Combining Shallow and Deep Learning for Aggressive Text Detection

Viktor Golem **Mladen Karan** **Jan Šnajder**
Faculty of Electrical Engineering and Computing, University of Zagreb
Text Analysis and Knowledge Engineering Lab
{viktor.golem,mladen.karan,jan.snajder}@fer.hr

Abstract

We describe the participation of team *TakeLab* in the aggression detection shared task at the TRAC1 workshop for English. Aggression manifests in a variety of ways. Unlike some forms of aggression that are impossible to prevent in day-to-day life, aggressive speech abounding on social networks could in principle be prevented or at least reduced by simply disabling users that post aggressively worded messages. The first step in achieving this is to detect such messages. The task, however, is far from being trivial, as what is considered as aggressive speech can be quite subjective, and the task is further complicated by the noisy nature of user-generated text on social networks. Our system learns to distinguish between open aggression, covert aggression, and non-aggression in social media texts. We tried different machine learning approaches, including traditional (shallow) machine learning models, deep learning models, and a combination of both. We achieved respectable results, ranking 4th and 8th out of 31 submissions on the Facebook and Twitter test sets, respectively.

1 Introduction

Violence has always been present in human society. As it evolved and technology improved over time, forms of violence changed as well. While only a few decades ago most of the physical and psychological abuse occurred face-to-face, today a lot of psychological violence gets through to the victim via the Internet, and in most cases, social networks. This kind of violence might even be worse, as it can take place at any time, regardless of the physical distance between the victim and the perpetrator. Moreover, social consequences that a person would endure for aggressive speech in real life are virtually absent on the Internet, lowering the inhibitions of potential perpetrators. Although it is next to impossible to prevent people from being rude in real-life conversations, violence through social networks might be alleviated by simply making it impossible to send or share offensive content.

The first step on that path is figuring out which among the millions of messages or posts are indeed aggressive. While it is always possible to rely on the users themselves, e.g., by allowing them to report offensive and inappropriate content, the most common and fastest way to detect aggressive posts is by using supervised machine learning. One option is to frame the task as a binary classification problem and learn a model that discerns between aggressive and non-aggressive speech. In this context, we can define aggressive speech as any kind of text which is offensive or inappropriate. A more ambitious alternative is to subcategorize the aggressive speech into specific types such as racism, sexism, homophobia, trolling, cyberbullying,[1] etc. A successful system employing either approach would obviously be of significant practical value, as it would facilitate detecting and intercepting the aggressive texts before they reach their intended victim, as well as allow implementation of disciplinary measures to discourage aggressive behaviour. This offers considerable motivation for pursuing this strand of research.

Building a system for aggressive text detection using machine learning was the goal at the TRAC1 shared task on aggression detection (Kumar et al., 2018). The goal was to build a system that can label messages from a given dataset as *openly aggressive*, *covertly aggressive*, or *not aggressive*. Open

[1] The difference between aggression and bullying is that aggression can be a one-off situation or remark that the perpetrator does not necessarily recognize as being wrong, whereas bullying is a malicious and targeted approach of repeated aggression where the perpetrator has an intent to harm the other person (Whitney and Smith, 1993).

Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, pages 188–198
Santa Fe, USA, August 25, 2018.

aggression in face-to-face conversation implies yelling, swearing, insulting and dominant attitude, while convert aggression refers to gossiping, inappropriate sarcasm and non-constructive criticism. The typology is arguably challenging when it comes to applying it to online communication, as some of the nuances required to recognize aggression are difficult to discern from text alone. The task turned out to be challenging for human annotators, even though they could refer to context of each message. Expectedly, the task turned out to be even more challenging for automated systems.

In this paper we describe our submissions to the shared task. We tackled the task using traditional (shallow) machine learning models, namely logistic regression and support vector machine (SVM), as well as deep learning models, namely convolutional neural networks (CNNs) and long short-term memory networks (LSTMs). To get the best of both worlds, we experimented with a combination of shallow and the deep learning models. We achieved respectable performance, ranking 4th and 8th out of 31 teams on the Facebook and Twitter test sets, respectively.

The rest of this paper is structured as follows. In Section 2 we give a brief overview of existing work on aggression detection and related tasks. Section 3 presents the data set, while the machine learning models we use are explained in Section 4. In Section 5 we present and discuss the results, followed by a conclusion and ideas for future improvements in Section 6.

2 Related Work

Aggressive speech i.e., abusive language on the web, comes in many flavours, including racism, sexism, homophobia, trolling, cyberbullying etc. Waseem et al. (2017) proposed a typology for various sub-types of abusive language. Similarly, an annotation schema for socially unacceptable discourse practices was proposed by Fišer et al. (2017). Focusing on microblogging, Founta et al. (2018) propose a characterization of abusive behaviour on Twitter. There is a considerable body of work dealing with detecting various types of abusive language; a good overview can be found in (Schmidt and Wiegand, 2017). While distinguishing between aggressive and non-aggressive speech is a already a challenging task, distinguishing between different subtypes of aggressive speech is of course even more difficult. This was observed for the case of open vs. covert aggression by Malmasi and Zampieri (2018), and served as primary motivation for this shared task (Kumar et al., 2018).

A separate issue, which further increases the difficulty of the task, is that there is no universally agreed-upon definition of aggressive speech – a situation which negatively affects the reliability of annotated data. A study by Ross et al. (2016) has shown that supplying hate-speech definitions to annotators can better align their view with the definition, but this does not positively affect annotation reliability. A related study by Waseem (2016) indicated that having expert knowledge during annotation can result in less annotation effort, but this does not necessarily lead to overall better prediction models.

Initial studies dealing with the detection of aggressive speech (or its many subtypes) rely on traditional text classification techniques, such as the naive Bayes classifier (Kwok and Wang, 2013; Chen et al., 2012; Dinakar et al., 2011), logistic regression (Waseem and Hovy, 2016; Davidson et al., 2017; Wulczyn et al., 2017; Burnap and Williams, 2015), or support vector machines (SVM) (Xu et al., 2012; Dadvar et al., 2013; Schofield and Davidson, 2017). For example, the work of Van Hee et al. (2015) focuses on classifying different subtypes of cyberbullying using an SVM. In a similar vein, Malmasi and Zampieri (2017) built a system to discern between hate speech and mere profanity using a combination of traditional models. Recent work also features dictionary based approaches for detecting abusive language in languages other than English (Tulkens et al., 2016; Mubarak et al., 2017). Other approaches focus on users instead of single texts, such as the work of Ribeiro et al. (2018), where the goal was to determine which social networks users resort to hate speech. They employed gradient boosting, adaptive boosting, and a semi-supervised learning method. Notable is also the work of Nobata et al. (2016), who employ a rich feature set and the regression model from Vowpal Wabbit (Langford et al., 2007), which outperformed even deep learning-based models.

In recent years deep learning-based approaches have become increasingly popular for this task. Pitsilis et al. (2018) worked on detecting offensive language in tweets using LSTM, while Gambäck and Sikdar (2017) used CNN for hate speech classification. A CNN model was also used by Zhang et al. (2018),

Text	Label
Well said sonu..you have courage to stand against dadagiri of Muslims	OAG
How does inflation react to all the after shocks of this demon...?	NAG
Not good job.....this guis creating a problem n our socacity	CAG
pakistani team a world racod 373 run in 20 over.	NAG
I visited 5 atm but I cont able to withdraw from money..not working..	CAG
Your nation is neither islamic nor humane	OAG

Table 1: An excerpt from the train set (OAG – openly aggressive, CAG – covertly aggressive, NAG – not aggressive).

but in a combination with a gated rectified unit (GRU) layer. The work of Potapova and Gordeev (2016) describes an approach to detect aggressive speech using CNN and random forest separately, but not their combination. Similarly, Gao and Huang (2017) explore a logistic regression model with a rich set of features and a bidirectional LSTM, without combining the models. Djuric et al. (2015) employ a logistic regression layer on top of representations learned on a huge hate-speech data set using paragraph2vec (Le and Mikolov, 2014). Several varieties of recurrent neural networks (RNN) as well as a CNN were also evaluated in (Pavlopoulos et al., 2017), yielding good results. A survey of related work indicates that, while both shallow and deep learning models have been extensively tested on this task, the approaches that build on their combination are few and far between.

Approaches most related to ours explore combinations of traditional and deep learning methods. Badjatiya et al. (2017) focus on detecting racism and sexism using a combination of models from different paradigms, specifically LSTM in combination with gradient boosting. Similarly, Park and Fung (2017) use logistic regression in combination with several variants of CNNs in a two step scenario. The first step distinguishes between abusive and non-abusive texts, while the second step distinguishes between different subtypes of abuse. The work most similar to ours is that of Mehdad and Tetreault (2016), where an SVM metaclassifier is trained on outputs of a variant of SVM- and an RNN-based model.

3 Dataset

The shared task dataset consists of 15,000 Facebook messages (train and validation portion combined), out of which 3,419 are labelled as openly aggressive, 5,297 as covertly aggressive, and 6,284 as not-aggressive. Table 1 lists some examples from the dataset. As mentioned in the introduction, data labeling was a challenging task as there is a certain degree of subjectivity present when determining the aggression type. From our experience, the most problematic are the openly vs. covertly aggressive cases, in particular the messages that contain no swear words, since swear words usually imply open aggression. Their absence, on the other hand, does not mean that the message is not openly aggressive. We also noticed that many of the texts had grammatical and typing errors, typical of user-generated content. Moreover, a smaller number of texts were not in English. Finally, an additional difficulty is posed by the fact that sometimes broader context is required to correctly classify a message. For example, the text from the third row of Table 1 need not necessarily be covertly aggressive; whether this is the case depends on what the speaker is referring to.

For the above reasons, performing this task manually is not trivial at all. This is also reflected in the final scores of machine learning algorithms, which are relatively low – the top performing systems on the shared task attained 0.64 and 0.60 weighted F1 measure on the Facebook and Twitter test sets, respectively.

4 Models

4.1 Experimental setup

The dataset provided by the task organizers had already been split into a train and development part. However, since it is not uncommon for the test data in such competitions to be unrepresentative of train data, we decided to use the official development set as held-out test data for all preliminary experiments to avoid overfitting our models and to obtain a realistic performance estimate.

We first evaluate our models using 5-fold cross-validation on the train data (the official train set). In each iteration of the cross-validation, models are fitted on four out of five folds, and they are used to label the remaining fifth fold, which is also considered a validation fold. Model hyperparameters (details below) are chosen (in each iteration separately) to maximize weighted F1-score on this validation fold, making the obtained cross-validation score an optimistic estimate of true model performance. This also produces five trained models, each trained on 4/5 of the train data. Labels on the held-out data, i.e., the official development set, are derived for each model by voting of these five models, which could be considered a type of bagging. The main motivation for this slightly atypical setup is ensuring that deep learning-based models have access to a validation fold in each iteration for regularization via early stopping.

For producing the official test set labels, we used an identical setup, but used the union of train and development sets as the train data and the official test set as the held-out test data.

4.2 Classification models

In order to classify the messages, we used three base models: logistic regression, CNN, and bidirectional LSTM (BiLSTM). Apart for lowercasing and tokenization, we did not perform any additional preprocessing.[2] We next describe our models.

The first model we considered was logistic regression. We explored three variants, each based on unigrams, bigrams, and character n-grams, respectively. For each of these variants we also include the following additional features:

- *Bad word occurrences* – boolean feature which indicates if the text contains a word from a list of inappropriate and likely offensive words, obtained from the web;[3]
- *POS tags* – number of occurrences for nouns, verbs, adverbs, adjectives, foreign words, and cardinal numbers (a total of six numerical features). We extract the counts using the POS tagger from NLTK (Loper and Bird, 2002);
- *Text length* in both characters and tokens (two numerical features);
- *Capitalization features* – the number of words that are capitalized and the number of words written in all caps (two numerical features);
- *Numerical tokens* – the number of tokens that represent a number;
- *Named entities* – the number of named entities; three numerical features corresponding to counts of named entities of type *person*, *organization* and *location*, respectively. We used the named entity recognizer (NER) from NLTK (Loper and Bird, 2002) to extract the named entities;
- *Sentiment polarity* – a single numerical feature indicating sentiment polarity of the text. We used the VADER (Gilbert, 2014) sentiment analysis system[4] to predict the sentiment.

Although the final result did not change much after adding all of these features, we did observe minor improvements. The only hyperparameter which was adjusted for this model is the inverse regularization strength C, for which the search interval was $\{2^{-}15, 2^{-}14, ..., 2^{5}\}$. The measure that was maximized was weighted F1-score on the validation fold of the train set. It is worth mentioning that we also tried a linear SVM instead of logistic regression and got very similar results with a somewhat longer training time.

[2] We did try to expand contractions (e.g., *can't* → *can not*) and remove non alphanumeric tokens, but this did not have a significant positive impact.

[3] `https://www.cs.cmu.edu/~biglou/resources/bad-words.txt`.

[4] Available online at `https://github.com/cjhutto/vaderSentiment`.

For deep learning models, we tried using a CNN and a BiLSTM network architectures. Inputs to both models were GloVe (Pennington et al., 2014) 300-dimensional word embeddings trained on 840 billion tokens from the Common Crawl or 200-dimensional word embeddings trained on 20 billion tweets.[5] Since the train and dev data are from Facebook, we believed the Twitter-based embeddings might fare better, as the messages from Facebook and tweets should be similar. For this reason, we tried both types of embeddings in our experiments, yielding two variants of each deep learning model. For the BiLSTM we used 100 units as the size of the hidden state and sigmoid transfer functions. The output of the BiLSTM layer is fed into a fully connected layer with three output neurons and a softmax transfer function. Both dropout and recurrent dropout hyperparameters were set to 0.2. Our CNN model used a single convolution layer containing 50 filters of width 3 and 30 filters of width 5, using the ReLU transfer function. This layer was followed by a max pooling layer and a dropout layer with a dropout rate of 0.5. Similarly as for the LSTM, the output of this layer was fed into a fully connected layer with three output neurons and a softmax transfer function. Both BiLSTM and CNN were trained by minimizing categorical cross-entropy using Adam (Kingma and Ba, 2015), with learning rate values of 0.001 and 0.0005, respectively.[6]

We have also tested voting combination of seven models: three logistic regressions, two LSTMs, and two CNNs. Logistic regressions differed in a way that sentences were vectorized (unigrams, bigrams, and character n-grams) and deep learning models used two different embedings we mentioned. Hard voting was used and ties were resolved by choosing the class that was most frequent in the train data.

Finally, we tried taking predictions of the seven models and feeding them into a metaclassifier. It is worth mentioning that we only used hard predictions for the voting classifier, as opposed to using probabilities of each class for the metaclassifier, although it does not seem that this had a significant impact on the final result. The metaclassifier we used was an SVM with an RBF kernel. Hyperparameters that were tuned were the inverse regularization strength C and the width of the kernel function γ. We optimized these using grid search in the range of $\{0.001, 0.01, 0.1, 1, 2, 4, 8\}$ for both hyperparameters.

5 Results

5.1 Preliminary evaluations

Before submitting the final results to the shared task organizers, we ran a number of preliminary evaluations. The results of our models using both 5-fold cross-validation on the train set as well as on the development set are given in Table 2. In general, results of BiLSTM and logistic regression are comparable, while the standalone models perform slightly worse. Among the standalone models, the best results were obtained with the BiLSTM, which is not surprising since this model is considered by many to be the state of the art. The second-best result was achieved by logistic regression with word bigrams as features, which is an interesting indicator that, although logistic regression is considered to be among the simplest traditional machine learning models, it can yield satisfactory results on this task. Results on the development set were slightly different, and this time logistic regression achieved the best results. Moreover, what we find the most interesting is the fact that both combinations of the models (voting and stacking with an SVM metaclassifier) yield somewhat better results than any of the seven standalone models, indicating that combining models is useful.

5.2 Test set results

For the final testing the task organizers made it possible to submit up to three models. We decided to trust the cross-validation scores more than those on the official development set and chosen (1) BiLSTM with Common Crawl embeddings, (2) logistic regression with bigrams, and (3) SVM trained on predictions of the seven models as our final submissions. Testing was conducted by the task organizers on two data sets – one from Facebook, same as the train data, and another one from Twitter. On the Twitter test set SVM had the best result among our models, as expected. Surprisingly enough, on the Facebook dataset the best results were obtained by the BiLSTM-common model; we investigate the reasons for this in

[5] Both pretrained GloVe embeddings are available online at `https://nlp.stanford.edu/projects/glove/`.

[6] All the hyperparameters were chosen to maximize weighted F1-score on the validation fold of the train data.

System	**F1 cross-validation**	**F1 train/dev**
Random Baseline	0.345	0.341
Logistic regression unigrams	0.560	0.567
Logistic regression char-ngrams	0.566	0.566
Logistic regression bigrams	0.570	0.583
BiLSTM-common	0.585	0.571
BiLSTM-twitter	0.575	0.561
CNN-common	0.570	0.561
CNN-twitter	0.562	0.580
Voting	0.591	0.595
SVM metaclassifier	**0.603**	**0.603**

Table 2: Results using cross-validation on the official train set (the first column) and on the official development set (the second column). The designations next to the deep learning models refer to the type of embeddings they were given as input.

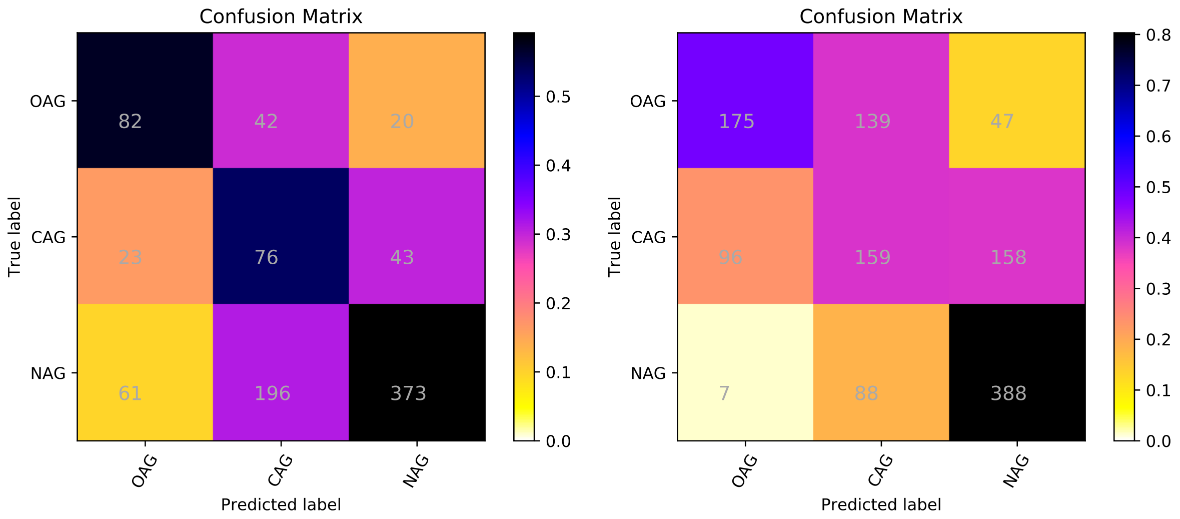
System	**F1 Facebook**	**F1 Twitter**
Random baseline	0.354	0.347
saroyehun	**0.642**	0.592
EBSI-LIA-UNAM	0.632	0.572
DA-LD-Hildesheim	0.618	0.552
TakeLab	0.616	0.565
sreeIN	0.604	0.508
vista.ue	0.581	**0.601**
Julian	0.601	0.599
uOttawa	0.597	0.569

Table 3: Test set results of models which where in top 5 on either dataset. Our (TakeLab) model refers to the the BiLSTM-common model and the SVM metaclassifier model for the Facebook and Twitter test sets, respectively.

the error analysis section below. Table 3 shows the F1-scores of the five best models on each dataset. There is an overlap, since two models were in top 5 on both the first and the second dataset. We ranked fourth on the first dataset and eighth on the second dataset out of 31 competitors. We omit statistical significance tests that compare our system to other systems, as we do not have access to their labels, but instead refer to (Kumar et al., 2018) for additional comparison details.

5.3 Error analysis

We next analyse predictions of our best performing models on the test data, focusing on the erroneous predictions. Figure 1 shows confusion matrices of our best models (BiLSTM-common for the Facebook test set and logistic regression with bigrams for the Twitter test set). The matrices reveal that it was much easier for our model to differ between open aggression and non-aggression than between covert aggression and non-aggression, which is intuitive and in line with findings of Malmasi and Zampieri (2018). Distinguishing between open and covert aggression was also quite challenging, especially on the second dataset. We have also noticed that all labels were distributed more or less evenly over the second dataset, which was the case with our train data as well. On the other hand, most of the sentences from the first dataset were labelled non-aggressive, and our model had a tendency to predict covert aggression when the sentence should have been labelled non-aggressive. There is a chance that this imbalance caused the SVM to perform worse than expected.



(a) Facebook

(b) Twitter

Figure 1: Confusion matrices of our best models. On the the Facebook (left) test set the matrix is for the BiLSTM-common model. On the Twitter test set (right) the matrix is for the SVM metaclassifier model.

Tweet	**True label**	**Voting**	**Predictions**
me and my mate swapnil patel watching bat swapnil patil!! too confused!! #indvsuae	**N**	**C**	C C N N C C N
wickets are falling faster than updates, lol #indvsuae	**C**	**N**	C C N N N N N
its like pikachu vs team rocket today .. #indvsuae	**C**	**N**	N N N N N N N
#shutdownjnu over democratic setup feeds these thugs	**C**	**O**	C C O O O O C
#ShutDownJNU ,it has never been an educational institution,must b handed over to Army to teach them proper lesson.	**O**	**C**	O O C C C C C
If glorifying a terrorist can be jusified under FoS, then you can say just any thing. #ShutDownJNU	**C**	**O**	O O O O O O O
Send Umar Khalid to Afzal before its too late. Don't let him become another Afzal Guru in real! #ShutdownJNU #MadarsaJNU	**O**	**C**	C C C C C C C

Table 4: Tweets, true labels, prediction of the voting classifier, and predictions of all the seven models, where O stands for open aggression, C stands for covert aggression and N for non-aggressive speech.

Table 4 shows several interesting misclassifications made by voting combination of our models, as well as predictions of every standalone model. [7] First three examples show misclassifications in which non-aggression was confused with covert aggression. In the first two examples we can see that some models were correct but the majority voted for the opposite class. Interestingly enough, none of the models predicted open aggression. The third example was classified as non-aggression by all models and we believe some human annotators would probably make the same decision. However, there are also situations where all of our models are in agreement but the label they have assigned is obviously incorrect. The remaining examples show cases where open and covert aggression were confused. Deciding between open and covert aggression would have probably been the most difficult for human annotators, so it is not surprising that this is challenging for machines too. Again, we can see that none of the models predicted non-aggression in this situation. Examples in which open aggression is mistaken for non-aggression are quite rare.

6 Conclusion and Future Work

In this paper we have proposed a machine learning model for the problem of detecting open and covert aggression speech using shallow machine learning models, deep learning, and their combination. Among standalone models, the best score was achieved by a bidirectional long short term memory network (BiLSTM). However, combining BiLSTMs with Convolutional Neural Networks (CNN), and variants of logistic regression via a support vector machine (SVM), that served as a metaclassifier, turned out to offer improvements in some cases. This speaks in favor of the idea that combining deep learning models with traditional ones through voting or stacking might result in the best performance. Even though the potential of deep learning itself is evident and deep learning models often do have exceptionally high standalone scores.

We have chosen logistic regression as the "representative" of traditional (shallow) models because it had good initial results and it was fast and easy to optimize, but our experiments were not sufficiently exhaustive to determine whether it really is the best traditional model for this task. However, the combination of traditional and deep learning paradigms did prove promising. Consequently, an immediate venue of future work would be adding other models into the voting ensemble and the SVM metaclassifier. Another possibility would be to explore approaches that combine tree kernels with word-embeddings derived from deep learning, such as the work described in (Plank and Moschitti, 2013; Kim et al., 2015). We feel another promising direction for future work would be addressing the highly noisy nature of the data. One option would be more rigorous preprocessing, such as using specialized tools for normalization of social media text (Han et al., 2013; Baldwin et al., 2015). Finally, the data set could be improved by adding additional context information, if possible, e.g., other social media posts collocated with the post in question, or posts by the same user at other points in time.

Acknowledgements

This research has been partly supported by the European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS).

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee.

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

[7] The point of the prediction column is not to show what each standalone model has predicted, but to demonstrate what was the voting classifier's dilemma. For the sake of completeness, the order of predictions is: logreg-unigrams, logreg-bigrams, logreg-charngrams, BiLSTM-common, CNN-common, BiLSTM-tweet, CNN-tweet.

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.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 71–80. IEEE.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *Advances in Information Retrieval*, pages 693–696. Springer.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of ICWSM*.

Karthik Dinakar, Roi Reichart, and Henry Lieberman. 2011. Modeling the detection of textual cyberbullying. In *The Social Mobile Web*, pages 11–17.

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.

Darja Fišer, Tomaž Erjavec, and Nikola Ljubešić. 2017. Legal Framework, Dataset and Annotation Schema for Socially Unacceptable On-line Discourse Practices in Slovene. In *Proceedings of the Workshop Workshop on Abusive Language Online (ALW)*, Vancouver, Canada.

Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. *arXiv preprint arXiv:1802.00393*.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*.

CJ Hutto Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5.

Jonghoon Kim, François Rousseau, and Michalis Vazirgiannis. 2015. Convolutional sentence kernel from word embeddings for short text categorization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 775–780.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. pages 1–13.

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

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

John Langford, Lihong Li, and Alex Strehl. 2007. Vowpal wabbit online learning project.

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

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202.

Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303.

Hamdy Mubarak, Darwish Kareem, and Magdy Walid. 2017. Abusive Language Detection on Arabic Social Media. In *Proceedings of the Workshop on Abusive Language Online (ALW)*, Vancouver, Canada.

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*, pages 145–153. International World Wide Web Conferences Steering Committee.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. In *Proceedings of the First Workshop on Abusive Language Online*, pages 41–45.

John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online*, pages 25–35.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1498–1507.

Rodmonga Potapova and Denis Gordeev. 2016. Detecting state of aggression in sentences using cnn. *arXiv preprint arXiv:1604.06650*.

Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. *arXiv preprint arXiv:1803.08977*.

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 *Proceedings of the Workshop on Natural Language Processing for Computer-Mediated Communication (NLP4CMC)*, Bochum, Germany.

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.

Alexandra Schofield and Thomas Davidson. 2017. Identifying Hate Speech in Social Media. *XRDS: Crossroads, The ACM Magazine for Students*, 24(2):56–59.

Stéphan Tulkens, Lisa Hilte, Elise Lodewyckx, Ben Verhoeven, and Walter Daelemans. 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.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Véronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In *International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 672–680.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Langauge Online*.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Irene Whitney and Peter K Smith. 1993. A survey of the nature and extent of bullying in junior/middle and secondary schools. *Educational research*, 35(1):3–25.

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

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

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Lecture Notes in Computer Science*. Springer Verlag.