Scmhl5 at TRAC-2 Shared Task on Aggression Identification: Bert Based Ensemble Learning Approach

Han Liu, Pete Burnap, Wafa Alorainy, Matthew L. Williams
Cardiff University, Cardiff, United Kingdom
{liuh48, burnapp, alorainyws, williamsm7}@cardiff.ac.uk

Abstract

This paper presents a system developed during our participation (team name: scmhl5) in the TRAC-2 Shared Task on aggression identification. In particular, we participated in English Sub-task A on three-class classification ('Overtly Aggressive', 'Covertly Aggressive' and 'Non-aggressive') and English Sub-task B on binary classification for Misogynistic Aggression ('gendered' or 'non-gendered'). For both sub-tasks, our method involves using the pre-trained Bert model for extracting the text of each instance into a 768-dimensional vector of embeddings, and then training an ensemble of classifiers on the embedding features. Our method obtained accuracy of 0.703 and weighted F-measure of 0.664 for Sub-task A, whereas for Sub-task B the accuracy was 0.869 and weighted F-measure was 0.851. In terms of the rankings, the weighted F-measure obtained using our method for Sub-task A is ranked in the 10th out of 16 teams, whereas for Sub-task B the weighted F-measure is ranked in the 8th out of 15 teams.

Keywords: Bert, Ensemble Learning, Aggression Identification, Word Embedding

1. Introduction

In the era of social networks, we have witnessed an increase in people misusing the platforms for propagating messages that are offensive and/or aggressive. Therefore, it has been a priority research topic for people to develop tools for automatic detection of offensive language (Burnap and Williams, 2015; Burnap and Williams, 2016).

Due to the rapid growth of data relating to online social interactions, machine learning approaches have been increasingly popular for natural language processing in social media analysis, such as word embedding through neural network based learning approaches. In this paper, we describe a system based on Bert embedding and ensemble learning, for participating in a shared task on aggression identification in the Second Workshop on Trolling, Aggression and Cyberbullying. In particular, we entered two sub-tasks (A and B) of the above-mentioned shared task, where one is about a three-class classification task for identifying that a text message is 'Overtly Aggressive' (OAG), 'Covertly Aggressive' (CAG) or 'Non-aggressive' (NAG), whereas the other one is about a binary classification task for identifying that a message is 'gendered' (GEN) or 'non-gendered' (NEGN). We obtained accuracy of 0.703 and weighted F-measure of 0.664 for Sub-task A, whereas for Sub-task B the accuracy and weighted F-measure were 0.869 and 0.851, respectively. Moreover, the weighted F-measure obtained using our method for Sub-task A is ranked in the 10th out of 16 teams, where the weighted F-measure ranked in the first place is 0.803. For Sub-task B, the weighted F-measure obtained using our method is ranked in the 8th out of 15 teams, where the weighted F-measure ranked in the first place 0.872.

The rest of this paper is organized as follows: Section 2 provides a review of recently published works on identification of aggressive languages. In Section 3, we describe the shared task dataset in detail and present the method that we adopted for developing our system for aggression identification. In Section 4, we report the results obtained on both the validation data and the test data. In Section 5, the conclusion of this paper is drawn and some further directions are suggested towards advancing the effectiveness of aggression identification.

2. Related Work

Since the spread of online offensive and/or aggressive language could lead to disruptive anti-social outcomes, it has become critical in many countries to consider the posting of such language as a legal issue (Banks, 2010) and to take actions against the propagation of aggression, cyberbullying and hate speech (Banks, 2011).

In the context of machine learning based identification of offensive and/or aggressive language, traditional approaches of feature extraction from text include Bag-of-Words (BOW) (Kwok and Wang, 2013; Liu et al., 2019a), N-grams (NG) in word level (Perez and Luque, 2019; Liu and Forss, 2014; Watanabe et al., 2018), NG in character level (Gambäck and Sikdar, 2017; Perez and Luque, 2019), typed dependencies (Burnap and Williams, 2016), part-of-speech tags (Davidson et al., 2017), dictionary based approaches (Tulkens et al., 2016) and othering lexicons (Burnap and Williams, 2016; Alorainy et al., 2019). Some traditional learning approaches used for training classifiers include Support Vector Machine (SVM) (Burnap and Williams, 2016; Indurthi et al., 2019; Perez and Luque, 2019; Orasan, 2018), Naive Bayes (NB) (Kwok and Wang, 2013; Liu et al., 2019a), Decision Trees (DT) (Watanabe et al., 2018; Liu et al., 2019a), Logistic Regression (LR) (Xiang et al., 2012; Waseem and Hovy, 2016), decision tree ensembles such as Random Forest (RF) (Burnap and Williams, 2015; Orasan, 2018) and Gradient Boosted Trees (Badjatiya et al., 2017), ensembles based on SVM (Malmasi and Zampieri, 2018) and fuzzy approaches (Liu et al., 2019a; Liu et al., 2019b).

Moreover, some challenges in terms of discriminating hate speech from profanity have been highlighted in (Malmasi and Zampieri, 2018) for justifying the necessity of extracting deeper features instead of superficial ones (e.g., BOW

and NG). From this perspective, embedding learning approaches have recently become the state of the art for automatic extraction of semantic features, e.g. Word2Vec (Nobata et al., 2016), Glove (Zhang et al., 2018; Badjatiya et al., 2017; Kshirsagar et al., 2018; Orasan, 2018), FastText (Pratiwi et al., 2018; Herwanto et al., 2019; Galery et al., 2018). There are also some end-to-end learning approaches of Deep Neural Networks (DNN) (Nina-Alcocer, 2019; Yuan et al., 2016; Ribeiro and Silva, 2019), e.g. Convolutional Neural Networks (CNN) (Gambäck and Sikdar, 2017; Park and Fung, 2017; Roy et al., 2018; Huang et al., 2018), Long-Short Term Memory (LSTM) (Badjatiya et al., 2017; Pitsilis et al., 2018; Nikhil et al., 2018; Kumar et al., 2018) and Gated Recurrent Unit (GRU) (Zhang et al., 2018; Galery et al., 2018) or combination of different DNN architectures in an ensemble setting (Madisetty and Desarkar, 2018), which are adopted for enhancement of feature representation and classification, based on word embeddings produced by Word2Vec, Glove or FastText. However, embedding approaches such as Word2Vec can not achieve contextualized representation of words, i.e. the same word used in different contexts is represented in the same numeric vector using the above-mentioned approaches, which could affect the classification performance due to the lack of contextual information from the features. In order to achieve effectively contextualized representation of features, some more advanced embedding approaches including ELMo (Bojkovsky and Pikuliak, 2019) and Bert (Mozafari et al., 2019; Nikolov and Radivchev, 2019) have recently been developed showing the state of the art performance for offensive and/or aggressive language identification and other similar tasks of natural language processing. There are also applications of Bert in the setting of ensemble learning, e.g. an ensemble of Bert models has been applied to an offensive language identification shared task (Risch et al., 2019).

3. Methodology and Data

In this section, we will provide details of the data set provided for the shared task and present the procedure of our method in detail.

3.1. Dataset

The dataset (Bhattacharya et al., 2020) provided for the shared task contains 6529 text instances in total, which involves a training set of 4263 instances, a validation set of 1066 instances and a test set of 1200 instances. The characteristics of the data set are shown in Table 1.

Table 1: Class Frequency on Training, Validation and Test Sets

Task	Class	Training Set	Validation Set	Test set
Sub-task EN-A	NAG	3375	836	690
	CAG	453	117	224
	OAG	435	113	286
Sub-task EN-B	NGEN	3954	993	1025
	GEN	309	73	175

For Sub-task A, the frequency distribution among the three classes 'NAG', 'CAG' and 'OAG' in the training set is 3375:453:435, whereas the distributions in the validation and test sets are 836:117:113 and 690:224:286, respectively. The above details indicate that the training set has a class frequency distribution very similar to the one in the validation set but the validation set and the test set show considerably different distributions, which may lead to the case that the performance obtained on the validation set is different from the one obtained on the test set.

For Sub-task B, the frequency distribution between the two classes 'NGEN' and 'GEN' is 3954:309, whereas the distributions in the validation and test sets are 993:73 and 1025:175, respectively. Similar to the characteristic found for Sub-task A, the above details for Sub-task B indicate again a considerable difference on the class frequency distribution between the validation set and the test set, while the training set and the validation set show very similar distributions. The above characteristic may also result in the case that the performance obtained on the validation set is different from the one obtained on the test set.

3.2. Method

The method used for Sub-task A on aggression identification involves two main steps, namely, extraction of embedding features and ensemble learning for classification. Before the two main steps, the text for each instance is preprocessed by removing hashtags, mentions and URLs, converting all words to their lower cases and transforming all emojis to their text descriptions.

In the feature extraction step, each text instance is transformed into a 768-dimensional feature vector by using the pre-trained Bert embedding model (Devlin et al., 2018). In particular, we used the base uncased model of Bert, which consists of 12 layers alongside 768 units per layer. In this setting, each token (word) is transformed into a 768-dimensional vector, so an instance that involves m tokens would be represented in the form of a $m \times 768$ matrix (m vectors). On this basis, the 768-dimensional feature vector of each instance is obtained by averaging the above-mentioned m word vectors.

In the classification step, the classifier is trained in the setting of ensemble learning. In particular, the creation of an ensemble through our designed approach involves four levels, namely, feature sub-sampling, class imbalance handling, multi-class handling and training of base classifiers. The whole framework of ensemble setting is illustrated in Fig. 1.

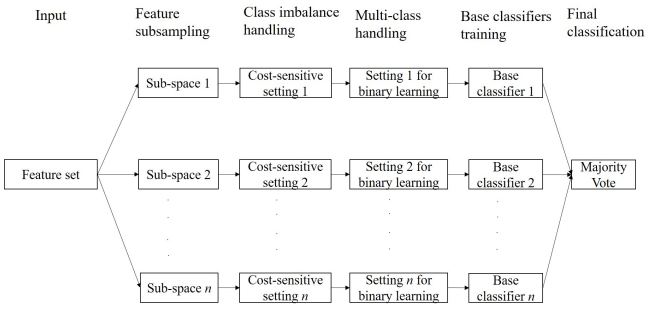

Figure 1: Framework of Ensemble Setting

In the top level for feature sub-sampling, the aim is to encourage the creation of diversity among base classifiers,

which is achieved by adopting the random subspace (RS) method (Ho, 1998) to draw n subsets of the original feature set, such that n different classifiers are trained on the n feature subsets.

In the second level for class imbalance handling, a cost-sensitive learning method is adopted to enable the classifier trained on each feature subset (drawn in the top level) to be cost-sensitive, no matter which one of the supervised learning algorithms is adopted for training classifiers.

In the third level for multi-class handling, the aim is to transform the 3-class classification problem for suiting a 2-class learning algorithm, i.e. some algorithms cannot directly perform multi-class learning, so a specific strategy of multi-class handling needs to be involved to enable that 2-class learning algorithms can work. Some popular strategies include 'one-against-all', 'one-against-one', 'random error correction code' and 'exhaustive error correction code'.

In the fourth level for training of base classifiers, a supervised learning algorithm needs to be adopted, where the Stochastic Gradient Descent algorithm is chosen in our setting for training n linear classifiers on the n feature subsets produced by the RS method. The final classification is made by fusing the outputs of n linear classifiers through majority voting.

The method used for Sub-task B on identification of misogynistic aggression is almost the same as the one adopted for Sub-task A, but the only difference is that the third level for multi-class handling is dropped, due to the fact that Sub-task B involves a binary classification problem. Therefore, the method used for Sub-task B involves three levels, namely, feature sub-sampling, class imbalance handling and training of base classifiers.

4. Results

In this section, we describe the experimental setup and discuss the results obtained in the development and testing stages.

4.1. Development Stage

In the development stage, we conducted experiments by using the pre-trained Bert embedding model and various learning algorithms, namely, Support vector machine (SVM), Naive Bayes (NB), Stochastic Gradient Descent (SGD) and a fuzzy rule learning approach (Fuzzy) (Huehn and Huellermeier, 2009), due to their relatively low computational complexity and the suitability of this kind of traditional learning algorithms for processing small data (Liu et al., 2019a). In particular, the results shown in Tables 2 and 3 were obtained by using the validation set for evaluating the performance of classifiers produced by various algorithms and determining which algorithm is used to train the base classifiers in the setting of random subspace based ensemble learning.

Before feature extraction, all the instances were pre-processed by removing hashtags, mentions and URLs and converting all words to their lower cases. Also, all the emojis were transformed into their text descriptions by using the emoji-java library[1].

In the feature extraction stage, each text instance was transformed into a 768-dimensional feature vector using the pre-trained base uncased model of Bert, which is based on the Java library of easy-bert[2]. The above decision is based on the considerations that a base Bert model requires less memory than a large Bert model and all words in the text for each instance have been converted to lower cases in the pre-processing stage leading to the unnecessity of using a cased Bert model.

In the classification stage, we used the implementations of various algorithms from the Weka library (Hall et al., 2009). In terms of hyper-parameter settings, SVM was set to normalize the training data and train a non-linear classifier using the polynomial kernel and the sequential minimal optimization algorithm (SMO) (Platt, 1998), where the complexity parameter C is set to 1.0 and the batch size is set to 100. The fuzzy rule learning approach was set to involve 2 runs of rule optimization and using 1/3 of the training data for rule pruning, where the product T-norm was used to compute the degree to which an instance is covered by a fuzzy rule and the rule stretching method (Huehn and Huellermeier, 2009) is adopted to classify any instances that are not covered by any fuzzy rules. SGD was set to train a linear classifier using the Hinge loss with the learning rate (lr) of 0.01 through 500 epochs, where the batch size was set to 100 and the regularization constant is set to 0.0001. Moreover, all of the algorithms (SVM, NB, Fuzzy and SGD) were adopted for training classifiers in a cost sensitive setting, i.e. the trained classifiers are made cost-sensitive by assigning higher cost to the case of misclassifying instances of the minority class. In addition, due to the case that SGD is essentially a two-class learning algorithm, the three-class classification problem was transformed to suit classifiers trained by SGD through using the 'random error correction code' method.

Table 2: Results on Validation Data for Sub-task EN-A

Method	F1(NAG)	F1(CAG)	F1(OAG)	F1(Weighted)	Accuracy
SVM	0.890	0.016	0.337	0.735	**0.796**
NB	0.557	**0.261**	0.084	0.475	0.414
Fuzzy	0.868	0.126	0.228	0.719	0.757
SGD	0.886	0.017	**0.367**	0.736	**0.796**
RS	**0.891**	0.101	0.269	**0.738**	0.794

For Sub-task A, the results obtained on the validation set are shown in Table 2, which indicates that SGD and SVM perform considerably better than NB and the fuzzy approach. Although SVM and SGD show almost the same performance in terms of weighted F-measure, SGD outperforms SVM for the minority class 'OAG'. Moreover, SGD is capable of training updateable classifiers in the setting of incremental learning, i.e., previously trained classifiers can be updated by learning incrementally from instances newly added into the training set. This is an essential advantage of SGD in comparison with SVM (based on SMO) that cannot effectively achieve incremental learning. Therefore, we chose to adopt the SGD algorithm for training and optimizing base classifiers in the setting of ensemble learning, in order to achieve a more effective way of advancing the per-

[1] https://github.com/vdurmont/emoji-java

[2] https://github.com/robrua/easy-bert

formance further using a new/updated data set without the need to retrain each base classifier.

The ensemble is created following the procedure shown in Fig. 1. In particular, the RS method is adopted to draw 10 feature subsets, where the size of each subspace is set to 0.5, so there are totally 10 base classifiers trained on the 10 feature subsets. The hyper-parameter settings of SGD are exactly the same as the ones described above about training a single classifier. The results shown in Table 2 indicate that the creation of an ensemble in the above settings leads to a marginal improvement of the performance in comparison with the production of a single classifier by SGD.

For Sub-task B, we followed the same procedure for text pre-processing, feature extraction and classification. For training of the classifiers, we adopted the same set of algorithms (with the same settings of hyper-parameters) for evaluating performance on the validation set. The results shown in Table 3 indicate again the phenomenon that SGD and SVM perform considerably better than NB and the fuzzy approach. Although SGD performs marginally worse than SVM in terms of weighted F-measure, SGD outperforms SVM for the minority class 'GEN'. As mentioned earlier in this section, SGD is capable of updating previously trained classifiers by learning incrementally from instances newly added into the training set, so we chose to adopt the SGD algorithm again for training and optimizing base classifiers in the setting of ensemble learning.

Table 3: Results on Validation Data for Sub-task EN-B

Method	F1(NGEN)	F1(GEN)	F1(Weighted)	Accuracy
SVM	0.967	0.171	0.912	0.936
NB	0.566	0.152	0.538	0.426
Fuzzy	0.96	0.146	0.904	0.923
SGD	0.959	0.265	0.911	0.922
RS	0.965	0.417	**0.928**	0.934

Following the same ensemble settings adopted for Sub-task A, an ensemble of SGD classifiers is built with a cost-sensitive setting for Sub-task B, but the step for multi-class handling is dropped, given that Sub-task B is a binary classification task. The results shown in Table 3 indicate that the creation of an ensemble leads to an improvement of the performance on weighted F-measure and the score for the minority class, in comparison with the production of a single classifier by using any one of the standard learning algorithms.

4.2. Testing Stage

Based on the results shown in Tables 2 and 3 for the two sub-tasks, we merged the training and validation sets for augmenting the sample size for creating an ensemble of classifiers in the above-described setting (based on Bert, RS and SGD). The results obtained on the test set for the two sub-tasks are shown in Table 4.

It can be seen from Table 4 that the performance obtained on the test set gets considerably lower (by about 7%) in comparison with the one obtained on the validation set for both Sub-tasks A and B, which is likely due to the difference on the data distribution between the two sets of instances, i.e. the weight of the majority class gets lower on the test set, in comparison with the weight on the validation set, for both Sub-tasks.

Table 4: Performance on Test Data

Task	Class	F1(Class)	F1(Weighted)	Accuracy
Sub-task EN-A	NAG	0.8152	0.6637	0.7025
	CAG	0.3106		
	OAG	0.5746		
Sub-task EN-B	NGEN	0.9264	0.8514	0.8692
	GEN	0.4120		

For Sub-task A, comparing the results shown in Table 2 and Table 4, we can see that the weighted F1-score gets lower on the test set, which seems to be due mainly to the case that the F1-score for the majority class 'NAG' gets lower. Moreover, the F1-scores for the other two classes 'CAG' and 'OAG' get much higher on the test set. Given that the class frequency distribution among the three classes 'NAG', 'CAG' and 'OAG' is 836:117:113 on the validation set and is 690:224:286 on the test set, it seems that the performance difference is likely to result from the difference on the data distribution.

For Sub-task B, comparing the results shown in Table 3 and Table 4, we can see again that the weighted F1-score gets lower on the test set, which seems to be due mainly to the case that the F1-score for the majority class 'NGEN' gets lower. Moreover, for the minority class 'GEN', the F1-score obtained on the test set is almost the same as the score obtained on the validation set. Given that the frequency distribution between the two classes 'NGEN' and 'GEN' is 993:73 on the validation set and is 1025:175 on the test set, it seems that the change in the data distribution does not really impact on the performance for the minority class 'GEN" but shows a considerable impact on the performance for the majority class 'NGEN'.

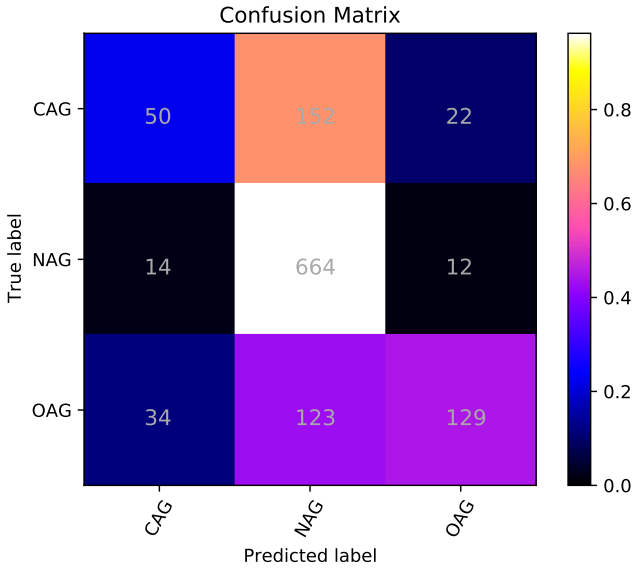

Figure 2: Sub-task EN-A, scmhl5 CodaLab 571565 (An ensemble of SGD classifiers trained on embedding features prepared by Bert and RS)

More detailed results obtained on the test set for the two sub-tasks are shown in Figs. 2 and 3 in the form of con-

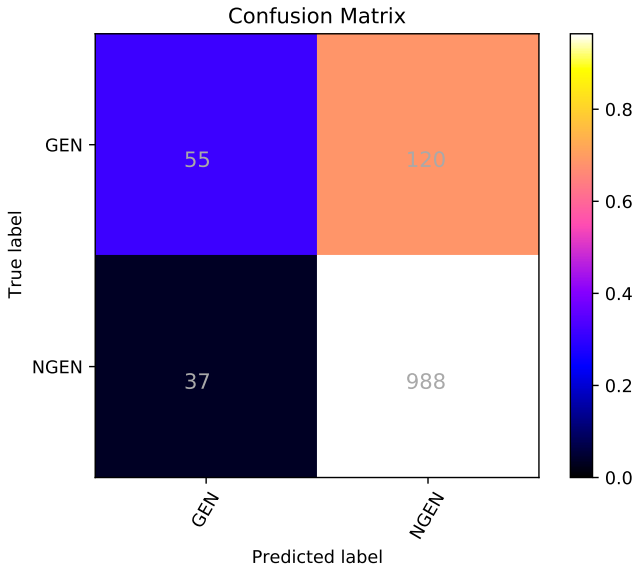

Figure 3: Sub-task EN-B, scmhl5 CodaLab 571564 (An ensemble of SGD classifiers trained on embedding features prepared by Bert and RS)

fusion matrixes, which indicate that the cases of incorrect classifications mainly result from false negatives for the minority class, i.e. some instances of aggressive language were not successfully detected due to the insufficient ability to generalize thoroughly on test instances.

Based on the results shown in Table 4 and Figs. 2 and 3, we tried to reduce the learning rate (lr) from 0.01 to 0.005 towards achieving better optimization of the parameters of the SGD classifiers, i.e. reducing the learning rate can generally help better avoid the case of local optimization. The results obtained by using the lower value of 'lr' are shown in Tables 5 and 6, which indicate that the performance gets slightly lower after reducing the learning rate for both sub-tasks A and B. The results suggest that the reduction of the learning rate may increase the chance of overfitting on a small data set and thus lower the generalization performance on test data.

Table 5: Results for Sub-task EN-A (obtained by deploying an ensemble of SGD classifiers trained on embedding features prepared by Bert and RS).

System	F1 (weighted)	Accuracy
Bert+RS+SGD(lr=0.01)	**0.6637**	0.7025
Bert+RS+SGD(lr=0.005)	0.6300	0.6842

Table 6: Results for Sub-task EN-B (obtained by deploying an ensemble of SGD classifiers trained on embedding features prepared by Bert and RS).

System	F1 (weighted)	Accuracy
Bert+RS+SGD(lr=0.01)	**0.8514**	0.8692
Bert+RS+SGD(lr=0.005)	0.8428	0.87

5. Conclusion

We participated in the shared task on aggression identification in the 2nd Workshop on Trolling, Aggression and Cyberbullying. In particular, we entered two English sub-tasks (A and B) for identifying the intensity of aggression (i.e. 'Overtly Aggressive', 'Covertly Aggressive' or 'Non-aggressive') and detecting misogynistic aggression (i.e. 'gendered' or 'non-gendered'). We built two systems for the above-mentioned sub-tasks, and both systems were built in the setting of ensemble learning based on the embedding features extracted using the pre-trained Bert model. We obtained a weighted F1-score of 0.664 for Sub-task A and a score of 0.851 for Sub-task B.

In future, we will explore the effectiveness of extracting multiple types of embedding features using various embedding models (e.g. Bert and ELMo), towards achieving more advanced settings of ensemble learning through both early fusion (in the feature level) and late fusion (in the classification level). It is also worth exploring the use of a larger volume of external data for updating the SGD classifiers in the setting of incremental learning, towards advancing the generalization performance further. In addition, we will add a further experiment by selecting a subset of the test set that has the same class frequency distribution as the validation set, in order to investigate whether the performance obtained on the test subset can be more similar to the one obtained on the validation set after making the class frequency distribution consistent between the two data sets.

Bibliographical References

Alorainy, W., Burnap, P., Liu, H., and Williams, M. (2019). The enemy among us: Detecting cyber hate speech with threats-based othering language embeddings. *ACM Transactions on the Web*, 13(3):1–26.

Badjatiya, P., Gupta, S., Gupta, M., and Varma, V. (2017). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760, Perth, Australia, 3-7 April.

Banks, J. (2010). Regulating hate speech online. *International Review of Law, Computers and Technology*, 24(3):233–239.

Banks, J. (2011). European regulation of cross-border hate speech in cyberspace: The limits of legislation. *European Journal of Crime, Criminal Law and Criminal Justice*, 19(1):1–13.

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., and Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression.

Bojkovsky, M. and Pikuliak, M. (2019). STUFIIT at SemEval-2019 Task 5: Multilingual hate speech detection on twitter with MUSE and ELMo embeddings. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 464–468, Minneapolis, Minnesota, USA, 6-7 June.

Burnap, P. and Williams, M. L. (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.

Burnap, P. and Williams, M. (2016). Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science*, 5(11).

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. In Marilyn Walker, et al., editors, *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans, Louisiana. Association for Computational Linguistics.

Galery, T., Charitos, E., and Tian, Y. (2018). Aggression identification and multi lingual word embeddings. In Ritesh Kumar, et al., editors, *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Gambäck, B. and Sikdar, U. K. (2017). Using Convolutional Neural Networks to Classify Hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.

Herwanto, G. B., Ningtyas, A. M., Nugraha, K. E., and Trisna, I. N. P. (2019). Hate speech and abusive language classification using fastText. In *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia, 5-6 December.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Huang, Q., Inkpen, D., Zhang, J., and Bruwaene, D. V. (2018). Cyberbullying intervention based on convolutional neural networks. In Ritesh Kumar, et al., editors, *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Huehn, J. C. and Huellermeier, E. (2009). FURIA: An algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19:293–319.

Indurthi, V., Syed, B., Shrivastava, M., Chakravartula, N., Gupta, M., and Varma, V. (2019). Fermi at SemEval-2019 Task 5: Using sentence embeddings to identify hate speech against immigrants and women on twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 70–74, Minneapolis, Minnesota, USA, 6-7 June.

Kshirsagar, R., Cukuvac, T., McKeown, K., and McGregor, S. (2018). Predictive embeddings for hate speech detection on twitter. In *Proceedings of the Second Workshop on Abusive Language Online (ALW2)*, pages 26–32, Brussels, Belgium, 31 October.

Kumar, R., Bhanodai, G., Pamula, R., and Chennuru, M. R. (2018). TRAC-1 shared task on aggression identification: IIT(ISM)@COLING'18. In Ritesh Kumar, et al., editors, *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kwok, I. and Wang, Y. (2013). Locate the hate: Detecting Tweets Against Blacks. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Liu, S. and Forss, T. (2014). Combining N-gram based similarity analysis with sentiment analysis in web content classification. In *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 530–537, Rome, Italy, 21-24 October.

Liu, H., Burnap, P., Alorainy, W., and Williams, M. L. (2019a). A fuzzy approach to text classification with two stage training for ambiguous instances. *IEEE Transactions on Computational Social Systems*, 6(2):227–240.

Liu, H., Burnap, P., Alorainy, W., and Williams, M. L. (2019b). Fuzzy multi-task learing for hate speech type identification. In *WWW '19 The World Wide Web Conference*, pages 3006–3012, San Francisco, CA, USA, 13-17 May.

Madisetty, S. and Desarkar, M. S. (2018). Aggression detection in social media using deep neural networks. In Ritesh Kumar, et al., editors, *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Malmasi, S. and Zampieri, M. (2018). Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Mozafari, M., Farahbakhsh, R., and Crespi, N. (2019). A BERT-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*, pages 928–940, Lisbon, Portugal, 10-12 December.

Nikhil, N., Pahwa, R., Nirala, M. K., and Khilnani, R. (2018). LSTMs with attention for aggression detection. In Ritesh Kumar, et al., editors, *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nikolov, A. and Radivchev, V. (2019). Nikolov-Radivchev at SemEval-2019 Task 6: Offensive tweet classification with BERT and ensembles. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 691–695, Minneapolis, Minnesota, USA, 6-7 June.

Nina-Alcocer, V. (2019). HATERecognizer at SemEval-2019 Task 5: Using features and neural networks to face hate recognition. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 409–415, Minneapolis, Minnesota, USA, 6-7 June.

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., and Chang, Y. (2016). Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–

153. International World Wide Web Conferences Steering Committee.

Orasan, C. (2018). Aggressive language identification using word embeddings and sentiment features. In Ritesh Kumar, et al., editors, *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Park, J. H. and Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter. In *1st Workshop on Abusive Language Online*, pages 41–45, Vancouver, Canada, 4 August.

Perez, J. M. and Luque, F. M. (2019). Atalaya at SemEval 2019 Task 5: Robust embeddings for tweet classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 64–69, Minneapolis, Minnesota, USA, 6-7 June.

Pitsilis, G. K., Ramampiaro, H., and Langseth, H. (2018). Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48(12):4730–4742.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In Bernhard Scholkopf, et al., editors, *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, USA. MIT Press.

Pratiwi, N. I., Budi, I., and Alfina, I. (2018). Hate speech detection on indonesian instagram comments using FastText approach. In *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Yogyakarta, Indonesia, 27-28 October.

Ribeiro, A. and Silva, N. (2019). INF-HatEval at SemEval-2019 Task 5: Convolutional neural networks for hate speech detection against women and immigrants on twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 420–425, Minneapolis, Minnesota, USA, 6-7 June.

Risch, J., Stoll, A., Ziegele, M., and Krestel, R. (2019). hpiDEDIS at GermEval 2019: Offensive language identification using a German BERT model. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*, pages 403–408, Erlangen, Germany, 8-11 October. German Society for Computational Linguistics & Language Technology.

Roy, A., Kapil, P., Basak, K., and Ekbal, A. (2018). An ensemble approach for aggression identification in English and Hindi text. In Ritesh Kumar, et al., editors, *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tulkens, S., Hilte, L., Lodewyckx, E., Verhoeven, B., and Daelemans, W. (2016). A Dictionary-based Approach to Racism Detection in Dutch Social Media. In *Proceedings of the Workshop Text Analytics for Cybersecurity and Online Safety (TA-COS)*, Portoroz, Slovenia.

Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of NAACL-HLT 2016*, pages 88–93, San Diego, California, USA, 12-17 June.

Watanabe, H., Bouazizi, M., and Ohtsuki, T. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, PP(99):1–11.

Xiang, G., Fan, B., Wang, L., Hong, J., and Rose, C. (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*, pages 1980–1984, Maui, Hawaii, USA, 29 October-2 November.

Yuan, S., Wu, X., and Xiang, Y. (2016). A two phase deep learning model for identifying discrimination from tweets. In *19th International Conference on Extending Database Technology*, pages 696–697, Bordeaux, France, 15-18 March.

Zhang, Z., Robinson, D., and Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Lecture Notes in Computer Science*. Springer Verlag.