A Comparison of Classical Versus Deep Learning Techniques for Abusive Content Detection on Social Media Sites

Hao Che
Dublin Institute of Technology, hao.chen@mydit.ie

Susan McKeever
Technological University Dublin, susan.mckeever@tudublin.ie

Sarah Jane Delany
Technological University Dublin, sarahjane.delany@tudublin.ie

Recommended Citation

Chen H., McKeever S., Delany S.J. (2018) A comparison of classical versus deep learning techniques for abusive content detection on social media sites. In:(Staab S., Koltsova O., Ignatov D. (eds)) *Social Informatics: SocInfo 2018*. Springer. Lecture Notes in Computer Science, vol 11185.

A Comparison of Classical Versus Deep Learning Techniques for Abusive Content Detection on Social Media Sites

Hao Chen(✉), Susan McKeever, and Sarah Jane Delany

Dublin Institute of Technology, Dublin, Ireland
hao.chen@mydit.ie, {susan.mckeever,sarahjane.delany}@dit.ie

Abstract. The automated detection of abusive content on social media websites faces a variety of challenges including imbalanced training sets, the identification of an appropriate feature representation and the selection of optimal classifiers. Classifiers such as support vector machines (SVM), combined with bag of words or ngram feature representation, have traditionally dominated in text classification for decades. With the recent emergence of deep learning and word embeddings, an increasing number of researchers have started to focus on deep neural networks. In this paper, our aim is to explore cutting-edge techniques in automated abusive content detection. We use two deep learning approaches: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). We apply these to 9 public datasets derived from various social media websites. Firstly, we show that word embeddings pre-trained on the same data source as the subsequent classification task improves the prediction accuracy of deep learning models. Secondly, we investigate the impact of different levels of training set imbalances on classifier types. In comparison to the traditional SVM classifier, we identify that although deep learning models can outperform the classification results of the traditional SVM classifier when the associated training dataset is seriously imbalanced, the performance of the SVM classifier can be dramatically improved through the use of oversampling, surpassing the deep learning models. Our work can inform researchers in selecting appropriate text classification strategies in the detection of abusive content, including scenarios where the training datasets suffer from class imbalance.

Keywords: Text classification · Abuse detection · Deep learning

1 Introduction

An increasing number of social media platforms facilitate users in posting their personal opinions online, resulting in rapid growth in the volume of user-generated content (UGC) over the past decade. This UGC inevitably carries the risk of containing inappropriate, potentially abusive content which aims to deliberately insult other online users through profane or hurtful language. Social

S. Staab et al. (Eds.): SocInfo 2018, LNCS 11185, pp. 117–133, 2018.
https://doi.org/10.1007/978-3-030-01129-1_8

media companies have a responsibility to combat abusive content by assessing or moderating posted content. The moderation strategies used in most websites can be categorised as either pre-published or post-published, depending on whether the moderation process is carried out before or after publication. In pre-published moderation, content posted by users will be checked before it is made available online. Usually, pre-published moderation relies on human moderators (e.g. BBC online news) or simple word filters (e.g. Both YouTube and Facebook provide functionality to allow users to make a list of blocked words). Human moderation of all content is expensive and lacks scalability, while word filters lack the ability to detect more subtle semantic abuse. In post-published moderation, the content is posted directly online, with abusive content detection reliant on crowdsourcing mechanisms, such as user reporting systems (e.g. Twitter) or moderators' determination (e.g. Reddit). In this case, the abusive content may have already resulted in negative consequences as it has been made available to an online audience. Given the huge volume of UGC, reliance on manual moderation of all content is impractical. The development of moderation tools to automatically review abusive content on social media websites is a priority.

Published research initially focused on the use of models based on traditional supervised classifiers in order to tackle abusive content detection. The text content was represented by a set of occurrence-based features such as bag of words or ngrams, and then fed into typical classifiers such as SVM or Naive Bayes. These feature representations count the frequency of words in text content but largely ignore word order and do not capture syntactic information. Adding hand-crafted features that are generated by experts can alleviate the shortcoming of traditional features. Nevertheless, this requires human effort and introduces domain specific dependencies into the model. In recent years, deep learning, as one of the solutions that can extract features automatically, has achieved state-of-the-art performance in many natural language processing (NLP) tasks such as sentiment analysis [13]. Likewise, recent studies of abusive content detection have focused on the use of deep learning based models. However, comparisons of traditional and deep learning approaches are difficult, due to the variety of datasets used across different researchers' work. Most researchers in this domain use their own private datasets, resulting in models that are dependent on their data and that cannot be compared to other work. In this paper, we address this issue by conducting an empirical comparison of traditional classification models and deep learning based models for abusive text detection. We use 9 datasets in order to generate results across a wide spectrum of data sources.

In addition, abusive datasets typically have an imbalance in class distribution, with a very small proportion of abusive instances. This is similar to the online reality (e.g. under 1% of abusive tweets are identified in Twitter [17,38]). In our work, we examine the impact of class imbalance by using multiple imbalanced datasets including both public sources and our own collected dataset. Our contributions are as follows: (1) We demonstrate the improvement on detection results using word embeddings that are pre-trained on a data source that is consistent with the classification data source; (2) Using an empirical comparison, we show that deep learning models have higher detection accuracy than the traditional SVM classifier when trained with extremely imbalanced datasets. However, when oversampling is used to address class imbalance, the performance of the SVM classifier increases far more rapidly than deep learning models; (3) Unlike most previous research efforts which typically use one dataset, we carried out our experiments on 9 datasets, thus generating results that are not tied to a single data source.

The reminder of this paper is structured as follows. Section 2 reviews the literature in the field; Sect. 3 describes the experimental datasets used for our work; Sect. 4 explains the methodologies that we have used to tackle the classification task; Sect. 5 presents the experiments and results; and Sect. 6 concludes the present work and discusses the future work.

2 Related Work

Automatically identifying abusive user-generated content on social media sites has attracted increasing attention from machine learning researchers over recent years. Existing strategies for abuse detection rely primarily on the use of supervised classification. In this section, we focus the literature review on two aspects, traditional machine learning techniques and deep learning neural networks.

2.1 Traditional Machine Learning

Much of the previous research uses traditional supervised classification algorithms to tackle abusive content detection. One of the key steps in generating a successful classification model is the use of appropriate features. The shallow approach to tackle the abuse detection task is to rely on the concept of lexical matching. Reynolds et al. [32] engineered features based on matching content words against a pre-defined profane words list. In order to avoid misspelling and abbreviation, Sood et al. [36] improved on the static keyword-based approach by using the Levenshtein Distance. However, a high percentage of profane words do not in fact constitute abusive content [20]. The typical content-based feature representations in abusive content detection are bag of words (BoW), and ngrams [41]. In addition, Mehdad et al. [24] have shown that using ngrams at the character level is more effective than using ngrams at the word level due to the out-of-vocabulary issue where the words are in the training data and not in the testing data. Apart from these simple surface features, abuse detection can also benefit from other knowledge based features. Xu et al. [38] included part of speech (POS) tagging to improve the classifier accuracy; Dadvar et al. [9] incorporated expert domain knowledge into feature engineering. They proposed a model where the feature space was designed by twelve experts who have a strong background in psychology, communication science and social studies; Yin et al. [39] demonstrated that the baseline result of simply using ngrams features was significantly improved by adding the other information such as contextual features and semantic features; Likewise, Chatzakou et al. [4] used features including user profile information and user network-based information.

In additional to feature representation investigation, many studies have concentrated on the classification algorithms. The widely used traditional classifiers in this domain include Naive Bayes (NB) [7,11], Logistic Regression (LR) [26,37], Support Vector Machines (SVM) [7,8,23,38,39], and Decision Tree (DT) [11]. However, there is no single classifier that generally achieves the best classification performance. Dinakar et al. [11] showed that NB outperformed DT in their experiment; Davidson et al. [10] found that LR and SVM tended to perform significantly better than other classifiers while Dadvar et al. [8] have shown the NB is slightly better than SVM. To avoid overfitting with a single classifier, Burnap [3] proposed an ensemble model which leveraged strengths of different types of classifiers and noted better performance than using a single classifier.

2.2 Deep Learning

Recent research has focused on the use of deep learning to tackle the task of abuse detection. In particular, this trend is sparked by the emergence of embedding techniques such as word2vec [25] and paragraph2vec [22] where each word or paragraph is represented by a vector in a low-dimension vector space. Both word and paragraph vectors are learned using a neural network that predicts context words given the current word, which preserves the syntactic and semantic information. One of the earliest research works on applying this embedding technique in the abuse detection domain is Djuric et al. [12]. They used paragraph2vec [22] to learn the distributed low-dimensional representation for comments that are then used as input to a logistic regression classifier. Serra et al. [35] also proposed a language model to generate a comment vector before inputting to a neural network based classifier. Given that using word2vec/paragraph2vec to represent the input text requires a huge amount of textual content, a lot of research uses pre-trained word embeddings such as W2V [25] by Google and Glove [29] by Stanford for the abuse detection task. Simple approaches to using pre-trained word embeddings for comment representation inlcude averaging [33] and concatenating [42] the word vectors of all words in the comment. Both of these approaches when combined with traditional classifiers resulted in poorer prediction performance to the more complex approaches such as using deep learning classifiers. Currently, convolutional neural network (CNN) and recurrent neural network (RNN) are widely used deep learning neural networks. Incorporating these with pre-trained word embedding representations for the input text, both Gamback [15] and Park et al. [27] have achieved success on the task of abuse detection by applying the CNN model. Gao et al. [16] used Bi-directional Long Short Term Memory (Bi-LSTM), a type of RNN model, to identify abusive comments. They found that this model had better classification results in comparison to logistic regression. Badjatiya et al. [1] carried out extensive experiments using different classifiers for the task of hate speech detection on a Twitter dataset. They found that deep learning models comment embedding generation, with those comments vectors then fed into a decision tree classifier delivered the best results. In addition, Zhang et al. [41] had good classification results with a combination model that extended the basic CNN by adding a RNN layer using gated recurrent unit.

Metadata is also of benefit to deep learning models. Pavlopoulos et al. [28] improved the performance of RNN model by adding the user-based embedding which is a dense vector that represents user profile information; Founta et al. [14] provided a unified deep learning architecture capable of leveraging extra information including sentiment polarity, hashtags existence, and emoticons usage, which increased area under the curve (AUC) by 5%. In addition, Pitsilis et al. [30] proposed an ensemble LSTM classifier that incorporated various features associated with user history information.

3 Datasets

A major barrier to the use of machine learning for identifying abusive user-generated content is the lack of recognised gold-standard labelled research datasets in the domain [34]. Most existing studies are carried out on datasets that are privately collected by the associated researchers. As a result, studies in detection of abusive content suffer from a lack of comparable empirical results against common datasets [34]. To alleviate this issue and generalise our results, we used nine datasets in this paper.

We gathered eight publically available labelled datasets from a variety of social media sites including Twitter [23,38], YouTube [8], MySpace [2,39], Formspring [32], Kongregate [39], and SlashDot [39]. In addition to these eight datasets, we collected our own user-generated abusive content dataset, using comments from a general news platform. We used crowd-sourcing to label the comments. We refer to our total 9 datasets as D1, D2 through to D9 for the rest of paper. The detail of these datasets are presented in our previous research [5,6]. Table 1 summarises the basic properties including the number of instances, average number of words across instances, the class distribution of positive (abusive) instances to negative (non-abusive) instances. We also include information about the approach and results published by the authors of each dataset publication, including the overall results achieved, the measurements used to evaluate, whether oversampling was used to improve the balance of classes in the dataset, the feature representation used and the classifier used.

With the exception of D1 which has a balance of classes, most datasets display class imbalance, with a very small proportion of abusive instances. In particular, the proportion of positive, abusive instances of D5, D6 and D7 is less than 5%. For these datasets, the original authors use oversampling of the positive (abusive) class instances in order to re-balance the class distribution, and thus improve the effectiveness of their classification models.

The previous work associated with these datasets focused on classic machine learning methods. As shown in Table 1, two researchers (D3, D4) used a lexicon matching approach where the text content was predicted as abusive based on whether it contained one of the pre-defined profane words. For D2, knowledge-based features such as users' profile information were manually engineered using domain expertise. The majority of researchers used word ngrams for feature representation and SVM as the classifier. In addition, logistic regression (LR) and rule-based classifiers have also been applied in some cases. In terms of evaluation, there is no standard performance measurement used across these datasets. Overall accuracy (D3, D4) is one of the measurements used for the classification task, which is a drawback when the dataset is imbalanced. Most of the work assessed the classifier using recall (D1, D8, D9), in particular positive recall (D5, D6, D7). AUC is also used in this domain (D2).

Table 1. Summary of dataset

	# of Instances	Avg. Length	Class Dist. (Pos./Neg.)	Oversample	Feature	Classifier	Results by Author	Metrics
D1	3110	15	42/58	No	Ngrams	SVM	0.79	Recall
D2	3466	211	12/88	No	Knowledge	SVM	0.57	AUC
D3	1710	337	23/77	No	Lexical	Rule-Based	0.64	Overall Acc
D4	13153	26	6/94	No	Lexical	Rule-Based	0.82	Overall Acc
D5	4802	5	1/99	Yes	Ngrams	SVM	0.14	Pos. Recall
D6	4303	94	1/99	Yes	Ngrams	SVM	0.12	Pos. Recall
D7	1946	56	3/97	Yes	Ngrams	SVM	0.35	Pos. Recall
D8	1340	13	13/87	No	Ngrams	LR	0.58	Recall
D9	2000	59	21/79	Yes	Ngrams	SVM	0.62	Recall

4 Methodology

In this section, we describe in detail the methods that we used in this work. We start with briefly explaining the data pre-processing. We then discuss the feature representations used, followed by the explanation of two types of classifiers, SVM and two off-the-shelf deep neural networks. We explain our use of oversampling for dataset re-balancing. Finally, we discuss the metrics used to evaluate and compare the classifiers' performance.

Pre-processing. Our first step was to pre-process the data in order to normalize text content. All capital letters in text were replaced by lowercase. The URL links were extracted and replaced by the generic term *url_links*. User names (name followed by the symbol '@') were also replaced by the anonymous term *@username*. Given that user-generated content is typically short, we did not implement dimensionality reduction techniques such as removing stop-words and stemming.

Feature Representation. We used two types of feature representation in this work: traditional text representations and word vectors. A typical traditional representation, ngrams are created by splitting the comment text into n continuous sequential word (or character) occurrences. In our previous work [5], we identified that word ngrams (1–4 word level) was the best performing feature representation. In addition, we applied document frequency reduction, removing the features that occur most and least often.

As an alternative to traditional feature representation, we used word vectors based on pre-trained word embeddings. From individual word vectors, we generated comment vectors, representing a user post. We perform comment embedding in two ways: In our first method, we simply averaged the word vectors for the words that appeared in the comment. We use this approach for comment vectors to input into the SVM classifier. The second method for comment embedding is used when combining word vector input with a deep learning classifier, whereby we feed word vectors into the deep learning model which automatically generates the comment embedding as part of layer determination.

Classifier. We used a support vector machine as a baseline classifier, given that it is a commonly used classifier that is shown to work well for the task of text classification. For our deep learning model comparison, we implemented two popular architectures, convolutional neural networks (CNNs) and recurrent neural networks (RNNs). We adopted the CNN model based on Kim's paper [21] and Bi-LSTM (Bi-directional Long Short Term Memory) structure [18,19] for the RNN model. We used word vectors as input for both models, and a softmax layer as output for predicting the probabilities of two classes (positive and negative). We used categorical cross entropy as the loss function and Adam optimiser to train the model. In addition, the two deep learning neural networks are performed as mini-batch gradient descent where the batch size is 50. As our datasets are not large enough to include a validation set split, we excluded the early stopping technique and set the number of epochs at 50 based on a pilot experiment. As our text content (user posts/comments) varies in length across instances, we used zero-padding in order to make each input the same size, setting this size to be length of the longest comment in the corresponding dataset. In addition, we used fine-tuning in order to update pre-trained word embeddings while modeling the classifier.

The choice of hyper-parameters plays an important role in the accuracy of deep learning models and optimising these parameters is always data-dependent. However, as we are performing our experiments on nine different datasets, we kept the same hyper-parameter settings for all datasets in order to make our results comparable across datasets. We attempted to apply optimal hyper-parameters based on the guidelines by Zhang et al. [40] for CNN and Reimers et al. [31] for RNN respectively. For the CNN model, we used rectified linear unit (ReLU) as the activation function, and multiple filters (the window sizes were 2,3,4) where each filter has 100 feature maps. In addition, we applied dropout during training process (rate is 0.5), and $l2$ regularization for avoiding overfitting. For the RNN model, we implemented one-layer Bi-LSTM and set the size of the hidden layer to 100. The other hyper-parameters are the same as those used with CNN. Finally, Table 2 lists our four end-to-end experimental configurations that we wish to compare: Configuration 1 is the classic feature representation and SVM classifier; Configuration 2 is a hybrid approach using a word vector representation with an SVM classifier. Configurations 3 and 4 are our two deep-learning approaches, using CNN and RNN classifiers with word vectors as input.

Table 2. The proposed learning configurations for the detection of abusive content

Configuration	Feature	Classifier
1	Ngrams	SVM
2	Average word vector	SVM
3	Word vector	CNN
4	Word vector	Bi-LSTM

Oversampling. According to the class distribution in Table 1, most datasets are imbalanced, containing a low proportion of abusive (positive) instances. To address this, we used resampling of positive instances in the training set before training the classifiers. To be specific, we randomly oversampled the minority class instances (abusive instances) in order to increase the class distribution to an appropriate balanced level. The balanced level was decided based on our previous work [5]. To allow for random selection, we oversampled twice and then averaged the results. In addition, given that the parameters of the neural networks were initialized randomly, we also trained our deep learning model twice and averaged the results. Oversampling was performed on the training set only, with test data untouched.

Measurement. We used stratified 10-fold cross validation for our model training. All results are reported using class accuracy metric, also known as recall. This is a standard text classification metric which indicates the ability of the classifier to find all instances of a specific class. We are particularly interested in accuracy over the positive class (abusive instance recall), as we assume that the consequence of failing to identify an abusive comment is more serious than neutral content being classified as abusive. Due to the imbalanced class distributions across the datasets, we also used average class accuracy (average recall) to avoid the scenario that the classifier is skewed by a single class.

5 Experiments and Results

In this section, we explain our experiments and results. As a precursor to comparing classical versus deep learning based approaches, we carried out an experiment to analyse the impact of different word embeddings on deep learning models. Secondly, we present the classification performance of the proposed four configurations from Table 2 on our datasets. Thirdly, we further investigate the capability of these four configurations in tackling the issue of class imbalance. We perform experiments on five extracted datasets with varying levels of class balance with and without using oversampling.

5.1 Word Embedding Experiment

The choice of word embedding to represent text input is a factor to be considered when evaluating the performance of deep learning classifiers. We assume that a word embedding model trained on the same source body of text as the downstream classification dataset will perform better. To validate this assumption, we compared three different word embedding strategies using our own D9 dataset. We use this dataset because we have a large corpora of news comments from the same source as D9 to use to learn the word vectors. Our corpus contains nearly 138 million tokens, as shown in Table 3. Firstly, we trained using the word2vec approach [25]. Once we have finished the training process, each word is represented by a 100 dimension vector, and the size of the word vocabulary is approximately 145,000. We then used our word vectors as input to two deep learning classifiers (CNN and RNN). We repeated the process using two popular pre-trained word embeddings, namely W2V [25] and Glove [29]. Both of these two pre-trained word embeddings are trained on corpora larger than ours, so in theory, giving a richer word vector representation. However, the percentage of overlap words between D9 and the training corpora used for learning the word embeddings is higher for the news corpus where 97% of words in D9 can be identified.

Table 3. Pre-trained word embeddings

	Source	# of Tokens	# of Vocabulary	Dimension	Overlap percentage
Glove	Wikipedia	6B	400K	100	94%
W2V	Google news	100B	3M	300	92%
News comments	News site	138M	145K	100	97%

Although our own word embedding training corpus is the smallest, the subsequent word vectors from this corpus achieve the best abusive recall using both CNN and RNN models as shown in Fig. 1. Compared to the results of using W2V and Glove, the abusive recall of using our own news comment word embeddings is an improvement of more than 20% for the CNN model and nearly 15% for RNN model respectively. Therefore, we suggest that word embeddings created from the same data source as the dataset used to train (and evaluate) the classifier is a practical strategy. For the remainder of our experiments, however, we use published word embedding as we do not have access to the various corpora from which our remaining eight datasets are derived. We apply Glove in the subsequent experiments as Glove achieves slightly better results than the W2V for D9 as noted from in Fig. 1.

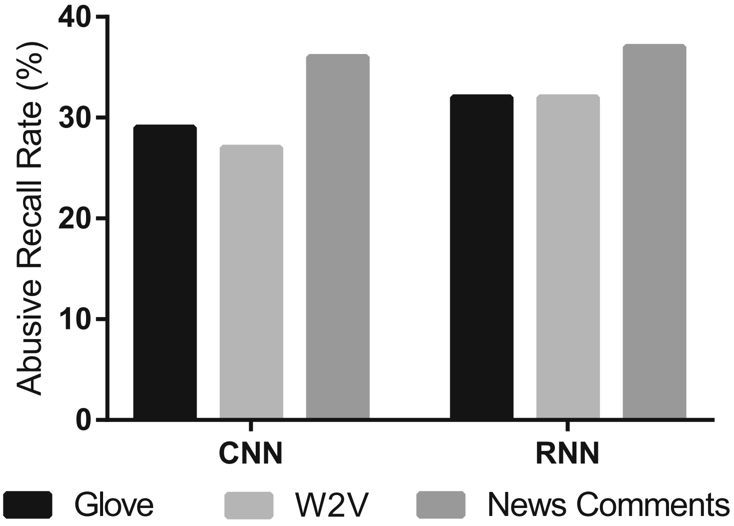

Fig. 1. The performance of deep learning models on the different word embeddings

5.2 Baseline Classification

The aim of this experiment is to assess the performance of the two deep learning models for the detection of abusive user posts and compared them to SVM with ngrams and word vector inputs. Table 4 shows abusive recall (%) and average recall (%) of our four configurations (Table 2) on nine datasets (Table 1). We highlighted the best result per dataset in bold. Generally, the performance of two SVM classifier configurations exceeded the two deep learning models for most datasets (7 out of 9). In particular, SVM with average word vectors achieved the highest abusive recall on 5 datasets. This is surprising given the shallow approach to generating comment vectors through averaging the words vectors for the words in the input. Configuration 1, SVM with ngrams, also has comparative performance, achieving 3 best results. For the deep learning models, CNN performs the best for D1 and D3. However, it performs the worst on the other 6 datasets. The performance of RNN is average in comparison to the SVM and CNN approaches, achieving neither best nor worst results for any dataset.

Table 4. Abusive recall and average recall (in brackets) of 4 classification configurations on 9 datasets. WV is short for word vectors.

	D1	D2	D3	D4	D5	D6	D7	D8	D9
Ngrams+SVM	70(75)	**35(62)**	91(93)	**62(77)**	**58(78)**	12(56)	18(58)	65(78)	33(60)
Avg. WV+SVM	65(71)	30(59)	66(76)	59(74)	**58(76)**	**51(71)**	**48(68)**	**77(85)**	**48(66)**
WV+CNN	**73(73)**	4(51)	**93(95)**	34(66)	57(78)	11(55)	14(57)	59(78)	29(61)
WV+RNN	68(73)	6(51)	81(89)	45(71)	50(75)	14(57)	18(58)	60(77)	32(60)

Overall, deep learning models proved to be less accurate classifiers than the classic SVM classifier. For example in D6, the abusive recall of the CNN model decreased nearly 40% compared to the SVM model with average word vector input. In D2, recall is approximately 30% lower for the RNN model as against the SVM using ngrams. Given that most datasets in this experiment have been oversampled to a relatively balanced level [5], we investigated whether oversampling boosts the performance of the various classifiers to different degrees. According to the summary of datasets in Table 1, we found that deep learning models usually perform worse on the scenario where the class distribution of the original dataset is very imbalanced. For example, the recall rates of the CNN model are 11% and 14% in D6 and D7 respectively where there were less than 3% abusive instances in the datasets. On the other hand, SVM performs worse on the dataset where the original class distribution is more balanced (e.g. D1 and D3). We suggest that these results reveal that the oversampling technique boosts the SVM performance more than the performance of the deep learning models. To investigate this we conducted more experimentation described below.

5.3 Experiments of Balancing and Oversampling

The aim of these experiments were to investigate the impact of class imbalance on deep learning classifiers. The approach taken was as follows: We adjusted the datasets so that their class distribution was close to balanced. We used the class distribution of D1 (42%/58%) in Table 1 as the baseline. We then randomly removed negative instances (under-sampled) on other datasets to reach this baseline class distribution. We had to exclude datasets D5, D6, D7 and D8 as the number of abusive instances in the resulting datasets was too low to conduct 10-fold cross validation. Therefore, five of the datasets were suitable for use in this experiment. A summary of the number of posts per class in the five datasets after undersampling is shown in Table 5.

Table 5. Dataset sizes after undersampling to get to 42%/58% class distribution

	D1	D2	D3	D4	D9
#of Pos.	1303	417	390	836	424
#of Neg.	1807	576	539	1154	586
#of Total	3110	993	929	1990	1010

We wanted to examine the impact of varying levels of class balance on performance of the models. For each dataset, using all non-abusive instances, we added abusive instances in order to measure performance at different levels of class imbalance. To do this, we created 10 different positive percentages per dataset, ranging from 1% to 42% (the whole dataset) in intervals of 5%. We then measured classification accuracy using our previous four configurations (Table 2) for each level of class distribution. In addition, we performed oversampling on each level of class distribution in order to investigate the impact of oversampling on performance. To be specific, at each level of class distribution, we randomly oversampled the abusive instances to reach the positive percentage of the baseline distribution 42%/58%. The results are shown in Fig. 2.

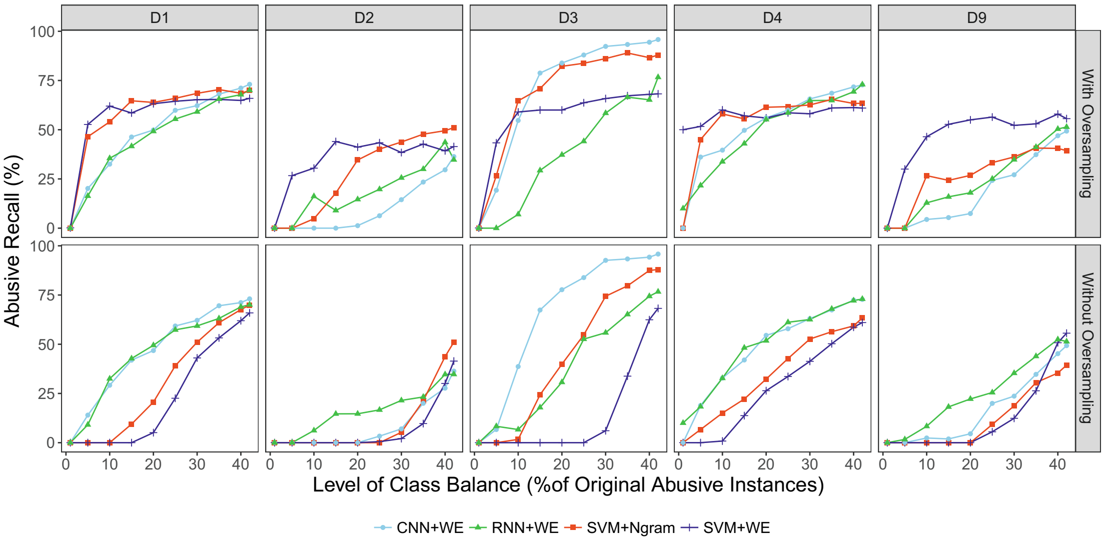

Fig. 2. Abusive Recall for all classifiers across datasets with varying levels of class balance, with and without oversampling. The x-axis represents the original positive percentage in the dataset before oversampling was applied.

Overall, balancing the dataset improves the classifiers' performance due to the increased levels of the positive class in the training data, which is to be expected. However, it is difficult to distinguish the best classifier configuration.

In general, using the original datasets without oversampling, both of the deep learning models outperform the SVM model when the dataset is extremely imbalanced. The SVM classifier with ngrams input cannot detect any abusive comment on D3 when the positive percentage is below 20%. It shows even worse abuse detection ability in D2 where abusive recall remains at zero until the positive class proportion in the dataset is increased above 30%. On the contrary, deep learning models achieve superior results when the dataset is highly skewed by the negative instances.

For example, at 5% positive proportion in D1 and D3, abusive recall is raised in both CNN and RNN models but two SVM model configurations have no ability to detect abusive comments. However, once oversampling is used, the results are quite different. The performance of both SVM configurations is rapidly boosted and outperforms both deep learning models at the low level of class balance. In addition, we note that the SVM configurations tend to saturate earlier than the deep learning models when re-balancing the dataset. For example, there is hardly any improvement after 20% for SVM (ngrams) for all datasets. However, the performance of deep learning models is increased at a close to linear rate as levels of class balance increase.

Depending solely on abusive recall to evaluate the performance of a classifier provides an incomplete picture. Increases in the proportion of the instances that are positive may sacrifice the negative class recall (i.e. the proportion of non-abusive instances correctly predicted as non-abusive). Therefore, we also investigated the average recall as shown in Fig. 3. Average recall starts approximately at 50% where the positive recall is around 0% and negative recall is close to 100% when the dataset is extremely imbalanced. Similar to the abusive recall, average recall increases as class balance increases, both with and without oversampling.

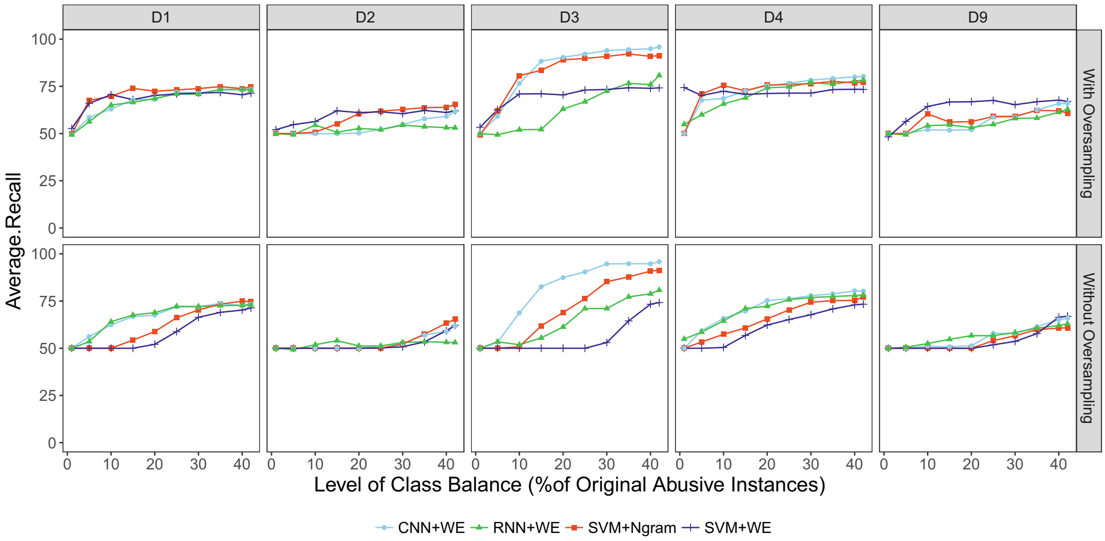

Fig. 3. Average Recall for all classifiers across datasets with varying levels of class balance, with and without oversampling. The x-axis represents the original positive percentage in the dataset before oversampling was applied.

To analyse the impact of oversampling on different classifier configurations, we re-organized our results and displayed it as Fig. 4 which compares with and without oversampling for each classifier across each dataset. It is interesting to note that the influence of oversampling for both deep learning models is limited, as shown in the top two rows in Fig. 4. In particular for the RNN model, there is barely any difference between two results with and without oversampling. Although the ability of the CNN model to detect abuse is boosted by oversampling, the gain is not comparable to the benefit that oversampling brings to the SVM models. Among the four classification configurations, the abusive recall of using SVM with average word embeddings is increased dramatically when oversampling is performed.

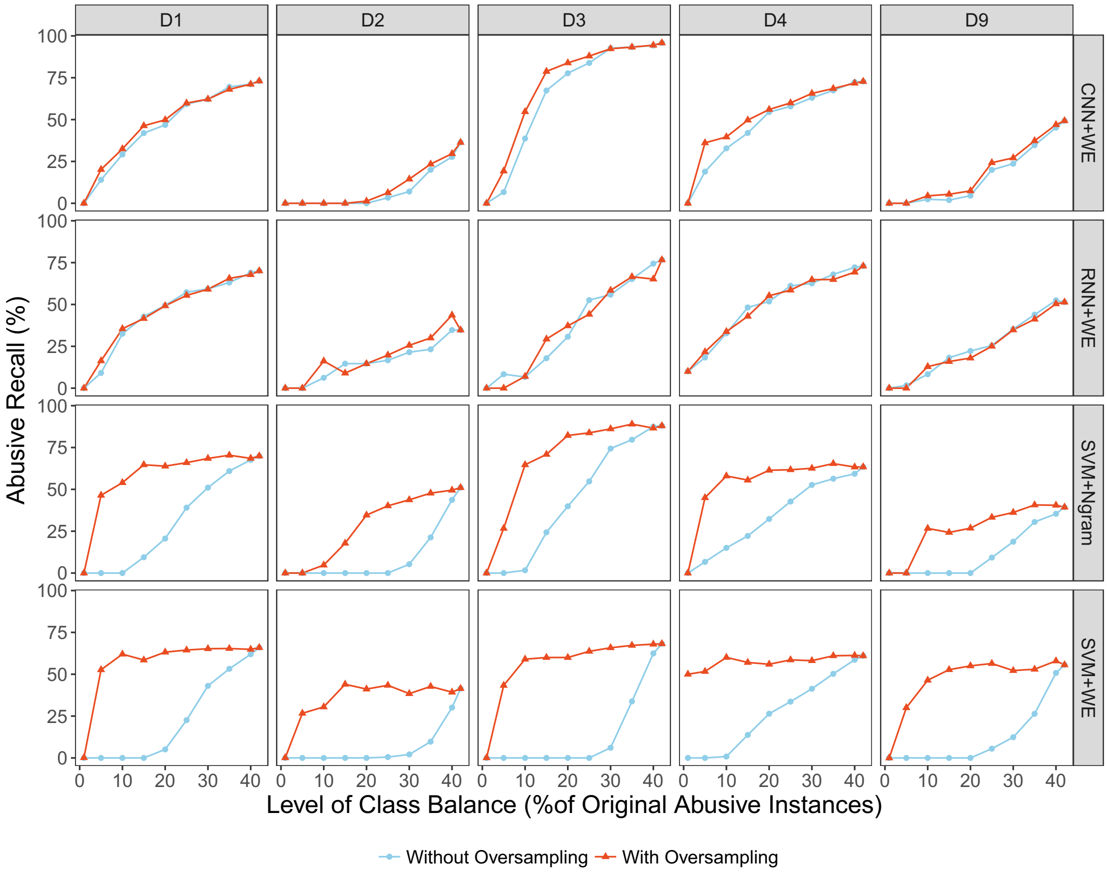

Fig. 4. Abusive Recall with and without oversampling for each classifier on each dataset across varying levels of class balance. The x-axis represents the original positive percentage in the dataset before oversampling was applied.

6 Conclusion

The purpose of our work was to explore the automatic detection of abusive content using a variety of supervised machine learning techniques. In particular we aimed to compare the more traditional approaches against the more recent neural network based approaches. We compared the following classification models, SVM classifier and two deep neural based classifiers: CNN and RNN. We also compared ngrams versus word embeddings for feature representation. We highlight the following points from this work: (1) Using word embeddings which were pre-trained on the same data source as the subsequent task is a benefit to the abuse detection task; (2) Based on results across nine datasets, we showed the SVM classifiers achieved the best results on balanced datasets, with balance achieved through oversampling; (3) We conducted a comprehensive analysis of the ability of the different classifiers to deal with class imbalance. The results show that deep learning models performed well on extremely imbalanced datasets while SVM had no ability to identify the minority abusive content class; (4) Once we applied oversampling techniques to re-balance the dataset, we revealed that oversampling is an effective approach to improve SVM performance while the improvement for deep learning based models is limited.

In future, we would like to investigate in-depth whether the classification results would be influenced by the intrinsic characteristics of individual datasets and sources such as the size of dataset, the average word length etc. Moreover, given the imbalanced nature of the data in the task of detecting abusive content, our future work will aim to modify the current deep learning models in order to improve abusive text detection when the training dataset is imbalanced.

References

1. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759–760. International World Wide Web Conferences Steering Committee (2017)
2. Bayzick, J., Kontostathis, A., Edwards, L.: Detecting the presence of cyberbullying using computer software. In: 3rd Annual ACM Web Science Conference (WebSci 11), pp. 1–2 (2011)
3. Burnap, P., Williams, M.L: Cyber hate speech on Twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet **7**(2), 223–242 (2015)
4. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., Vakali, A.: Mean birds: detecting aggression and bullying on Twitter. In: Proceedings of the 2017 ACM on Web Science Conference, pp. 13–22. ACM (2017)
5. Chen, H., Mckeever, S., Delany, S.J.: Harnessing the power of text mining for the detection of abusive content in social media. In: Angelov, P., Gegov, A., Jayne, C., Shen, Q. (eds.) Advances in Computational Intelligence Systems. AISC, vol. 513, pp. 187–205. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-46562-3_12
6. Chen, H., Mckeever, S., Delany, S.J.: Presenting a labelled dataset for real-time detection of abusive user posts. In: Proceedings of the International Conference on Web Intelligence, pp. 884–890. ACM (2017)
7. Chen, Y., Zhou, Y., Zhu, S., Xu, H.: 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), pp. 71–80. IEEE (2012)
8. Dadvar, M., Trieschnigg, D., de Jong, F.: Experts and machines against bullies: a hybrid approach to detect cyberbullies. In: Sokolova, M., van Beek, P. (eds.) AI 2014. LNCS, vol. 8436, pp. 275–281. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06483-3_25
9. Dadvar, M., Trieschnigg, R.B., de Jong, F.M.G.: Expert knowledge for automatic detection of bullies in social networks. In: 25th Benelux Conference on Artificial Intelligence, BNAIC 2013, TU Delft (2013)
10. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. arXiv preprint arXiv:1703.04009 (2017)
11. Dinakar, K., Reichart, R., Lieberman, H.: Modeling the detection of textual cyberbullying. Soc. Mob. Web **11**(02), 11–17 (2011)

12. Djuric, N., Zhou, J., Morris, R., Grbovic, M., Radosavljevic, V., Bhamidipati, N.: Hate speech detection with comment embeddings. In: Proceedings of the 24th International Conference on World Wide Web, pp. 29–30. ACM (2015)
13. dos Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 69–78 (2014)
14. Founta, A.-M., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., Leontiadis, I.: A unified deep learning architecture for abuse detection. arXiv preprint arXiv:1802.00385 (2018)
15. Gambäck, B., Sikdar, U.K.: Using convolutional neural networks to classify hate-speech. In: Proceedings of the First Workshop on Abusive Language Online, pp. 85–90 (2017)
16. Gao, L., Huang, R.: Detecting online hate speech using context aware models. arXiv preprint arXiv:1710.07395 (2017)
17. Gao, L., Kuppersmith, A., Huang, R.: Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach. arXiv preprint arXiv:1710.07394 (2017)
18. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
20. Hosseinmardi, H., Mattson, S.A., Rafiq, R.I., Han, R., Lv, Q., Mishra, S.: Detection of cyberbullying incidents on the instagram social network. arXiv preprint arXiv:1503.03909 (2015)
21. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
22. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
23. Mangaonkar, A., Hayrapetian, A., Raje, R.: Collaborative detection of cyberbullying behavior in Twitter data. In: 2015 IEEE International Conference on Electro/Information Technology (EIT), pp. 611–616. IEEE (2015)
24. Mehdad, Y., Tetreault, J.: Do characters abuse more than words? In: Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 299–303 (2016)
25. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
26. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y.: Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, pp. 145–153. International World Wide Web Conferences Steering Committee (2016)
27. Park, J.H., Fung, P.: One-step and two-step classification for abusive language detection on Twitter. arXiv preprint arXiv:1706.01206 (2017)
28. Pavlopoulos, J., Malakasiotis, P., Bakagianni, J., Androutsopoulos, I.: Improved abusive comment moderation with user embeddings. arXiv preprint arXiv:1708.03699 (2017)
29. Pennington, J., Socher, R., Manning, C.D., Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
30. Pitsilis, G.K., Ramampiaro, H., Langseth, H.: Detecting offensive language in tweets using deep learning. arXiv preprint arXiv:1801.04433 (2018)

31. Reimers, N., Gurevych, I.: Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799 (2017)
32. Reynolds, K., Kontostathis, A., Edwards, L.: Using machine learning to detect cyberbullying. In: 2011 10th International Conference on Machine Learning and Applications and Workshops (ICMLA), vol. 2, pp. 241–244. IEEE (2011)
33. Sax, S.: Flame wars: automatic insult detection (2016)
34. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pp. 1–10 (2017)
35. Serra, J., Leontiadis, I., Spathis, D., Blackburn, J., Stringhini, G., Vakali, A.: Class-based prediction errors to detect hate speech with out-of-vocabulary words. In: Abusive Language Workshop, vol. 1. Abusive Language Workshop (2017)
36. Sood, S., Antin, J., Churchill, E.: Profanity use in online communities. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1481–1490. ACM (2012)
37. Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C.: Detecting offensive tweets via topical feature discovery over a large scale Twitter corpus. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1980–1984. ACM (2012)
38. Xu, J.-M., Jun, K.-S., Zhu, X., Bellmore, A.: 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, pp. 656–666. Association for Computational Linguistics (2012)
39. Yin, D., Xue, Z., Hong, L., Davison, B.D., Kontostathis, A., Edwards, L.: Detection of harassment on web 2.0. In: Proceedings of the Content Analysis in the WEB, vol. 2, pp. 1–7 (2009)
40. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)
41. Zhang, Z., Luo, L.: Hate speech detection: a solved problem? The challenging case of long tail on Twitter. arXiv preprint arXiv:1803.03662 (2018)
42. Zhong, H., et al.: Content-driven detection of cyberbullying on the instagram social network. In: IJCAI, pp. 3952–3958 (2016)