Using Convolutional Neural Networks to Classify Hate-Speech

Björn Gambäck and **Utpal Kumar Sikdar**
Department of Computer Science
Norwegian University of Science and Technology
NO–7491 Trondheim, Norway
gamback@ntnu.no utpal.sikdar@gmail.com

Abstract

The paper introduces a deep learning-based Twitter hate-speech text classification system. The classifier assigns each tweet to one of four predefined categories: racism, sexism, both (racism and sexism) and non-hate-speech. Four Convolutional Neural Network models were trained on resp. character 4-grams, word vectors based on semantic information built using word2vec, randomly generated word vectors, and word vectors combined with character n-grams. The feature set was down-sized in the networks by max-pooling, and a softmax function used to classify tweets. Tested by 10-fold cross-validation, the model based on word2vec embeddings performed best, with higher precision than recall, and a 78.3% F-score.

1 Introduction

During the Spring of 2017, parliamentary committees in Germany and the UK strongly criticised leading social media sites such as Facebook, Twitter and Youtube (Google) for failing to take sufficient and quick enough action against hate-speech, with the German government threatening to fine the social networks up to 50 million euros per year if they continue to fail to act on hateful postings (and posters) within a week (Thomasson, 2017).

When called to witness in front of the UK Home Affairs Committee, all the social media companies refused to reveal both the number of people they employ to battle hate-speech and the amount they spend on this. However, Google claimed to have invested "hundreds of millions" while Facebook stated that they had thousands of people working on the problem. The German government estimated that the companies combined already spend some 50 million euros per year and that the suggested new German law would increase that amount by 50% (CDU/CSU & SPD, 2017, p.14).

Regardless of the resources actually devoted by the social media networks, it is clear that their current efforts are not enough: "we are disappointed at the pace of development of technological solutions" (Home Affairs Committee, 2017, p.24). The UK and German governments also indicate that they are moving in the direction of treating online content providers in analogy with publishers of printed material, with the same obligations to abide to publishing laws.

With legislation in other countries set to follow (Nielsen, 2017), properly identifying hate-speech is a pressing issue, not only for the major players, but also for smaller companies, clubs, and organisations that allow for user-generated content on their sites (albeit the current German law proposal makes an exception for sites with less than 2 million users). Many such sites currently use slow, manual moderation, which mean that abusive posts will be left online for too long without appropriate action being taken or that content will be published with delay (which might be unacceptable to the users, e.g., in online chat rooms).

Following the work by Collobert et al. (2011), deep neural networks have been shown to effectively solve several language processing tasks such as part-of-speech tagging, sentiment analysis, and named entity recognition. Here a Convolutional Neural Network (CNN) model with various features is utilised for hate-speech categorisation. Word vectors based on semantic information are built for all tokens using an unsupervised learning algorithm, word2vec. The word vectors are merged with a set of extracted features, down-sized using max-pooling, and together with character n-grams (4-grams) fed to the neural network model to predict the categories of each tweet.

Proceedings of the First Workshop on Abusive Language Online, pages 85–90,
Vancouver, Canada, July 30 - August 4, 2017.

The paper is organised as follows: Previous work on hate-speech identification is discussed in Section 2. Section 3 describes the deep learning-based hate-speech categorisation strategy, while experiments and results are reported in Section 4. Finally, Section 5 summarises the discussion.

2 Related Work

Although the above-noted law-maker interest in the issue is fairly recent, the task of identifying hate speech and abusive language in online content has already been topical in the research community for 20 years. Spertus (1997) built the decision tree-based classifier 'Smokey' which utilised 47 syntactic and semantic sentential features. When trained on a small set of 720 web page posts manually annotated (as "flame", "okay" or "maybe") and evaluated on 502 other messages, 'Smokey' performed well on classifying the non-inflamatory messages, but fell completely short on flame texts (thus obtaining an accuracy of only 88.2% on a task with a majority-class baseline of 86.1%).

Addressing the dataset size problem, Sood et al. (2012) collected 1.6 million comments from a Yahoo! social news site, of which 6,500 were randomly selected for annotation by 221 persons on Amazon Mechanical Turk (AMT). Several Support Vector Machine classifiers were trained on varying-size parts of this dataset using mainly word n-gram features, indicating that classification performance kept improving with increased datasets, but not as rapidly after the data size had passed 1,500 items. Looking at another set of AMT-annotated Yahoo! news posts, Nobata et al. (2016) experimented with several different word-internal, n-gram-based, syntactic, and distributional semantic features, concluding that character n-grams alone contribute sufficiently strongly for an online gradient descent learner to perform well on this type of data.

Moving away from features based solely on the language used in online messages, Chen et al. (2012) proposed a model also taking into account the posting patterns of the users in order to single out persons exhibiting abusive behaviour. Similarly, Buckels et al. (2014) aimed to extract traits from online user behaviour that would indicate antisocial personality. This is of particular importance for swift moderation of online chat rooms, as addressed by, e.g., Yin et al. (2009) and Papegnies et al. (2017), with the latter suggesting several types of features (at the morphological, syntactic and user behaviour levels) that can be used for identifying when gamers on a French MMO (massively multiplayer online) game site move from discussing game-related issues to posting personal inflammatory remarks.

Of particular relevance to the present work are previous efforts on identifying abusive language on Twitter. Xiang et al. (2012) created offensive-language topic clusters using Logistic Regression over a set of 860,071 tweets automatically annotated using a boot-strapping technique and supplemented with a dictionary of 339 offensive words. When tested on 4,029 randomly selected tweets collected just after the training set, the lexicon-enhanced clustering outperformed a keyword matching baseline. Logistic Regression and a dictionary was also utilised by Davidson et al. (2017); however, they used crowd-sourcing to create their hate-speech dictionary and aimed to separate the tweets into three classes: hate-speech, offensive language, and neither. Working on a set of 24,802 manually labelled tweets, they achieved good recall and precision overall, but noted that almost 40% of the actual hate-speech tweets were misclassified, although with 3/4 of those being mistaken for offensive language only.

A recurring problem with several of these experiments has been that the annotated datasets have not always been made publically available. However, Ross et al. (2016) had a set of 541 German tweets annotated, in particular addressing the issues of annotator and annotation reliability, and what information should be provided to the annotators. Waseem (2016) discusses similar issues while providing a set of 6,909 English tweets hate-speech annotated by CrowdFlower users,[1] and extending a previous such dataset (Waseem and Hovy, 2016). This dataset will be used in the experiments reported below.

Wulczyn et al. (2016) also used CrowdFlower to obtain human annotations of 115,737 comments on Wikipedia as to whether they contained personal attacks and harassment. They furthermore experimented with strategies to automatically expand the dataset, comparing Multi-Layer Perceptrons (a single-hidden-layer neural network) to Logistic Regression, and word n-grams to character n-grams; concluding the Logistic Regression with character n-grams performed best.

[1] https://www.crowdflower.com/

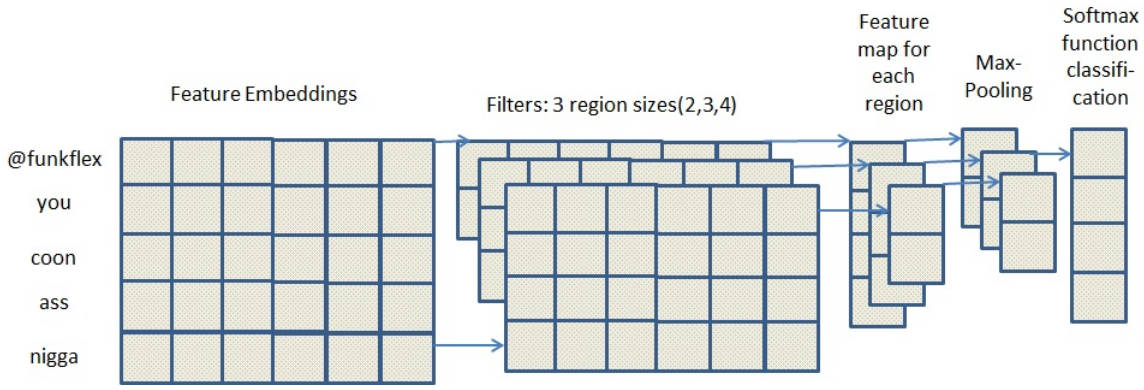

Figure 1: Hate-speech classifier

3 CNN-based Hate-Speech Classification

This section describes the hate-speech identification system architecture based on Convolutional Neural Networks (CNN). An overview of the system is shown in Figure 1. The first step of the system is to generate feature embeddings. Feature embeddings for all words were constructed by using word embeddings and character n-grams.

The word embeddings were generated in two ways, through word2vec (Mikolov et al., 2013a,b) and through random vectors. In the random vector setting, all the words in the corpora are initialised with random values. In the word2vec version, word vectors are generated based on the context. There are two types of such embeddings: continuous-bags-of-words (CBOW) and skip-gram models. In the CBOW architecture, the model predicts the current word from a window of surrounding context words. In the skip-gram model, the context words are predicted using the current word.

In addition to the word embeddings, length 28 one-hot character n-gram vectors were generated, with 26 elements for the English alphabet, one for digits, and one for all other characters/symbols. The feature embeddings were produced by concatenating the word embeddings with these character n-gram vectors.

A pooling layer in the network converts each tweet into a fixed length vector, capturing the information from the entire tweet. A max-pooling layer then captures the most important latent semantic factors from the tweets.

On the output side, a softmax layer calculates the class probability distributions for each tweet and assigns the hate-speech classes / labels based on the probability values.

4 Experiments

Four approaches to hate-speech classification were tested, based on different feature embeddings. All models were applied to the English Twitter hate-speech dataset created by Waseem (2016).[2] Each tweet in the dataset has been annotated by one Expert annotator and three Amateur annotators, with four labels: non-hate-speech (84% of the data), racism, sexism, and both (i.e., racism *and* sexism).

Waseem (2016) defined the "Expert" annotators as those having both a theoretical and applied knowledge of hate speech (those were recruited among feminist and antiracism activists), while the "Amateur" annotations were obtained by crowd-sourcing (on the CrowdFlower platform). We combined the annotated tags for each tweet based on majority voting, where the Expert was given double unit votes and each of the Amateurs was given a single unit vote.

The class distributions of the dataset are shown in Table 1. The total size of the dataset (6,655 tweets) is slightly lower than the original set (Waseem reported it as containing 6,909 tweets), since some of the annotated tweets were unavailable or had been deleted.

Data	Number of tweets
Racism	91
Sexism	946
Both (racism & sexism)	18
Non-hate-speech	5600
Total	6655

Table 1: Twitter hate-speech dataset statistics

[2] http://github.com/zeerakw/hatespeech

	System setup	Precision	Recall	F_1-score
CNN	Random vectors	**0.8668**	0.6726	0.7563
	word2vec	0.8566	0.7214	**0.7829**
	Character n-grams	0.8557	0.7011	0.7695
	word2vec + character n-grams	0.8661	0.7042	0.7738
Logistic Regression with character n-grams (Waseem and Hovy, 2016)		0.7287	**0.7775**	0.7389

Table 2: System performance (10-fold cross-validated)

4.1 Results

The average 10-fold cross-validated results for all four Convolutional Neural Network (CNN) models are shown in Table 2, and compared to the Logistic Regression (LogReg) model used by Waseem and Hovy (2016).

In the first CNN model, random word vectors were considered as feature embeddings when training the network. This baseline model achieved precision, recall and F-score values of 86.68%, 67.26% and 75.63%, respectively, marking a drastic improvement in precision compared to the LogReg model, but at the expense of lower recall. In the second approach, word2vec word vectors were taken as feature embeddings to learn the CNN model, resulting in clearly (7.3%) improved recall, for an F-score of 78.29%, even though the precision actually was slightly reduced compared to using the random vectors.

The third and fourth models both added character n-grams to the input of the CNN model. In line with the experiments reported on the same dataset by Waseem and Hovy (2016), length 4 character n-grams were used. In the third model, only the character n-gram were considered as feature embeddings when training the CNN model, while in the fourth model, the feature embeddings were generated by concatenating word2vec word embeddings and character n-grams. Tested by 10-fold cross-validation, the latter system showed better precision (86.61%) than recall (70.42%), for an F-score of 77.38%.

However, although the character n-grams thus helped a little in improving precision, the word2vec model without character n-grams still achieved the best results of all the compared models, with the precision, recall and F-score values of 85.66%, 72.14% and 78.29%, respectively. Note that all CNN models convincingly outperformed Logistic Regression in terms of both precision and F_1-score, while the LogReg model achieved better recall than all the neural network models.

4.2 Error Analysis

An error analysis was carried out for each of the 10 folds. The confusion matrices are shown in Table 3. It can be observed that the model overall did not identify many tweets as hate-speech tweets. This may be due to insufficient training instances. Furthermore, the system wrongly identified some non-hate-speech tweets as hate-speech.

In particular, the system was not able to identify properly the category 'both', since the examples of this category are very few (1 or 2 per fold) with respect to the whole set of training instances. The system performed better in the 'sexism' category than in the other hate-speech categories ('both' and 'racism') because the number of tweets of this category are larger.

5 Conclusion and Future Work

Here we have experimented with a system for Twitter hate-speech text classification based on a deep-learning, Convolutional Neural Network model. The classifier assigns each tweet to one of four predefined categories: racism, sexism, both (racism and sexism) and neither.

Two CNN models were created based on different input vectors sets that were fed to the neural networks for training and classification. Word vectors based on semantic information were built using an unsupervised strategy, word2vec, and compared to a randomly generated vector baseline. In additional, two CNN models were trained on character 4-grams, as well as on a combination of word vectors and character n-grams. The feature set is down-sized in the networks by a max-pooling layer, while a softmax layer is utilised to assign the tweets their most probable label category.

True \ CNN	Fold-1				Fold-2				Fold-3				Fold-4				Fold-5			
	b	s	r	n	b	s	r	n	b	s	r	n	b	s	r	n	b	s	r	n
both	0	0	0	1	0	2	0	0	0	0	0	1	1	1	0	0	0	1	0	0
sexism	1	70	0	25	0	71	0	20	0	78	0	19	0	82	0	18	0	69	0	23
racism	0	0	5	6	0	0	0	6	0	0	2	8	0	0	5	7	0	0	1	8
neither	0	13	1	543	0	15	4	547	0	11	2	544	0	11	3	537	0	13	0	550
	Fold-6				**Fold-7**				**Fold-8**				**Fold-9**				**Fold-10**			
both	1	1	0	0	0	1	0	0	0	1	0	0	0	2	0	1	1	3	0	0
sexism	0	66	0	17	0	72	0	18	0	80	0	33	0	70	0	26	0	70	0	18
racism	0	0	3	1	0	0	1	3	0	0	6	9	0	0	1	9	0	0	6	4
neither	0	9	4	563	0	10	0	560	0	7	2	527	0	16	0	540	0	6	0	562

Table 3: Confusion matrices for each fold, with rows showing the true labels and columns system outputs. Legend: 'b' = both, 's' = sexism, 'r' = racism and 'n' = neither.

Trained and tested by 10-fold cross-validation, the system based on word2vec word vectors performed best overall, with an F_1-score of 78.3%. Adding character n-grams slightly increased the precision, but resulted in lower recall and F-score.

The tested models and neural network architectures could be extended in several ways: The word2vec embeddings used here were built on skip-grams that predict the context words using the current word. An alternative would be to use continuous-bags-of-words that basically do the opposite and predict the current word from a window of surrounding context words. Also, following Waseem and Hovy (2016) only length 4 character n-grams were used. Clearly it would be interesting to explore whether these are uniformly ineffective when changing the n-gram size.

The experiments reported here were carried out on a convolutional network architecture, but other types of deep neural networks could obviously be tried. In particular, the bi-directional Long Short-Term Memory (LSTM) recurrent neural network architecture has shown itself to be useful to language processing problems where utilising the sequential nature of the input is more essential, such as named entity recognition and sentiment analysis, although most of the best performing systems in SemEval 2016 (the International Workshop on Semantic Evaluation; Task 4: Sentiment Analysis in Twitter) actually utilised convolutional neural networks or combinations of CNNs and other approaches (Nakov et al., 2016).

A long those lines, Sikdar and Gambäck (2017) report experiments with a set-up for named entity recognition combining an LSTM with a more traditional machine learning classifier based on Conditional Random Fields (CRF). Such an approach could be tested also for the abusive language classification task, either using the LSTM/CRF combination or including CNN.

Acknowledgments

The work reported here was carried out within the CZ09 Czech-Norwegian Research Programme under Project Contract 7F14047, HaBiT ("Harvesting big text data for under-resourced languages"; http://www.habit-project.eu) funded by the Research Council of Norway (NFR) and the Czech Republic's Ministry of Education, Youth and Sports (MŠMT) through the EEA/Norway Financial Mechanism.

Thanks to the four anonymous reviewers for comments that helped improve the paper.

References

Erin E. Buckels, Paul D. Trapnell, and Delroy L. Paulhus. 2014. Trolls just want to have fun. *Personality and Individual Differences* 67:97–102.

CDU/CSU & SPD. 2017. Gesetzentwurf der Fraktionen der CDU/CSU und SPD: Entwurf eines Gesetzes zur Verbesserung der Rechtsdurchsetzung in sozialen Netzwerken (Netzwerkdurchsetzungsgesetz — NetzDG). Drs. 18/12356, Deutscher Bundestag, Berlin, Germany.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*. IEEE Computer Society, Amsterdam, The Netherlands, pages 71–80.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from

scratch. *Journal of Machine Learning Research* 12:2493–2537.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*. American Association for Artificial Intelligence, Toronto, Canada. To appear.

Home Affairs Committee. 2017. Hate crime: abuse, hate and extremism online. Fourteenth Report of Session 2016–17 HC 609, House of Commons, London, UK.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*. Curran Associates, Red Hook, NY, USA, pages 3111–3119.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 Task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*. ACL, San Diego, California.

Nikolaj Nielsen. 2017. EU states back bill against online hate speech. EUobserver, May 24. https://euobserver.com/justice/138009.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Montreal, Canada, pages 145–153.

Etienne Papegnies, Vincent Labatut, Richard Dufour, and Georges Linarès. 2017. Impact of content features for automatic online abuse detection. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: Proceedings of the 18th International Conference*. Springer, Budapest, Hungary.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2016. Measuring the reliability of hate speech annotations: The case of the European refugee crisis. In *3rd Workshop on Natural Language Processing for Computer-Mediated Communication*. Bochum, Germany, pages 6–9.

Utpal Kumar Sikdar and Björn Gambäck. 2017. Named entity recognition for Amharic using stack-based deep learning. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: Proceedings of the 18th International Conference*. Springer, Budapest, Hungary.

Sara Owsley Sood, Elizabeth F. Churchill, and Judd Antin. 2012. Automatic identification of personal insults on social news sites. *Journal of the American Society for Information Science and Technology* 63(2):270–285.

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence*. American Association for Artificial Intelligence, Providence, Rhode Island, pages 1058–1065.

Emma Thomasson. 2017. German cabinet agrees to fine social media over hate speech. Reuters, Apr 5. http://uk.reuters.com/article/idUKKBN1771FK.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? Annotator influence on hate speech detection on Twitter. In *Proceedings of 2016 EMNLP Workshop on Natural Language Processing and Computational Social Science*. ACL, Austin, Texas, pages 138–142.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, San Diego, California, pages 88–93.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2016. Ex Machina: Personal attacks seen at scale. *CoRR* abs/1610.08914.

Guang Xiang, Bin Fan, Ling Wang, Jason I. Hong, and Carolyn P. Rose. 2012. 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*. ACM, Maui, Hawaii, pages 1980–1984.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian D. Davison, April Kontostathis, and Lynne Edwards. 2009. Detection of harassment on Web 2.0. In *Proceedings of the Content Analysis in the WEB 2.0 Workshop at WWW2009*. Madrid, Spain.