamback et al. (2017), Park et al. (2017), and Gao et al. (2017). Zhang et al. (2018) introduced a complex deep learning model that combines both CNN and RNN and achieved better results than using CNN alone. Founta et al. (2018) proposed an advanced framework that augments the metadata (e.g. emoticons us-

Table 1: The Summary of 10 Datasets

	Published	Data Source	#of Instances	Avg. #Words	Class Dist. (Pos./Neg.)	Features	Classifier	Metrics
D1	(Xu et al. 2012)	Twitter	3110	15	42/58	Ngrams	SVM	Recall
D2	(Dadvar, Trieschnigg, and de Jong 2014)	YouTube	3466	211	12/88	Rule-Based	SVM	AUC
D3	(Bayzick, Kontostathis, and Edwards 2011)	MySpace	1710	337	23/77	Lexical	Rule-Based	Overall Acc
D4	(Reynolds, Kontostathis, and Edwards 2011)	Formspring	13153	26	6/94	Lexical	Rule-Based	Overall Acc
D5	(Yin et al. 2009)	Kongregate	4802	5	1/99	Ngrams	SVM	Pos. Recall
D6	(Yin et al. 2009)	SlashDot	4303	94	1/99	Ngrams	SVM	Pos. Recall
D7	(Yin et al. 2009)	MySpace	1946	56	3/97	Ngrams	SVM	Pos. Recall
D8	(Mangaonkar, Hayrapetian, and Raje 2015)	Twitter	1340	13	13/87	Ngrams	LR	Recall
D9	(Chen, McKeever, and Delany 2017c)	News Forum	2000	59	21/79	Ngrams	SVM	Recall
D10	(Wulczyn, Thain, and Dixon 2017)	Wikipedia	115864	67	12/88	Ngrams	LR,MLP	AUC

age, and sentiment polarity), which increases the area under the curve (AUC) measure by approximately 5%. In addition to the use of distributed word representations, some works explored distributed features at sentence level. Both Djuric et al. (2015) and Zhao et al. (2018) applied unsupervised approach, paragraph2vec (Le and Mikolov 2014), to generate low-dimensional sentence embedding for abusive comments classification. Badjatiya et al. (2017) used the simple calculation approach, simply averaging the word embeddings to produce sentence representations. This improves the F1 measurement by 18% in comparison to the use of n-grams.

Methodology

The overall goal of our work is to evaluate the various distributed representations for the task of abusive content detection. In this section, we detail the methodology that we used in our work.

Datasets & Pre-processing

The lack of gold-standard labelled training datasets is a major obstacle to the task of classifying abusive user generated comments. Most research efforts in this domain have used datasets that are only available for their own studies. In this paper, we perform our experiments on the 10 publicly available datasets (Xu et al. 2012; Dadvar, Trieschnigg, and de Jong 2014; Bayzick, Kontostathis, and Edwards 2011; Reynolds, Kontostathis, and Edwards 2011; Yin et al. 2009; Mangaonkar, Hayrapetian, and Raje 2015; Chen, McKeever, and Delany 2017c; Wulczyn, Thain, and Dixon 2017) derived from a variety of social media platforms including Twitter, YouTube, MySpace, and Wikipedia. All datasets were labelled manually as part of their original publication work as abusive or not abusive. Table 1 gives an overview of the 10 datasets, including the basic properties such as the number of total instances, average number of words across instances, the class distribution of positive instances (abuse) to negative instances (non-abuse), and the approaches (features, classifier, and evaluation metric) that were used in the original research. We acknowledge that these datasets cover different types of abuse such as 'cyberbullying' or 'harassment'. In this paper, we categorise them as abusive content. The 10 datasets will be referred to as D1, D2 through to D10 for the rest of paper.

The original studies associated with these 10 datasets use traditional feature representations. The most common representation used are word/character n-grams. D3 and D4 use lexical features which are based on whether the text content contains pre-defined profane terms; Researchers who generated D2 incorporate information from abuse experts and design a set of rules to identify target comments. SVM is the classifier of choice in the 6 out 10 datasets. Other algorithms such as logistic regression (LR) and multilayer perceptron (MLP) are used in D8 and D10 respectively.

Most datasets in Table 1 are imbalanced with a small proportion of positive (abusive) instances. In particular, D5, D6 and D7 only contain less than 5% of user comments are labelled as abuse. To address this issue, we use resamplling techniques to re-balance the class distribution before feeding into the classifier algorithms. Resampling is applied to training data only. We use two resampling techniques, oversampling and undersampling. For most of the datasets (D2 to D9) with a small quantity of instances, we randomly oversample the minority instances to increase the proportion of abusive comments. For the large dataset (D10), we carry out the opposite approach to randomly undersampling majority (non-abuse) instances. After resampling, each dataset has a balanced class distribution.

The data is normalised in the following ways: All non-English characters are removed, and then all characters changed to lowercase; Mentioned user names, which are preceded by the symbol '@', are replaced by the anonymous term as '@username'; All hyperlinks are unified as the generic term 'url_links'; Considering the social media user comments are typically short, we do not implement stemming or remove stop-words.

Features & Classifiers

Our aim is to compare the performance of the various feature representations shown in Fig. 1 when applied to the task of abusive comment classification. We acknowledge that social media provides other sources as potentially useful features, for example, likes, number of followers, number following. However this information is not consistently available across different types of social media platforms. As our experiments are carried out on the multiple datasets, we do not use such dataset-dependent features. We set the baseline feature representation as traditional n-grams where the text content is represented by n continuous sequential words. Based on our previous work (Chen, McKeever, and Delany 2017b; 2017a), we identify that 1 to 4 word grams achieves the

Table 2: The Configurations of Different Feature Representations with Classifiers

Feature Category	Level of Representation	Implementation	Feature	Classifier	Config.
Traditional			N-grams	SVM	1
Distributed	Word Level (Word Embedding)		Glove	CNN	2
			Glove	Bi-LSTM	3
	Sentence Level (Sentence Embedding)	Simple Calculation	Avg. WV	SVM	4
		Pre-trained Supervised	InferSent	SVM	5
		Pre-trained Unsupervised	Sent2Vec	SVM	6
		Pre-trained Multi-task	SentEncoder	SVM	7
		Pre-trained Multi-task	GenSen	SVM	8

best performance. We then apply document frequency reduction to remove the features that occur the most and the least frequently in the dataset. To assess the word level distributed representations shown in Fig. 1, we apply pre-trained word embeddings. We input the vectors of words to a deep learning-based classifier. For the sentence level representations in Fig. 1, we use two approaches. To assess the first 'simple calculation' approach, we average the vectors of the words in the user comment, producing a single vector to represent the text in the comment. We use this approach for comment representation to fed to the SVM classifier. In the second approach, we use multiple existing models that generate sentence embeddings. In such cases, the user comment is mapped to a vector which then is treated as an input for the SVM classifier.

We use SVM as the abuse detection classifier for the traditional n-grams feature set as it is a widely used classification algorithm with high quality performance in text classification. However, using SVM with word embeddings as input is not practical as the individual comment is represented by a set of word vectors (matrix) instead of a vector. We therefore use two common deep learning neural networks, convolutional neural network (CNN) and recurrent neural network (Bi-LSTM). In detail, we use a sequence of word vectors as the input for both models and softmax as the output layer. We acknowledge that choosing hyper-parameters is an important element for the deep learning classifier, and optimisation of these parameters requires a validation dataset. Since our experiments are ran on the multiple datasets which are not large enough to provide a separate validation set, we use the same hyper-parameters for all datasets. We use the optimal settings from the guidelines provided by Zhang et al. (2015) for CNN, and Reimers et al. (2017) for RNN. The hyper parameters used are shown in Table 3.

Evaluation Metric

As shown in Table 1, we note that previous studies associated with our datasets for abusive text classification use a variety of evaluation metrics. For example, the studies of D3 and D4 use accuracy which is one of the common performance measures for text classification. A majority evaluate the classifier using recall (D1, D8, D9). In particular positive recall (recall of the abusive class) has been seen as important by researchers using D5, D6 and D7. Area Under the ROC Curve (AUC) is also used in some cases (D2 and D10).

In our work, we standardise the evaluation metric as class accuracy (recall) as it indicates the ability of the model to identify all instances of a specific class. We assume that in a real-life scenario, the cost of false negatives (abusive content identified as non-abusive) is higher than false positives (non-abusive content identified as abuse). Therefore, we focus on positive (abusive) recall. We also report average recall to examine the performance across both classes. The methodology used is stratified 10-fold cross validation.

Table 3: The Details of Hyper-parameters for Deep Learning Classifiers (Reg. is a short for Regularization)

	Activation Function	Filter Size	Feature Maps	Dropout Rate	Reg.	Recurrent Units	Mini Batch	Epoch
CNN	ReLu	3,4,5	100	0.5	*l2*	NA	50	50
RNN		NA	NA			100		

Experiments & Results

We compare the performance of the distributed feature representations presented in our Figure 1 categorisation, using abusive text detection as our associated classification challenge. Our experiments are performed on the 10 labelled datasets (Table 1). For our baseline approach, we use traditional n-grams for feature representation and an SVM classifier (Config. 1). We then apply seven other feature representation/classifier configurations, as shown in Table 2 (Configs 2 to 8). For representation at word level, we use Glove pre-trained word embeddings rather than Word2vec as the former slightly outperforms the latter based on our previous finding (Chen, McKeever, and Delany 2018). The text content is mapped to a sequence of word vectors, and used as input to the classifier. We evaluate the performance of word embeddings on the two popular deep learning architectures, CNN (Config. 2) and Bi-LSTM (Config. 3). For representation at sentence level, we use two types of embedding techniques, simple calculation and pre-trained. Both techniques convert an individual user comment to a dense vector which is fed into the SVM classifier. The simple calculation vector is obtained by averaging the vectors of the words in the text content (Config. 4). For the pre-trained model, we use four sentence embeddings covering three classes in our categorisation, supervised, unsupervised and multi-task. The four proposed sentence embedding methods are **InferSent** (Config. 5), **Sent2Vec** (Config. 6), **SentEncoder** (Config. 7), and **GenSen** (Config. 8). They are recently released by Conneau et al. (2017), Pagliardini et al. (2017), Cer et al. (2018), and Subramanian et al. (2018) respectively.

As shown in Table 4, these sentence embeddings are trained by various types of neural network architectures using a variety of training corpora. Both InferSent and Sent2Vec use a single training corpus while multiple training corpora are used in SentEncoder and GenSen. For the selection of training models, Conneau et al. (2017) compare 6 different neural network architectures and indicate the complex BiLSTM with max pooling obtains the best sentence embedding. However, Cer et al. (2018) who proposed SentEncoder use the simple averaging neural network as it uses less resource consumption than other deep neural networks. Sent2vec is based on a modification of Word2vec (Mikolov et al. 2013b) while GenSen (Subramanian et al. 2018) uses Bi-directional Gated Recurrent Unit (GRU) as the encoder to generate the sentence distributed representations. These four pre-trained sentence embeddings have different sentence dimensions. Conneau et al. (2017) investigate the 4 scales of dimensions ranging from 512 to 4096, noting that higher dimensions are generally better than the lower dimensions. Nevertheless, the higher dimensions need larger computational resources, so the decision of dimension is trade-off and scenario-dependent.

Table 4: Details of Each Sentence Embedding Model

Config.	Sentence Embedding	Dimension	Category	Training Corpora	Training Model
5	InferSent	4096	Supervised	SNLI	BiLSTM-Max
6	Sent2Vec	600	Unsupervised	Wiki	Paragraph2vec
7	SentEncoder	512	Multi-task	Wiki, Web News, Q&A, SNLI etc.	Deep Averaging Network
8	GenSen	2048	Multi-task	BookCorpus, SNLI, NMT etc.	BiGRU

Our experiments compare the 8 explained configurations applied to the 10 datasets. We present the results in Table 5. We mainly focus on positive (i.e. abusive text) recall (%) as it allows us to see the accuracy of identifying abusive comments. Average recall (%) are also provided to make sure the non-abusive comments are also correctly classified. We highlight the best results in each category. In addition, false positive (%) is given (Table 6) to indicate the error rate of non-abusive examples incorrectly identified as abusive. In order to statistically validate the results, we use the Friedman test for multiple comparisons, followed by Dunn-Bonferroni as the post hoc statistical test for pairwise comparisons. The significance level is set to 0.05 (5%).

Overall, the SentEncoder (Config. 7) achieves the highest positive recall amongst both the 5 sentence embedding configurations and across all 8 feature representations. It achieves the best results on 9 out of the 10 datasets even though this embedding is generated by the simplest neural network architecture, a deep averaging network, compared to the other embeddings that use complex neural networks such as RNN (Configs. 5 & 8). We suggest that the strong results obtained using SentEncoder is linked to the use of multiple training corpora. Unlike the other sentence embeddings, SentEncoder uses the largest language corpora that covers a variety of language styles from different sources, including both formal style language (e.g. Wikipedia, News articles) and conversational style language (e.g. Q&A forum). The diversity of training language corpora appears to be a promising way to achieve high quality sentence representations. Notably, SentEncoder has the lowest dimension (512) amongst all the pre-trained sentence embedding models. Given that user comments are typically short in length, high-dimension vectors may contain noise information and adversely impact the results. For example, the exception where SentEncoder does not achieve the best result is for D3. D3 is the dataset with the highest average length comments (337 words) amongst the 10 datasets.

For sentence embeddings generated via the simple calculation model, we observe that averaged word vectors (Config. 4) perform competitively when compared to the other three sentence embeddings (Configs. 5, 6 & 8). The pairwise statistical test for positive recall results shows no significant difference amongst them. However, it is interesting to note that the averaged word embedding model (Config. 4) outperforms the two word embedding with deep learning neural networks configurations (Configs. 2,3). For example, the positive recall of using CNN model in D7 (Config. 2) is 14%, which is worse than the result of Config. 4 at approximately 35%. We acknowledge that the comparison of Config. 4 with Configs. 2 & 3 is difficult given the use of different classifiers. As all of them use Glove word embedding as the input, we can assume the SVM classifier performs better than the two deep neural networks. In our previous work (Chen, McKeever, and Delany 2018), we found that the use of resampling provides a greater boost to performance for SVM classifiers than deep learning neural networks. However, for the comparison between the two deep learning neural networks (CNN and BiLSTM), we note that there is no significant difference.

In addition, we observe that the baseline configuration (Config. 1) achieves strong results. Our statistical significance test shows there are no differences between Config. 1 and most of distributed representations with the exception of the SentEncoder sentence embedding (Config. 7). The traditional n-grams feature representation can provide a solid baseline for abusive content detection.

While identification of abusive text is our focus, we are cognisant of the risk of blocking valid user posts via classification errors with the negative class. To explore this, we examine the false positive rates for each configuration across the datasets, as shown in Table 6. The InferSent (Config. 5) has the lowest false positive rate amongst 5 sentence embedding approaches, with the lowest incidence of categorising safe comments as abusive. However, testing statistical significance across the 5 sentence embedding configurations (Config. 4 - Config. 8), we note that the only significant differences in the false positive results are the better perfor-

Table 5: The Positive Recall % (Average Recall %) of 8 Configurations on 10 Datasets. The best result is bold font.

Config.	Features	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
1	Traditional	70(75)	35(62)	91(93)	62(77)	58(78)	12(56)	18(58)	65(78)	33(60)	80(86)
2	Word Embedding	**73(73)**	4(51)	**93(95)**	34(66)	**57(78)**	11(55)	14(57)	59(78)	29(61)	75(85)
3		68(73)	**6(51)**	81(89)	**45(71)**	50(75)	**14(57)**	**18(58)**	**60(77)**	**32(60)**	**76(81)**
4	Sentence Embedding	65(71)	30(59)	66(76)	59(74)	58(76)	51(71)	48(68)	77(85)	48(66)	73(81)
5		77(77)	21(56)	**93(95)**	59(77)	60(80)	28(64)	23(61)	66(82)	41(65)	**82(86)**
6		65(70)	38(60)	92(94)	61(74)	52(75)	18(58)	26(60)	61(76)	45(62)	74(82)
7		**77(78)**	**42(63)**	85(89)	**77(84)**	**69(84)**	**55(76)**	**58(76)**	**89(93)**	**58(72)**	**82(86)**
8		74(71)	29(58)	90(93)	59(75)	48(74)	22(61)	19(59)	75(86)	42(63)	**82(86)**

Table 6: The False Positive % (Error Rate) of 8 Configurations on 10 Datasets. The best result is bold font.

Config.	Features	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
1	Traditional	20	11	5	8	2	0	2	9	13	8
2	Word Embedding	27	**2**	**3**	**2**	1	1	**1**	**3**	**7**	**5**
3		**22**	4	**3**	3	**0**	**0**	2	6	12	14
4	Sentence Embedding	23	12	14	11	6	9	12	7	16	11
5		23	**9**	**3**	**5**	**0**	**1**	**1**	**2**	**11**	**10**
6		25	18	4	13	2	2	6	9	21	**10**
7		**21**	16	7	9	1	3	6	3	14	**10**
8		32	13	4	9	1	1	1	3	16	**10**

mance of InferSent (Config. 5) over the Simple Calculation (Config. 4) and Sent2Vec (Config. 6). From earlier results discussion, SentEncoder(Config. 7) was the best performing configuration on abusive text detection. Since the difference between SentEncoder (Config. 7) and InferSent is negligible, we determine that the SentEncoder is still the best sentence embedding model for abusive language detection, considering both abusive detection and misclassification of valid non-abusive posts.

We also note that the false positive rates of Config. 4 are significantly higher than both deep learning models (Config. 2 & 3). Our earlier finding of the Simple Calculation Config. 4 as a better detector of abusive content than the deep learning model configurations is now caveated by an awareness of the higher error rate incurred by Config. 4 on classifying safe comments as abusive.

Conclusion & Future Work

The focus of this work was to investigate the performance of distributed representations for text when applied to a specific text classification task - abusive user comment detection. Firstly, we summarised and provided a categorisation of the various distributed representation approaches, incorporating word and sentence (sentence, document, paragraph) embedding models. Secondly, we conducted an empirical comparison of the effectiveness of the various distributed representation when used for classification of abusive text. Based on our results across 10 social media datasets, we conclude: (1) The multi-task sentence embedding significantly outperforms the other pre-trained sentence embeddings; (2) Using simple averaging word embedding with an SVM classifier achieves good results, and out performs the approach of word embedding with deep learning neural networks. We attribute this, based on our previous work (Chen, McKeever, and Delany 2018), to resampling the imbalanced training datasets. (3) Traditional n-grams show strong performance in comparison to the several of the distributed representation approaches. Future work could focus on combining both traditional features and distributed features to model a classifier for the task of abusive detection. Furthermore, we would like to produce a new sentence embedding model that uses a relevant abusive language training corpus, which may improve the state-of-the-art results.

References

Agarwal, A.; Xie, B.; Vovsha, I.; Rambow, O.; and Passonneau, R. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, 30–38. Association for Computational Linguistics.

Arora, S.; Liang, Y.; and Ma, T. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Badjatiya, P.; Gupta, S.; Gupta, M.; and Varma, V. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, 759–760. International World Wide Web Conferences Steering Committee.

Bayzick, J.; Kontostathis, A.; and Edwards, L. 2011. Detecting the presence of cyberbullying using computer software. In *3rd Annual ACM Web Science Conference (WebSci '11)*, 1–2.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606.*

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chatzakou, D.; Kourtellis, N.; Blackburn, J.; De Cristofaro, E.; Stringhini, G.; and Vakali, A. 2017. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference*, 13–22. ACM.

Chen, Y.; Zhou, Y.; Zhu, S.; and Xu, H. 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)*, 71–80. IEEE.

Chen, H.; McKeever, S.; and Delany, S. J. 2017a. Abusive text detection using neural networks. In *AICS*.

Chen, H.; McKeever, S.; and Delany, S. J. 2017b. Harnessing the power of text mining for the detection of abusive content in social media. In *Advances in Computational Intelligence Systems*. Springer. 187–205.

Chen, H.; McKeever, S.; and Delany, S. J. 2017c. Presenting a labelled dataset for real-time detection of abusive user posts. In *Proceedings of the International Conference on Web Intelligence*, 884–890. ACM.

Chen, H.; McKeever, S.; and Delany, S. J. 2018. A comparison of classical versus deep learning techniques for abusive content detection on social media sites. In *International Conference on Social Informatics*, 117–133. Springer.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Dadvar, M.; Jong, d. F.; Ordelman, R.; and Trieschnigg, D. 2012. Improved cyberbullying detection using gender information. In *Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop (DIR 2012)*. University of Ghent.

Dadvar, M.; Trieschnigg, R. B.; and de Jong, F. M. 2013. Expert knowledge for automatic detection of bullies in social networks. In *25th Benelux Conference on Artificial Intelligence, BNAIC 2013*. TU Delft.

Dadvar, M.; Trieschnigg, D.; and de Jong, F. 2014. Experts and machines against bullies: A hybrid approach to detect cyberbullies. In *Canadian Conference on Artificial Intelligence*, 275–281. Springer.

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

Founta, A.-M.; Chatzakou, D.; Kourtellis, N.; Blackburn, J.; Vakali, A.; and Leontiadis, I. 2018. A unified deep learning architecture for abuse detection. *arXiv preprint arXiv:1802.00385*.

Gambäck, B., and Sikdar, U. K. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, 85–90.

Gan, Z.; Pu, Y.; Henao, R.; Li, C.; He, X.; and Carin, L. 2016. Learning generic sentence representations using convolutional neural networks. *arXiv preprint arXiv:1611.07897*.

Gao, L., and Huang, R. 2017. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*.

Hill, F.; Cho, K.; Korhonen, A.; and Bengio, Y. 2015. Learning to understand phrases by embedding the dictionary. *arXiv preprint arXiv:1504.00548*.

Hill, F.; Cho, K.; and Korhonen, A. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Hinton, G. E.; McClelland, J. L.; Rumelhart, D. E.; et al. 1984. *Distributed representations*. Carnegie-Mellon University Pittsburgh, PA.

Hosseinmardi, H.; Mattson, S. A.; Rafiq, R. I.; Han, R.; Lv, Q.; and Mishra, S. 2015. Detection of cyberbullying incidents on the instagram social network. *arXiv preprint arXiv:1503.03909*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, 3294–3302.

Lai, S.; Liu, K.; He, S.; and Zhao, J. 2016. How to generate a good word embedding. *IEEE Intelligent Systems* 31(6):5–14.

Le, Q., and Mikolov, T. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, 1188–1196.

Logeswaran, L., and Lee, H. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Mangaonkar, A.; Hayrapetian, A.; and Raje, R. 2015. Collaborative detection of cyberbullying behavior in twitter data. In *Electro/Information Technology (EIT), 2015 IEEE International Conference on*, 611–616. IEEE.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Mikolov, T.; Le, Q. V.; and Sutskever, I. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

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

Pagliardini, M.; Gupta, P.; and Jaggi, M. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

Park, J. H., and Fung, P. 2017. One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Reimers, N., and Gurevych, I. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.

Reynolds, K.; Kontostathis, A.; and Edwards, L. 2011. Using machine learning to detect cyberbullying. In *Machine learning and applications and workshops (ICMLA), 2011 10th International Conference on*, volume 2, 241–244. IEEE.

Rücklé, A.; Eger, S.; Peyrard, M.; and Gurevych, I. 2018. Concatenated p-mean word embeddings as universal cross-lingual sentence representations. *arXiv preprint arXiv:1803.01400*.

Sood, S.; Antin, J.; and Churchill, E. 2012. Profanity use in online communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1481–1490. ACM.

Sood, S. O.; Churchill, E. F.; and Antin, J. 2012. Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology* 63(2):270–285.

Subramanian, S.; Trischler, A.; Bengio, Y.; and Pal, C. J. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Van Hee, C.; Jacobs, G.; Emmery, C.; Desmet, B.; Lefever, E.; Verhoeven, B.; De Pauw, G.; Daelemans, W.; and Hoste, V. 2018. Automatic detection of cyberbullying in social media text. *arXiv preprint arXiv:1801.05617*.

Wang, X.; Liu, Y.; Chengjie, S.; Wang, B.; and Wang, X. 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. 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)*, volume 1, 1343–1353.

Wieting, J.; Bansal, M.; Gimpel, K.; and Livescu, K. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Wulczyn, E.; Thain, N.; and Dixon, L. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, 1391–1399. International World Wide Web Conferences Steering Committee.

Xu, J.-M.; Jun, K.-S.; Zhu, X.; and Bellmore, A. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 656–666. Association for Computational Linguistics.

Yang, Y.; Yuan, S.; Cer, D.; Kong, S.-y.; Constant, N.; Pilar, P.; Ge, H.; Sung, Y.-H.; Strope, B.; and Kurzweil, R. 2018. Learning semantic textual similarity from conversations. *arXiv preprint arXiv:1804.07754*.

Yin, D.; Xue, Z.; Hong, L.; Davison, B. D.; Kontostathis, A.; and Edwards, L. 2009. Detection of harassment on web 2.0. *Proceedings of the Content Analysis in the WEB* 2:1–7.

Zhang, Y., and Wallace, B. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Zhang, Z.; Robinson, D.; and Tepper, J. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European Semantic Web Conference*, 745–760. Springer.

Zhao, X., and Caverlee, J. 2018. Vitriol on social media: Curation and investigation. In *International Conference on Social Informatics*, 487–504. Springer.