Aggression Detection in Social Media: Using Deep Neural Networks, Data Augmentation, and Pseudo Labeling

Segun Taofeek Aroyehun
CIC, Instituto Politécnico Nacional
Mexico City, Mexico
aroyehun.segun@gmail.com

Alexander Gelbukh
CIC, Instituto Politécnico Nacional
Mexico City, Mexico
www.gelbukh.com

Abstract

With the advent of the read-write web which facilitates social interactions in online spaces, the rise of anti-social behaviour in online spaces has attracted the attention of researchers. In this paper, we address the challenge of automatically identifying aggression in social media posts. Our team, *saroyehun*, participated in the English track of the Aggression Detection in Social Media Shared Task. On this task, we investigate the efficacy of deep neural network models of varying complexity. Our results reveal that deep neural network models require more data points to do better than an NBSVM linear baseline based on character n-grams. Our improved deep neural network models were trained on augmented data and pseudo labeled examples. Our LSTM classifier receives a weighted macro-F1 score of *0.6425* to rank *first* overall on the Facebook sub-task of the shared task. On the social media sub-task, our CNN-LSTM model records a weighted macro-F1 score of *0.5920* to place *third* overall.

1 Introduction

The read-write web has enabled user-generated content on several online platforms. A major part of these platforms is the social media websites. These websites facilitate social interactions by way of posting comments and replying to comments submitted by other users. As with offline interactions, interactions taking place online are subject to anti-social behaviours such as trolling, flaming, abuse, bullying, and hate speech. With globally increasing internet penetration, which has enabled unprecedented access to the web, the illusion of 'invisibility' by web users has given rise to growing anti-social behaviour in online spaces. Consequently, web users are exposed to mental, psychological, and emotional distress if such behaviour goes unchecked.

In an attempt to make the web more civil, the automatic detection of content that contains or could have the effect of abuse, aggression or hate speech on social media platforms has being an important task. However, most accessible (with publicly available datasets) studies on automatic hate speech detection has focused on Twitter. The shared task on aggression detection (Kumar et al., 2018a) aims to benchmark classifiers developed for the automatic identification aggression in social media platforms by making annotated data collected from Facebook available. The task is to develop a multi-class classifier that could make subtle distinction among texts that belong to one of three categories: overtly aggressive, non-aggressive and covertly aggressive.

Motivated by existing work that applies machine learning to the task of automatic hate speech detection, we investigate the effectiveness of models that rely on little or no feature engineering. The presence of noisy content such as misspellings, acronyms, code-mixing, and incorrect grammar in social media posts makes the extraction of linguistic features using Natural Language Processing (NLP) tools highly error-prone. As a result, we experimented with linear (NBSVM with n-grams) and deep learning models (CNN, LSTM, BiLSTM, and combinations thereof). The linear model serves as our baseline (hard to beat) while we refine our deep learning models for this task. We aim to examine (1) the relationship between model complexity and effectiveness and (2) the dataset requirements in order to accurately detect aggression in social media posts.

Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying, pages 90–97
Santa Fe, USA, August 25, 2018.

In the rest of this paper, Section 2 outlines related work. Sections 3 and 4 present the data and our methodology respectively. The results of our experiments are in Section 5 and we conclude in Section 6.

2 Related Work

The task of aggression detection in social media can be considered as a document classification task. This task can also be sub-divided into binary classification or multi-class classification. In the context of detection of aggressiveness, the binary classification would imply the presence or absence of some anti-social phenomena such as abuse (Nobata et al., 2016) in a given example (abusive or not abusive). Whereas in the multi-class scenario, specific types of anti-social behaviour are of interest (Waseem et al., 2017) such as racism, sexism, hate speech, and bullying. It has been observed that contents which contain anti-social phenomena are rare in a collection of social media posts. This usually leads to imbalanced datasets where posts which lack the phenomena of interest are overwhelming. This problem is even more pronounced in the multi-class scenario which leads to difficulty in learning discriminative features by classifiers. In (Malmasi and Zampieri, 2018), it was concluded that subtle distinction between types of anti-social behaviour: profanity and hate speech is a difficult task for machine learning classifiers. By extension, it will be difficult to differentiate between overt and covert aggression. Also, (Davidson et al., 2017) submits that posts that does not contain explicit aggressive words are likely to be difficult to identify. This would be likely applicable to the covertly aggressive class in this study.

Acknowledging the cost of annotating data for hateful comments in social media, (Gao et al., 2017) proposed a system that leverages the availability of unlabeled data. Their result shows improvement over systems that rely only on manually annotated data. This approach is most related to our work.

The identification of aggression in social media is closely related to existing studies in hate speech, abuse, and cyberbullying detection. Methods used to tackle these tasks as supervised classification broadly falls into two. One approach is based on manual feature engineering. With the feature engineering approach, extracted features serve as input to classic machine learning algorithms such as naive bayes, logistic regression, support vector machines, and random forest (Schmidt and Wiegand, 2017; Malmasi and Zampieri, 2017). The other approach is based on deep neural networks that automatically learn features from input data. (Gambäck and Sikdar, 2017) employed convolutional neural networks to classify hate speech and (Zhang et al., 2018) used a combination of convolutional neural network and gated recurrent unit (GRU) for the same task.

Rather than rely only on the textual content of social media posts in identifying hate speech, (Qian et al., 2018; Founta et al., 2018) investigated the use of metadata about the users or the posts. However, the metadata may not be readily available from the social media platforms. In addition, this approach may breakdown where a social media platform allows anonymous posting.

3 Data

In this section, we describe the datasets used in this work and the data augmentation strategy. Also, the details of the pseudo labeling approach is presented.

Dataset We used the annotated data provided by the task organizers. The details of the annotation procedure is in (Kumar et al., 2018b). The dataset consists of posts collected from Facebook. The posts are related to entities (organizations, individuals or issues) in India. The dataset covers both English and Hindi. We used the English part in this work. The dataset provides three high-level tagsets namely: covertly aggressive, non-aggressive, and overtly aggressive. Each post is annotated with one of these tagsets. The English dataset consists of 12000 training examples, 3001 development examples, 916 test examples from Facebook, and 1257 examples from an undisclosed social media platform. On the training set, the number of examples per class is unbalanced. The covertly aggressive class is approximately 34% of the training examples; non-aggressive class is approximately 42% of the training examples and overtly aggressive class is approximately 24% of the training examples. The minimum and maximum number of tokens on the training set are 1 and 1200 respectively. We observed that the training set contains redundant posts: different post ID but duplicate content.

Data Preprocessing We used well-formed texts in our experiments. We lowercased all alphabetic characters. We removed punctuation, digits, URL, repeated characters, non-English characters, alphanumerics, and usernames. In addition, we decoded emoji(emoticons) into their text equivalents and transformed hashtags into their constituent tokens where possible. To address misspellings, we used a spell-checker.

Data Augmentation and Pseudo Labeling A common technique to improve model generalization in computer vision is to enlarge the dataset using label-preserving transformation(s) (Simard et al., 2003). This is usually achieved via known invariant operations such as rotation and scaling of images in the training set. Inspired by this technique, we simulate transformations by translating each example to an intermediate language and back to English. We translated into four intermediate languages: French, Spanish, German, and Hindi. We observed marginal increase in model performance using the four languages. Machine translation is known to yield conceptually equivalent output in a target language. We used the Google Translate API for the translation. We rely on a weak notion of label-preserving transformation for text by translating the original training text to an intermediate language and backtranslate to English. We hypothesize that the translation into several intermediate languages will help diversify our augmented training set in terms of different lexical choice using each of the intermediate languages as a source language and English as the target language in the backtranslation phase. Even though the process of translation and backtranslation is error prone, this procedure increases the size of our training set by a factor of 5. The resulting augmented training set is highly interdependent. With this technique, our DNN models outperform the linear baseline model.

One of the test sets provided by the shared task organizers for evaluation is data collected from an undisclosed social media platform. In order to diversify our training examples, we used a related dataset collected from Twitter for hate speech detection which is publicly available [1]. There are two datasets with a total of 22075 tweets (subject to change based on availability on the Twitter platform). We hypothesize that the dataset contains the phenomenon we are interested in. So, we used the model with the highest performance on the development dataset to label the Twitter dataset. The examples with the pseudo labels were added to the original training dataset and our deep learning models were retrained.

4 Methodology

Our approach was to develop a baseline model and a number of deep neural network models. The models as well as our submissions are presented in this section.

Baseline Model A support vector machine (SVM) model that uses naive bayes (NB) log-count ratio features (NBSVM) is proposed in (Wang and Manning, 2012). NBSVM demonstrates consistent performance across text classification tasks. We used the logistic regression classifier in place of the SVM. We consider this model a strong linear baseline. We implemented two variants of the NBSVM model. One based on word n-grams and the other based on character n-grams. We found the model based on character n-grams superior on the development dataset for aggression detection. The character n-grams NBSVM serves as our baseline based on which we compare our deep neural network models during our experiments.

Input Representation The representation of inputs to the deep neural network models is the embedding layer. The embedding layer encode each word in the vocabulary used in the model. We experimented with different pre-trained word vectors for the embedding layer including word2vec, Glove, SSWE, and fastText. We observed that the coverage of the pre-trained embeddings are limited. FastText has the highest vocabulary coverage in our experiment (with about 5000 missing entires). Thus, instead of using a pre-trained word vector file to look up the vector representation for each word, we used the fastText pre-trained model (Mikolov et al., 2018) to infer the vector representation for each word in the vocabulary of our model. This allows us to use the sub-word level information to derive vector representation for words not in the vocabulary of the corpus (Wikipedia) on which the fastText model was pre-trained. This facility is especially important for social media posts which usually contains typos and

[1] https://github.com/ZeeraKW/hatespeech

abbreviations. In addition, we used a randomly initialized embedding which is trained with the network to learn task-specific word representation. So, the input representation to our deep neural network models is a concatenation of 300 dimensional word vectors derived from fastText model and a 50 dimensional task-specific word vector.

Deep Neural Network Models We experimented with seven deep learning models for aggression detection: CNN, LSTM, BiLSTM, CNN-LSTM, LSTM-CNN, CNN-BiLSTM, and BiLSTM-CNN. The models are of varying complexity, with the combination of BiLSTM and CNN: CNN-BiLSTM and BiLSTM-CNN being the most complex neural architecture. CNNs have being used for text classification (Kim, 2014). CNN is able to learn features from words or phrases in different positions in the text. LSTMs model long-term dependencies in text and has been found to be useful for text classification . BiLSTMs consists of two LSTMs. One encode information in a sequence in the forward direction and the other in the backward direction. The past and future information is available to the model in making classification decisions. CNNs and (Bi)LSTMs are complementary in their modeling capabilities. Each of CNNs and (Bi)LSTMs capture information at different scales. As such, we explored whether there could be potential improvements in combining multiple information at different scales for aggression detection. We explored passing the input representation into the CNN and feed the local features learnt by the CNN into the (Bi)LSTM in the CNN-(Bi)LSTM model. Conversely, we also passed the input representation into the (Bi)LSTM and fed the long-term features learnt by the (Bi)LSTM into the CNN in the (Bi)LSTM-CNN model. The representations learnt by the CNN, (Bi)LSTM, and combinations thereof serve as input to the fully-connected layer. The output of the fully-connected layer is a softmax output (probability) which indicates the chance of belonging to each of the three classes. The class with the highest probability is the predicted class.

In training our deep neural network models, we used the sparse categorical cross-entropy as our objective function. We chose the RMSprop optimizer to minimize the loss function through backpropagation within 5 epochs. We used feature dropout and earlystopping as regularization mechanisms to avoid overfitting. A spatial dropout (Tompson et al., 2015) probability of 0.2 was applied to the embedding layer.

Submissions Our team, 'saroyehun' submitted three systems for evaluation on the unseen test dataset. Our first submission is the predictions obtained from the LSTM model. This model was trained using the augmented dataset. The second submission uses the representations learnt by the CNN and LSTM in the CNN-LSTM set-up. This submission was trained on the larger training set consisting of augmented and pseudo labeled examples. In addition, we computed sentiment score for each example and used the score as an additional feature to the representations learnt by the CNN-LSTM layers before making predictions. The third submission is an ensemble of the predictions made by our deep neural network models trained on (1) augmented data and (2) combination of augmented and pseudo labeled data, and sentiment score for each post. The final predictions was obtained by majority voting.

System	F1 (weighted)	F1 (weighted)$^{+}$	F1 (weighted)$^{++}$
NBSVM	**0.5116**	0.5135	–
CNN	0.4989	0.5520	0.5741
LSTM	0.4940	**0.5679**	0.5561
BiLSTM	0.4714	0.5274	0.5776
CNN-LSTM	0.4702	0.5296	**0.5822**
LSTM-CNN	0.3679	0.5624	0.5729
CNN-BiLSTM	0.3839	0.5272	0.5573
BiLSTM-CNN	0.4836	0.5523	0.5440

Table 1: Weighted Macro-F1 Scores on the English Development set
$^{+}$: Data Augmentation
$^{++}$: Data Augmentation and Pseudo Labeling

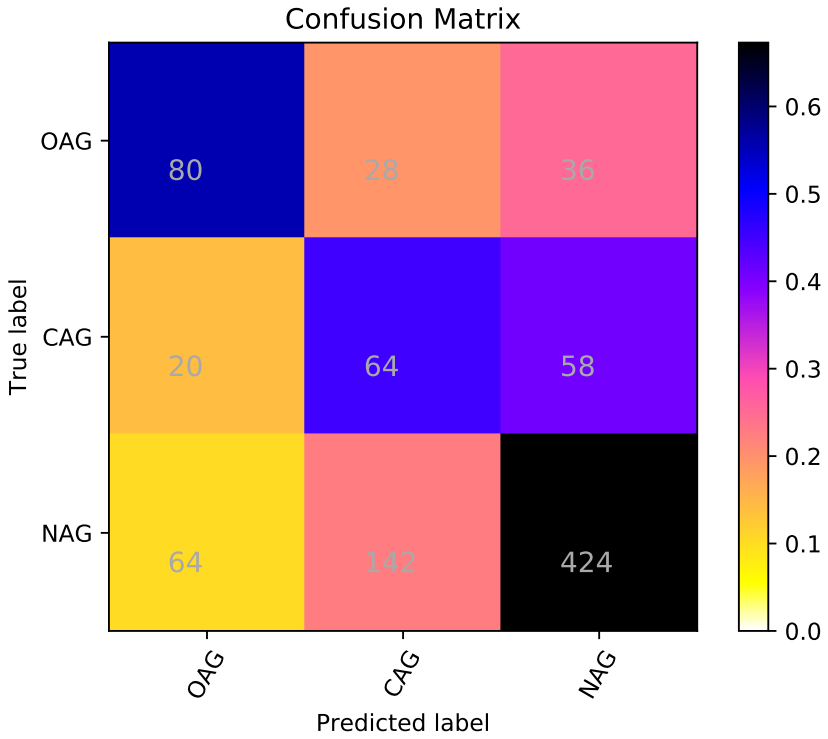

Figure 1: Confusion Matrix of the LSTM Predictions on the English (Facebook) Test Set

5 Results

This section shows the results obtained on the English development and the test datasets (Facebook and Social Media) in terms of weighted macro-F1 scores.

In Table 1, the performance of the baseline model (NBSVM) and the DNN models are presented. Using the training dataset as given for training, the NBSVM model achieves the highest performance. The table shows the impact of data augmentation and pseudo labeling on the models. It can be observed that using data augmentation, the performance gain on the NBSVM is very minimal (0.0019) compared to the DNN models where the maximum performance gain of 0.1433 is observed on the LSTM model. The combination of data augmentation and pseudo labeling results in slight performance improvement on the DNN models except for the LSTM model. Overall, data augmentation results in approximately 5% weighted macro-F1 improvement on the development set over the NBSVM baseline model. Also, the combination of data augmentation and pseudo labeling yields about 7% weighted macro-F1 gain over the linear baseline model. Pseudo labeling gives a marginal improvement of about 2%. The substantial gain in performance (about 5%) is from the introduction of data augmentation. These gains demonstrate the effectiveness of the data augmentation approach and complementarity of pseudo labeling in our approach. In addition, we experimented with and without the sentiment score as a feature. However, we did not observe any substantial gain in performance.

Table 2 shows the performance of the outputs of the three systems we submitted for evaluation on the Facebook test set. All our submissions are better than the random baseline score provided by the task organizers. All our systems exhibit comparable performance; they are all within 0.05 F1 score of each other. The LSTM model trained on the augmented training set achieved the best weighted macro-F1 score of **0.6425**. This score ranks **first** on the shared task which attracted **30** submissions. It is clear that a complex model (CNN-LSTM) does not necessarily give the best result. Figure 1 shows the confusion matrix for our best system, the LSTM model. From the confusion matrix, one can see that the most mistakes made by the model is on the non-aggressive class (NAG) in absolute terms; the model predicts covertly aggressive (CAG) when the true label is NAG. However, in relative terms, the most difficult

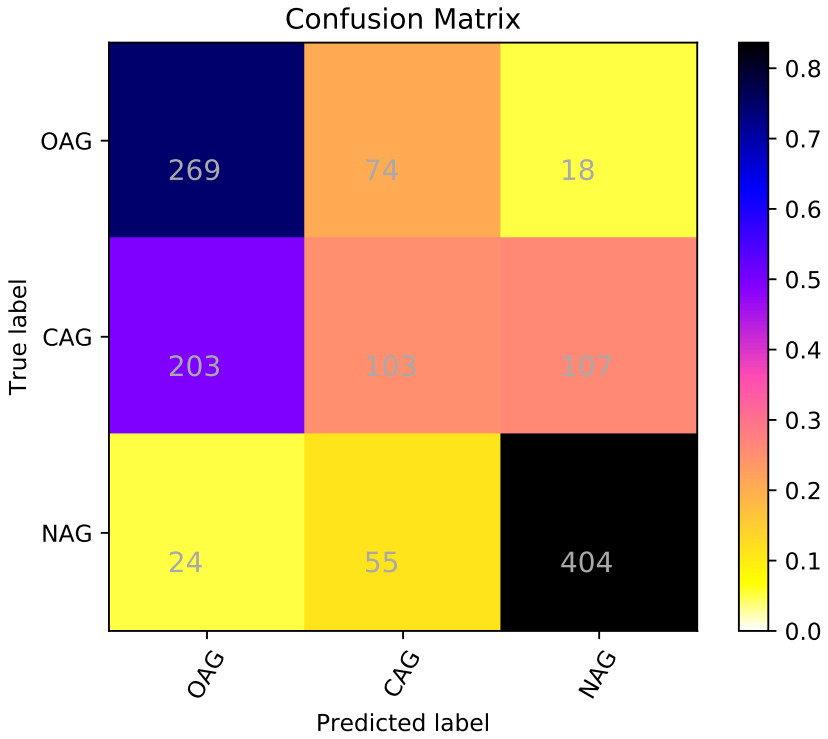

Figure 2: Confusion Matrix of the CNN-LSTM Predictions on the English (Social Media) Test Set

class for the model is the CAG. Here, the most mistakes result from the model predicting NAG where the true labels are CAG. In addition, one can see that the model performs best on the NAG class followed by the overtly aggressive class (OAG) and the least performance is recorded on the CAG class.

System	**F1 (weighted)**
Random Baseline	0.3535
LSTM	**0.6425**
CNN-LSTM	0.6058
Ensemble	0.5897

Table 2: Weighted Macro-F1 Scores on the English (Facebook) Test set

Table 3 outlines the scores achieved by our systems on the surprise test dataset collected from an unnamed social media platform. On this dataset, our CNN-LSTM model trained on the combination of the augmented training data, and pseudo labeled Twitter data as well as sentiment score as a feature received our best weighted macro-F1 score of **0.5920**. This score places the system on an overall ranking of **third** among 30 systems that participated in this track. In this case, a complex model aided by an out-of-domain data is superior. In figure 2, the confusion matrix gives us an insight into how the CNN-LSTM model performs on each class. It can be observed that the model made the most mistakes on the CAG class; predicting OAG when the true class is CAG. The model performs best on the NAG class followed by the OAG class and records the least performance on the CAG class.

Going by the performance trend observed in our results, the performance of our best model for each track seems to reflect the distribution of examples per class in the original training set. One can see that our models show better performance on classes with more training examples compared with classes having lesser training examples. Also, it is unsurprising that even though the OAG class has the least number of examples, the performance of our models on the OAG class is better than on the CAG class which has more examples. The performance of the OAG class validates a previous finding that a post which explicitly contains aggressive word(s) is easy to classify. Given our result, it is obvious that the

System	F1 (weighted)
Random Baseline	0.3477
LSTM	0.5400
CNN-LSTM	**0.5920**
Ensemble	0.5682

Table 3: Weighted Macro-F1 Scores on the English (Social Media) Test set

finding is true even when the category containing explicit word(s) is the minority class.

6 Conclusion

In this paper, we address the challenge of automatically detecting aggression in social media posts. We developed a linear baseline classifier using NBSVM with character n-grams as features. We conducted experiments with deep neural network (DNN) models of varying complexity ranging from CNN, LSTM, BiLSTM, CNN-LSTM, LSTM-CNN, CNN-BiLSTM to BiLSTM-CNN. In order to do better than our linear baseline using deep neural networks, we experimented with data augmentation, pseudo labeling, and sentiment score as a feature. We observed the greatest improvement from our data augmentation strategy. The improved DNN models were better than the linear baseline model on the development set. Hence, for this task the DNN models require more data points than a non-DNN model (in this case NBSVM). As our results show, a complex model does not necessarily bring performance improvement on this task. The classifier based on LSTM received the best weighted macro-F1 score on the English Facebook task. With a score of 0.6425 on the Facebook test set we rank first out of 30 submissions. The CNN-LSTM classifier receives a weighted macro-F1 score of 0.5920 on the social media test set. With this score, we achieved an overall ranking of third out of 30 teams that participated in the social media track.

The performance trend of our models on the three classes shows that for both tracks, our models performed best on the NAG class and least on the CAG class. This is consistent with previous work on the nature of implicit offensive language. The trend reflects the distribution of examples per class in the training set provided for this task. We see a better performance on classes with more training examples and lower performance otherwise. A natural avenue for future work is the investigation of approaches to improve performance on classes with minimal number of training examples.

Acknowledgement

The last author acknowledges the support of the Mexican government via CONACYT (SNI) and the Instituto Politécnico Nacional grant SIP-20181792.

References

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

Antigoni-Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. 2018. A unified deep learning architecture for abuse detection. *CoRR*, abs/1802.00385.

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, Alexis Kuppersmith, and Ruihong Huang. 2017. Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 774–782, Taipei, Taiwan.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar.

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.

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*, 30:1–16.

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)*.

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.

Jing Qian, Mai ElSherief, Elizabeth M Belding, and William Yang Wang. 2018. Leveraging intra-user and inter-user representation learning for automated hate speech detection. *arXiv preprint arXiv:1804.03124*.

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.

Patrice Y Simard, David Steinkraus, John C Platt, et al. 2003. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, pages 958–962.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.

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*.

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.