Filtering Aggression from the Multilingual Social Media Feed

Sandip Modha
DA-IICT, Gandhinagar
sjmodha@gmail.com

Prasenjit Majumder
DA-IICT, Gandhinagar
prasenjit.majumder@gmail.com

Thomas Mandl
University of Hildesheim, Hildesheim
mandl@uni-hildesheim.de

Abstract

This paper describes the participation of team DA-LD-Hildesheim from the Information Retrieval Lab(IRLAB) at DA-IICT Gandhinagar, India in collaboration with the University of Hildesheim, Germany and LDRP-ITR, Gandhinagar, India in a shared task on Aggression Identification workshop in COLING 2018. The objective of the shared task is to identify the level of aggression from the User-Generated contents within Social media written in English, Devnagiri Hindi and Romanized Hindi. Aggression levels are categorized into three predefined classes namely: 'Overtly Aggressive', 'Covertly Aggressive' and 'Non-aggressive'. The participating teams are required to develop a multi-class classifier which classifies User-generated content into these pre-defined classes. Instead of relying on a bag-of-words model, we have used pre-trained vectors for word embedding. We have performed experiments with standard machine learning classifiers. In addition, we have developed various deep learning models for the multi-class classification problem. Using the validation data, we found that validation accuracy of our deep learning models outperform all standard machine learning classifiers and voting based ensemble techniques and results on test data support these findings. We have also found that hyper-parameters of the deep neural network are the keys to improve the results.

1 Introduction

Social media become the popular platform for the common man to the celebrities to discuss or to give their opinions about any real-world events. In the last a few years, the growth of social media users was enormous. With this large user base, a massive amount of User-generated content is posted continuously on Social Media. Social media gives freedom of speech and anonymity to its users. However, often Social media users abuse this liberty to spread abuses and hate through the posts or comments. On many occasions, these User-generated contents are offensive or actively aggressive in nature. Many times, such contents written in a way that might defame or insult individuals or groups of people without actually using any explicit hate-related or abusive words. Genuine social media users may become the victim of such abusive or hate comments. Recently, many cases of suicide have been reported by mainstream media due to trolling or cyberbullying in social media.

Day by day, Anti-Social behavior like Abuse, trolling and cyberbullying is becoming more common than before on the Social media platform. It is high time for researcher, industry to develop a system which identifies problematic posts. Social media posts might contain words which can be considered as either highly or open aggressive or have hidden aggression. Sometimes posts do not have any aggression. Based on these, posts or comments are classified into three classes namely: 'Overtly Aggressive', 'Covertly Aggressive' and 'Non-aggressive' by the track organizers (Kumar et al., 2018a). Henceforth, in rest of paper, we will denote these classes by these abbreviations namely: OAG, CAG, NAG respectively. Table 1 shows the sample posts belonging to these classes.

Our approach for the shared task TRAC (Kumar et al., 2018a) is based on Machine Learning and Deep learning. Track organizers have provided training and validation data as per the schedule. Initially, we have implemented all standard machine learning classifier along with voting based ensemble technique and prepare baseline results on validation data. Thereafter, we have started to develop deep

Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, pages 199–207
Santa Fe, USA, August 25, 2018.

Id	Text	Class-label
1	Do you see further downside in bank nifty till expiry	NAG
2	Demonitisation like a medicine, it might be sour but slowly and steadily	CAG
3	Woman shame on u	OAG

Table 1: sample post for each class

learning model for the multi-class classification. Since Dataset was created from Facebook posts and comments, we have chosen fastText pre-trained vector for the word-embedding. Our deep learning models were based on Bidirectional LSTM, Convolution neural network. In section 3.2 we will discuss our approaches. It is important to note that we did not engage in any specific pre-processing on the text before they were fed into the deep neural net.

Results declared by track organizers show that our deep learning models perform top in the Social media Hindi dataset, third in English Facebook dataset, fifth in Facebook Hindi dataset. So, overall our models perform fairly well across all the datasets.

The rest of this paper is organized as follows. In section 2 we briefly discuss related work in this area. In section 3, we present the corpus statistics and methodology. In section 4, we present results and give the brief analysis. In section 5, we will give our final conclusions along with future works.

2 Related Work

Hate speech and sentiment analysis are the well studied are in the field of Natural Language Processing. Aggression identification shared task (Kumar et al., 2018a) is more specific than hate speech and sentiment analysis task. In (Xu et al., 2012) authors introduce cyberbullying to the NLP community. They have performed various binary classification on tweets text with bullying perspective to determine whether the user is cyberbully or not, from the sentiment perspective. They reported binary classification accuracy around 81%.

In (Kwok and Wang, 2013), authors have tried to classify tweets against black or not. They have collected two classes (racist and non-racists) of tweets and used the Naive Bayes classifier for binary classification. The average accuracy for the binary classification was around 76 %. (Djuric et al., 2015) also build a binary classifier to classify in between hate speech and clean user comments on a website. Authors have proposed to learn distributed low-dimensional representations of comments using word embedding model such as paragraph2vec model. Authors have created corpus comprises 56,280 comments containing hate speech and 895456 clean comments from Yahoo finance. They have reported AUC score around 0.8007 using CBOW paragraph2vec model against 0.7889 in the bag-of-words model using term frequency.

In (Burnap and Williams, 2015), authors explore cyber hate on Twitter. They have collected tweets for the specific domain in a two-week time window. Collection of 450,000 tweets was annotated as hateful or genuine. They have performed binary classification using SVM, BLR, RFDT, Voting base ensemble and hey achieved best F1-score of 0.77 in the voted ensemble. In (Malmasi and Zampieri, 2017), authors have used NLP based lexical approach to address the multi-class classification problem. They have used character N-gram, word N-gram and word skip-gram feature for the classification.

In (Schmidt and Wiegand, 2017), authors have described the key areas that have been explored to detect hate speech. They have surveyed different types of features used for hate speech classification. They have categorized features in Simple surface features, word generalization features, sentiment features, linguistic features, lexical resources features, Knowledge-based features, and Meta-Information features. Simple surface features include features like character level unigram/n-gram, word generalization features include features like the bag-of-words, clustering, word embedding, paragraph embedding. Linguistic features include PoS tag of tokens. list of bad words or hate words can be considered as lexical resources. Existing knowledge base like ConceptNET, Bullyspace can be used as features.

Another type of problematic post is classified in the shared task eRisk(Losada and Crestani, 2016) within the Cross-Language Evaluation Forum (CLEF). Here, risk situations regarding health and safety are of interest and the research is dedicated to identifying such situations from social media data. In the first classification task, posts which indicate depression of a user need to be found. To simulate the identification of such condition early, the shared task provides 10% of the user data for 10 weeks.

Most of the approaches discussed above are the lexical approaches. Our approach is based on pre-trained word embedding. We have developed deep learning models which implicitly learn features from the text. Some of the lexical features like the length of the post, number of unique words are added explicitly in our model but the system performance was degraded.

3 Methodology and Data

In this section, first, we describe the datasets used in the experiment in 3.1 and in section 3.2 we will discuss our approaches in details

3.1 Dataset

Track organizers provided 15,001 aggression-annotated Facebook Posts and Comments each in Hindi (Romanized and Devanagari script) and English for training and validation (Kumar et al., 2018b). Table 2 shows a detail description of the training and validation Dataset.

Class	English Corpus		Hindi Corpus	
	# Training	**# Validation**	**# Training**	**# Validation**
NAG	5052	1233	2275	538
CAG	4240	1057	4869	1246
OAG	2708	711	4856	1217
Total	12000	3001	12000	3001

Table 2: Class distribution in the Training Dataset

Table 2 shows that there is class-Imbalance in the training data. For English corpus posts from NAG class are the highest and OAG class is lowest. Similarly, for Hindi corpus, posts from OAG class are the highest and NAG class is lowest. We will discuss the effect of class Imbalance on results in the methodology section. Table 3 gives details of the test data corpus. It is worth to note that track organizers provided an open domain Facebook corpus for training and validation but for testing, in addition to Facebook English and Hindi corpus, a Twitter corpus dataset was provided from the completely different domain for both English and Hindi languages.

Test Dataset	**# of posts**
Facebook English Corpus	916
Twitter English Corpus	1257
Facebook mixed script Hindi Corpus	970
Twitter mixed script Hindi Corpus	1194

Table 3: Test Data Corpus statistics

3.2 Methodology

In this section, we describe our various runs in details. First, we have implemented all standard machine learning classifiers like Multinomial Naive Bayes, Logistic Regression, SGD, KNN, SVC, Decision Tree, Random forest, various voting based ensemble soft and hard classifier with different text representation schemes like count based and TF/IDF to prepare baseline results. On the Facebook English Validation dataset, we got the best-weighted F1 = 0.57 in Logistic Regression with TF/IDF text representation scheme. KNN classifier gives worst weighted F1 = 0.36 on English Validation data. However soft voting based Ensemble of Logistic Regression, Naive Bayes, and Random Forest gives best weighted F1 =0.58 with count-based text representation scheme. As discussed in section 3.1, there is a class imbalance in the training Dataset. We have performed the experiments with equal class labels of training data but weighted F1 decreased substantially on validation data. So, we concluded that there is no need to tackle the class imbalance for this dataset.

3.2.1 Word Embedding for Text representation

Initially, we have implemented both variants of Word2vec namely CBOW and Skip-gram for word-embedding, but the accuracy was around 50%. We obtained the pre-trained Glove vector with different dimensions, but the accuracy improved only marginally. Finally, we settled with fastText (Mikolov et al., 2018) which is an extension of Word2vec. The fastText consider each word as N-gram characters. A word vector for a word is computed from the sum of the n-gram characters. Word2vec and Glove consider each word as a single unit and provide a word vector for each word. Since Facebook users make a lot of mistakes in spelling, typos, fastText is more convenient than Glove and Word2vec. In the followings, the main advantages of fastText are given over other embedding techniques.

1. fastText can generate word embedding for a word which is not processed during the training from its N-gram character features. Word2vec and Glove can't generate word embedding for unseen word.

2. For the rare word, fastText generates better word embedding than Word2vec and Glove due to the n-gram character features

3.2.2 Model Architecture

Track organizers had provided four Datasets from the open domain of Facebook and Twitter for the testing. For each dataset, we have submitted three runs. In this subsection, we will discuss the architecture of our deep neural network models and machine learning models which we developed during the training phase and tested on validation data.

Bidirectional LSTM This is our first deep neural network model which we developed to submit our first run. There is two LSTM layer instead of one in Bidirectional LSTM. Bidirectional LSTM is the extension of the single LSTM. Model parameters are as follows: Maximum features 10000 words; length of the sequence is 500. We believe that embedding layer is the most critical layer for the model. fastText (Mikolov et al., 2018) pre-trained vector is used for word embedding with embed size is 300. We have designed simple bidirectional LSTM with two fully connected layers. We add dropout to the hidden layer to counter overfitting. There are 50 memory units in LSTM and hidden layer. Adam optimization algorithm is used to update the weight matrix. 3 epochs are sufficient for the model to get overfitted.

Single LSTM with higher dropout: This model is based on traditional LSTM model with higher dropout. The first layer is an embedding layer with Maximum features 10000, The length of the sequence is 1024(maximum length of comment in the dataset). We believe that embedding layer is the most critical layer for the model. fastText (Mikolov et al., 2018) pre-trained vectors are used for word embedding with embed size is 300. There are 64 memory units in LSTM layer plus one dense layer with 256 nodes. We add dropout around 0.5 to counter overfitting in the hidden layer. Adam optimization algorithm is used to update the weight matrix. 3 epochs are sufficient to get the best validation accuracy around 58.4 % on English corpus and 61.1 % on Hindi corpus.

Model based on Convolution Neural Network This model is based on Convolution Neural Network. The first layer is an embedding layer with maximum features 10000; the length of the sequence is 1024(maximum length of comment in the dataset). The fastText (Mikolov et al., 2018) pre-trained vectors are used for word embedding with embed size is 300. We have added a one-dimensional convolution layer with 100 filters of height 2 and stride 1 to target biagrams. In addition to this, Global Max Pooling layer added. Pooling layer fetches the maximum value from the filters which are feed to the dense layer. There are 256 nodes in the hidden layer without any dropout. Validation accuracy is around 58.9 % on English corpus and 62.4 % on Hindi corpus.

Model based on Convolution Neural Network with different Filter height This model is by and large same with previous CNN model except for different one-dimensional filters with height 2,3,4 to target bigrams, trigrams, and four-grams features. After convolution layer and max pool layer, model concatenate max pooled result from each of one-dimensional convolution layer, then build one output layer on top of them. We have implemented this model from (Zhang and Wallace, 2015). There are 256 nodes in the hidden layer with 0.2 dropouts. Validation accuracy is around 57.6 % on English corpus and 62.4 % on Hindi corpus.

Model based Bidirectional GRU and Convolution Neural Network This model is the hybrid model of Recurrent Neural Network and Convolution Neural network. The model contains one embedding layer with pre-trained weight matrix from fastText, max features is 10000 and embed size is 300 followed by bidirectional GRU layer with 128 units and one- dimensional convolution layer with 64 filters having filter height 2. Validation accuracy is around 59 % on English corpus.

Voting based ensemble model This model is voting based ensemble model of Bidirectional LSTM, Single LSTM, CNN, Logistic Regression, BIGRU with CNN, the soft ensemble of (random forest classifier, Naive Bayes classifier, Logistic Regression).

Model based on Logistic Regression During the training phase, we got best validation accuracy from using logistic regression among all standard classifier. TF/IDF gives better accuracy than count based text representation scheme. We have set logistic regression parameters like N-gram, minimum document frequency using grid search. We have done little pre-processing on corpus like Non-ASCII character removal. Stop words are not removed. Validation accuracy is around 57.8 % on English corpus and 59.18 % on Hindi corpus.

4 Results

In this section, we first present results on the validation dataset. Table 4 and 5 show results on the English and the Hindi validation data corpus respectively.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 (weighted)**	**Text Repre. scheme**
Naive Bayes	0.5734	0.58	0.58	0.57	count based
Logistic Regression	0.5768	0.56	0.56	0.56	TF/IDF
KNN	0.4415	0.42	0.44	0.38	Count based
Linear SVC	0.5631	0.56	0.56	0.56 0	TF/IDF
Decision Tree	0.4841	0.48	0.48	0.48	count base
SGD	0.5691	0.57	0.57	0.57	TF/IDF
Random Forest	0.5101	0.5	0.51	0.49	TF/IDF
Soft ensemble	0.5894	0.59	0.59	0.58	count based
Hard Ensemble	0.5751	0.57	0.58	0.57	count based
LSTM NN + fasttext	0.584	0.59	0.58	0.59	word embedding
Convolution NN + fasttext	0.589	0.58	0.59	0.58	word embedding
Convolution Ngram+fasttext	0.576	0.6	0.58	0.58	word embedding
BIGRU+FASTTEXT	0.59	na	na	na	word embedding
Bidirectional LSTM + fasttext	0.5928	0.59	0.58	0.58	word embedding

Table 4: Results on English (Facebook) validation data

As we look at the result on validation data, we obtained best weighted F1 score and validation accuracy for English dataset in the soft ensemble of Logistic Regression, Random Forest, and SGD and in the Bidirectional LSTM. For Hindi corpus, we got the best weighted F1 score and validation accuracy in Convolution Neural Network.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 (weighted)**	**Text Repre. scheme**
Naive Bayes	0.5718	0.59	0.57	0.56	TF/IDF
Logistic Regression	0.5991	0.62	0.6	0.59	TF/IDF
KNN	0.3318	0.43	0.33	0.34	Count based
Linear SVC	0.5784	0.58	0.58	0.58 0	TF/IDF
Decision Tree	0.5211	0.52	0.52	0.52	count base
SGD	0.5924	0.60	0.59	0.59	TF/IDF
Random Forest	0.5511	0.56	0.55	0.55	TF/IDF
Soft ensemble	0.5981	0.61	0.60	0.60	TF/IDF
Hard Ensemble	0.5944	0.60	0.59	0.59	TF/IDF
LSTM NN + fasttext	0.611	0.6454	0.6111	0.6042	word embedding
Convolution NN + fasttext	0.624	0.6278	0.6241	0.6244	word embedding
Convolution Ngram NN	0.624	0.63	0.62	0.62	word embedding

Table 5: Results on Hindi (Facebook) validation data

System	**F1 (weighted)**
Random Baseline	0.3535
BiDirectional LSTM with fastText-run-1	0.5959
LSTM with higher dropout-run2	**0.6178**
N-Gram CNN-run-3	0.5580
top team saroyehun	0.6424

Table 6: Results for the English (Facebook) dataset

Table 6, 7, 8 and 9 shows our models results on different test datasets which are created from Facebook and Twitter posts. Figure 1, 2, 3 and 4 show the confusion matrices for the each dataset. One can observe that weighted F1 score on the test data is better than the validation dataset. Posts in the datasets are related to different events. Without any given context, our model gives a weighted F1 score around 0.6178 for English Facebook dataset and 0.6081 for Hindi Facebook Dataset. The Hindi and English Social media test datasets were created from the Twitter with 3 or 4 domain like # JNUShutdown, #Cricket2015, #demonetization. We have trained models on Facebook comments or posts and tested on Twitter posts. It is worth to note that there are lexical differences between Twitter posts and Facebook posts. Twitter posts are 140 characters long and the majority contain user mentions, external URL, and Hashtags while most of Facebook posts are longer than Twitter posts does not have Hashtags or user mentions in the text. However, our model gives a weighted F1 score around 0.5520 for English Twitter dataset and for mixed script Hindi Twitter dataset, our model gives F1 weighted around 0.4992 in model based on Convolution Neural network.

4.1 Result Analysis

In this subsection, we will present result analysis. The results on validation data show that models based on LSTM and CNN marginally outperform (around 2 % to 3%) standard machine learning classifiers with respect to weighted F1- score and accuracy on Facebook English corpus and Hindi corpus. Table 10 shows tweets which belong to the CAG class are classified under the NAG class. Table 11 shows the tweets which belong to NAG class are classified under the OAG class.

The main reasons for the posts which are failed to classified under CAG class are the unavailability of the context and difference with the wisdom of the human assessor. Same reasons can be applied to the tweets which are classified in OAG but actually they belong to NAG class.

System	F1 (weighted)
Random Baseline	0.3477
BIGRU-CNN-withfasttext-run-1	0.5486
CNN-withfastText-run-2	**0.5520**
BiDirectional LSTM with fastText-run-3	0.5423
Top team vista.ue	0.6008

Table 7: Results for the English (Social Media(Twitter) Dataset

System	F1 (weighted)
Random Baseline	0.3571
CNN-withfastText-run-1	**0.6081**
N-Gram CNN-run-2	0.5965
logistic Regression with Tf/IDF -run-3	0.6034
top team na14	0.6450

Table 8: Results for the Hindi (Facebook) Dataset

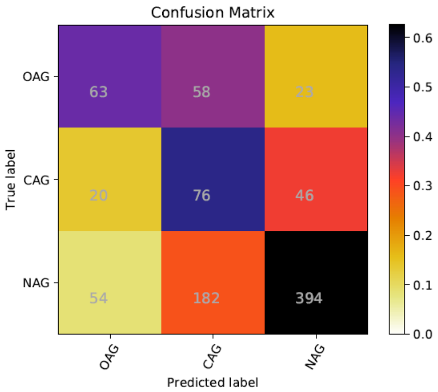

Figure 1: Confusion Matrix for EN-FB task

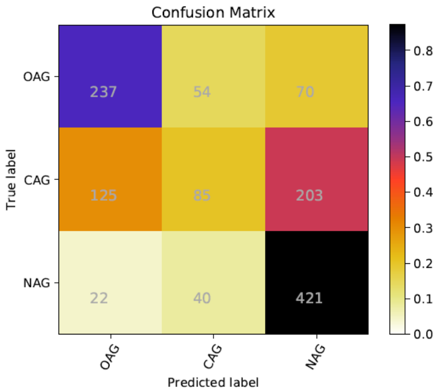

Figure 2: Confusion Matrix for EN-TW task

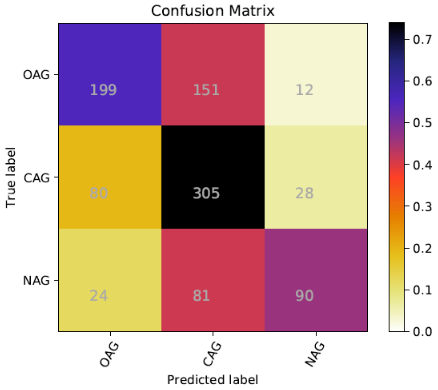

Figure 3: Confusion Matrix for HI-FB task

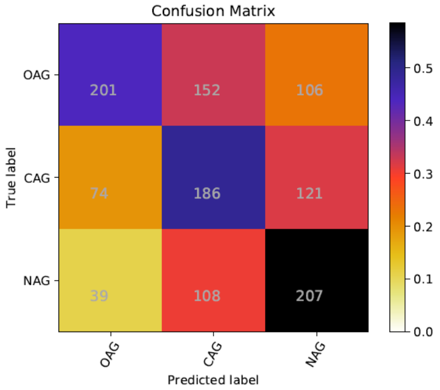

Figure 4: Confusion Matrix for HI-TW task

System	**F1 (weighted)**
Random Baseline	0.3206
CNN-withfastText-run-1	**0.4992**
LSTM with higher dropout-run2	0.4579
voting based ensemble	0.4797
top team DA-LD-Hildesheim	0.4992

Table 9: Results for the Hindi (Social Media(Twitter)) Dataset.

Post-id	**Text**	**Actual Class**	**Predicted Class**
facebook-467581	modi should learn from him how what to communicate common people not to corporate people.	CAG	NAG
facebook-442501	The name and the meaning has changed of RBI to reverse bank of India.	CAG	NAG

Table 10: Tweets which are hard to classify in CAG class

Post-id	**Text**	**Actual Class**	**Predicted Class**
facebook-408314,	Pakistan is a terrorist country and no body would like to play with Pakistan.	NAG	OAG
facebook-355578,	Gandhi Killer Ram (Nathu) Temple in India Big Shame 4India.	NAG	OAG
facebook-400906	You have to goes for this facts in U N O in USA and then you can claims for Kashmir.	NAG	OAG

Table 11: Tweets belongs to NAG but classified under OAG class

5 Conclusion

After performing exhaustive experiments, we conclude that Deep Neural Network with proper word embedding marginally outperforms all standard machine learning classifier and ensemble techniques. The critical parameters for the models are the batch size and learning rate. We also concluded that higher drop out will help to counter model overfitting and improvise a standard evaluation metric. CNN and LSTM are the better models for these datasets. On the English test corpus, we obtained a better F1 score for NAG class and poor F1-score for CAG class which supports the previous (Malmasi and Zampieri, 2017) findings. For the Facebook Hindi test corpus, the same seems not to be true. We obtained a better F1 score for CAG class than NAG class. In the future, we will focus on various pre-trained word embedding models and study how the word is represented by this model. we have planned to develop deeper Neural nets and identify optimal parameters using grid search. It is also to be noted that the model leads to poor result on test data created from the different source than the training corpus source.

Acknowledgements

We would like to acknowledge undergraduate students of LDRP-ITR Mr.Daksh Patel, Mr.Aditya Patel, Mr.Yash Madhani, Mr.Kishan Thakkar, and Miss. Krishna Tank for their support in post hoc analysis and they have prepared bad-word files for both the English and Hindi languages(Romanized and Devanagari). In the future, we would like to incorporate these lexical resources features in our models.

References

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

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

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018a. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)* Santa Fe, USA.

Ritesh Kumar, Aishwarya N. Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018b. Aggression-annotated Corpus of Hindi-English Code-mixed Data. In *Proceedings of the 11th Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting Tweets Against Blacks. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*

David E. Losada and Fabio Crestani. 2016. A test collection for research on depression and language use. In *Conference Labs of the Evaluation Forum*, pages 28–39. Springer.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

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