Comparative Studies of Detecting Abusive Language on Twitter

Younghun Lee* **Seunghyun Yoon*** **Kyomin Jung**
Dept. of Electrical and Computer Engineering
Seoul National University, Seoul, Korea
younggnse@gmail.com {mysmilesh,kjung}@snu.ac.kr

Abstract

The context-dependent nature of online aggression makes annotating large collections of data extremely difficult. Previously studied datasets in abusive language detection have been insufficient in size to efficiently train deep learning models. Recently, *Hate and Abusive Speech on Twitter*, a dataset much greater in size and reliability, has been released. However, this dataset has not been comprehensively studied to its potential. In this paper, we conduct the first comparative study of various learning models on *Hate and Abusive Speech on Twitter*, and discuss the possibility of using additional features and context data for improvements. Experimental results show that bidirectional GRU networks trained on word-level features, with Latent Topic Clustering modules, is the most accurate model scoring 0.805 F1.

1 Introduction

Abusive language refers to any type of insult, vulgarity, or profanity that debases the target; it also can be anything that causes aggravation (Spertus, 1997; Schmidt and Wiegand, 2017). Abusive language is often reframed as, but not limited to, offensive language (Razavi et al., 2010), cyberbullying (Xu et al., 2012), othering language (Burnap and Williams, 2014), and hate speech (Djuric et al., 2015).

Recently, an increasing number of users have been subjected to harassment, or have witnessed offensive behaviors online (Duggan, 2017). Major social media companies (i.e. Facebook, Twitter) have utilized multiple resources—artificial intelligence, human reviewers, user reporting processes, etc.—in effort to censor offensive language, yet it seems nearly impossible to successfully resolve the issue (Robertson, 2017; Musaddique, 2017).

The major reason of the failure in abusive language detection comes from its subjectivity and context-dependent characteristics (Chatzakou et al., 2017). For instance, a message can be regarded as harmless on its own, but when taking previous threads into account it may be seen as abusive, and vice versa. This aspect makes detecting abusive language extremely laborious even for human annotators; therefore it is difficult to build a large and reliable dataset (Founta et al., 2018).

Previously, datasets openly available in abusive language detection research on Twitter ranged from 10K to 35K in size (Chatzakou et al., 2017; Golbeck et al., 2017). This quantity is not sufficient to train the significant number of parameters in deep learning models. Due to this reason, these datasets have been mainly studied by traditional machine learning methods. Most recently, Founta et al. (2018) introduced *Hate and Abusive Speech on Twitter*, a dataset containing 100K tweets with cross-validated labels. Although this corpus has great potential in training deep models with its significant size, there are no baseline reports to date.

This paper investigates the efficacy of different learning models in detecting abusive language. We compare accuracy using the most frequently studied machine learning classifiers as well as recent neural network models.[1] Reliable baseline results are presented with the first comparative study on this dataset. Additionally, we demonstrate the effect of different features and variants, and describe the possibility for further improvements with the use of ensemble models.

2 Related Work

The research community introduced various approaches on abusive language detection. Razavi

* Equal contribution.

[1] The code can be found at: https://github.com/younggns/comparative-abusive-lang

Proceedings of the Second Workshop on Abusive Language Online (ALW2), pages 101–106
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics

et al. (2010) applied Naïve Bayes, and Warner and Hirschberg (2012) used Support Vector Machine (SVM), both with word-level features to classify offensive language. Xiang et al. (2012) generated topic distributions with Latent Dirichlet Allocation (Blei et al., 2003), also using word-level features in order to classify offensive tweets.

More recently, distributed word representations and neural network models have been widely applied for abusive language detection. Djuric et al. (2015) used the Continuous Bag Of Words model with paragraph2vec algorithm (Le and Mikolov, 2014) to more accurately detect hate speech than that of the plain Bag Of Words models. Badjatiya et al. (2017) implemented Gradient Boosted Decision Trees classifiers using word representations trained by deep learning models. Other researchers have investigated character-level representations and their effectiveness compared to word-level representations (Mehdad and Tetreault, 2016; Park and Fung, 2017).

As traditional machine learning methods have relied on feature engineering, (i.e. n-grams, POS tags, user information) (Schmidt and Wiegand, 2017), researchers have proposed neural-based models with the advent of larger datasets. Convolutional Neural Networks and Recurrent Neural Networks have been applied to detect abusive language, and they have outperformed traditional machine learning classifiers such as Logistic Regression and SVM (Park and Fung, 2017; Badjatiya et al., 2017). However, there are no studies investigating the efficiency of neural models with large-scale datasets over 100K.

3 Methodology

This section illustrates our implementations on traditional machine learning classifiers and neural network based models in detail. Furthermore, we describe additional features and variant models investigated.

3.1 Traditional Machine Learning Models

We implement five feature engineering based machine learning classifiers that are most often used for abusive language detection. In data preprocessing, text sequences are converted into Bag Of Words (BOW) representations, and normalized with Term Frequency-Inverse Document Frequency (TF-IDF) values. We experiment with word-level features using n-grams ranging from 1 to 3, and character-level features from 3 to 8-grams. Each classifier is implemented with the following specifications:

Naïve Bayes (NB): Multinomial NB with additive smoothing constant 1
Logistic Regression (LR): Linear LR with L2 regularization constant 1 and limited-memory BFGS optimization
Support Vector Machine (SVM): Linear SVM with L2 regularization constant 1 and logistic loss function
Random Forests (RF): Averaging probabilistic predictions of 10 randomized decision trees
Gradient Boosted Trees (GBT): Tree boosting with learning rate 1 and logistic loss function

3.2 Neural Network based Models

Along with traditional machine learning approaches, we investigate neural network based models to evaluate their efficacy within a larger dataset. In particular, we explore Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and their variant models. A pre-trained GloVe (Pennington et al., 2014) representation is used for word-level features.

CNN: We adopt Kim's (2014) implementation as the baseline. The word-level CNN models have 3 convolutional filters of different sizes [1,2,3] with ReLU activation, and a max-pooling layer. For the character-level CNN, we use 6 convolutional filters of various sizes [3,4,5,6,7,8], then add max-pooling layers followed by 1 fully-connected layer with a dimension of 1024.

Park and Fung (2017) proposed a HybridCNN model which outperformed both word-level and character-level CNNs in abusive language detection. In order to evaluate the HybridCNN for this dataset, we concatenate the output of max-pooled layers from word-level and character-level CNN, and feed this vector to a fully-connected layer in order to predict the output.

All three CNN models (word-level, character-level, and hybrid) use cross entropy with softmax as their loss function and Adam (Kingma and Ba, 2014) as the optimizer.

RNN: We use bidirectional RNN (Schuster and Paliwal, 1997) as the baseline, implementing a GRU (Cho et al., 2014) cell for each recurrent unit.

From extensive parameter-search experiments, we chose 1 encoding layer with 50 dimensional hidden states and an input dropout probability of 0.3. The RNN models use cross entropy with sigmoid as their loss function and Adam as the optimizer.

For a possible improvement, we apply a self-matching attention mechanism on RNN baseline models (Wang et al., 2017) so that they may better understand the data by retrieving text sequences twice. We also investigate a recently introduced method, Latent Topic Clustering (LTC) (Yoon et al., 2018). The LTC method extracts latent topic information from the hidden states of RNN, and uses it for additional information in classifying the text data.

3.3 Feature Extension

While manually analyzing the raw dataset, we noticed that looking at the tweet one has replied to or has quoted, provides significant contextual information. We call these, *"context tweets"*. As humans can better understand a tweet with the reference of its context, our assumption is that computers also benefit from taking context tweets into account in detecting abusive language.

As shown in the examples below, (2) is labeled abusive due to the use of vulgar language. However, the intention of the user can be better understood with its context tweet (1).

(1) I hate when I'm sitting in front of the bus and somebody with a wheelchair get on.

↳ (2) I hate it when I'm trying to board a bus and there's already an as**ole on it.

Similarly, context tweet (3) is important in understanding the abusive tweet (4), especially in identifying the target of the malice.

(3) Survivors of #Syria Gas Attack Recount 'a Cruel Scene'.

↳ (4) Who the HELL is "LIKE" ING this post? Sick people....

Huang et al. (2016) used several attributes of context tweets for sentiment analysis in order to improve the baseline LSTM model. However, their approach was limited because the meta-information they focused on—author information, conversation type, use of the same hashtags or emojis—are all highly dependent on data.

In order to avoid data dependency, text sequences of context tweets are directly used as an additional feature of neural network models. We use the same baseline model to convert context tweets to vectors, then concatenate these vectors with outputs of their corresponding labeled tweets. More specifically, we concatenate max-pooled layers of context and labeled tweets for the CNN baseline model. As for RNN, the last hidden states of context and labeled tweets are concatenated.

Labels	Normal	Spam	Hateful	Abusive
Number (%)	42,932 (60.5)	9,757 (13.8)	3,100 (4.4)	15,115 (21.3)

Table 1: Label distribution of crawled tweets

4 Experiments

4.1 Dataset

Hate and Abusive Speech on Twitter (Founta et al., 2018) classifies tweets into 4 labels, *"normal"*, *"spam"*, *"hateful"* and *"abusive"*. We were only able to crawl 70,904 tweets out of 99,996 tweet IDs, mainly because the tweet was deleted or the user account had been suspended. Table 1 shows the distribution of labels of the crawled data.

4.2 Data Preprocessing

In the data preprocessing steps, user IDs, URLs, and frequently used emojis are replaced as special tokens. Since hashtags tend to have a high correlation with the content of the tweet (Lehmann et al., 2012), we use a segmentation library[2] (Segaran and Hammerbacher, 2009) for hashtags to extract more information.

For character-level representations, we apply the method Zhang et al. (2015) proposed. Tweets are transformed into one-hot encoded vectors using 70 character dimensions—26 lower-cased alphabets, 10 digits, and 34 special characters including whitespace.

4.3 Training and Evaluation

In training the feature engineering based machine learning classifiers, we truncate vector representations according to the TF-IDF values (the top 14,000 and 53,000 for word-level and character-level representations, respectively) to avoid overfitting. For neural network models, words that appear only once are replaced as unknown tokens.

[2] WordSegment module description page: https://pypi.org/project/wordsegment/

	Normal			**Spam**			**Hateful**			**Abusive**			**Total**		
Model	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NB (word)	.776	.916	.840	.573	.378	.456	.502	.034	.063	.828	.744	.784	.747	.767	.741
NB (char)	.827	.805	.815	.467	**.609**	.528	.452	.061	.107	.788	.832	.803	.752	.751	.744
LR (word)	.807	.933	.865	.616	.365	.458	.620	.161	.254	.868	.844	.856	.786	.802	.780
LR (char)	.808	.934	.866	.618	.363	.457	.636	.183	.283	**.873**	.848	.860	.788	.804	.783
SVM (word)	.757	.967	.850	**.678**	.190	.296	**.836**	.034	.065	.865	.757	.807	.773	.775	.730
SVM (char)	.763	**.968**	.853	**.680**	.198	.306	**.805**	.070	.129	**.876**	.775	.822	.778	.781	.740
RF (word)	.776	.945	.853	.581	.213	.311	.556	.109	.182	.852	.819	.835	.757	.781	.745
RF (char)	.793	.934	.857	.568	.252	.349	.563	.150	.236	.853	.856	.854	.765	.789	.760
GBT (word)	.806	.921	.860	.581	.320	.413	.506	.194	.279	.854	.863	.858	.772	.794	.773
GBT (char)	.807	.913	.857	.560	.346	.428	.472	.187	.267	.859	.859	.859	.770	.791	.772
CNN (word)	.822	.925	.870	.625	.323	.418	.563	.182	.263	.846	.916	.879	.789	.808	.783
CNN (char)	.784	.946	.857	.604	.180	.264	.663	.124	.204	.848	.864	.856	.768	.787	.747
CNN (hybrid)	.820	.926	.869	.616	.322	.407	.628	.180	.265	.853	.910	.880	.790	.807	.781
RNN (word)	.856	.887	.870	.589	.514	.547	.577	.194	.287	.844	**.934**	**.887**	**.804**	**.815**	**.804**
RNN (char)	.606	**.999**	.754	.000	.000	.000	.000	.000	.000	.000	.000	.000	.367	.605	.457
RNN-attn (word)	.846	.898	**.872**	.593	.469	.520	.579	.194	.283	.849	.925	.886	.800	.814	.800
RNN-LTC (word)	**.857**	.884	**.871**	.583	.525	**.551**	.564	**.210**	**.302**	.846	.932	**.887**	**.804**	**.815**	**.805**
CNN (w/context)	.828	.910	.867	.609	.341	.429	.505	**.246**	**.309**	.840	.914	.875	.786	.804	.784
RNN (w/context)	**.858**	.880	.869	.577	**.527**	**.549**	.534	.175	.256	.840	**.937**	.885	.801	.813	.801

Table 2: Experimental results of learning models and their variants, followed by the context tweet models. The top 2 scores are marked as bold for each metric.

Since the dataset used is not split into train, development, and test sets, we perform 10-fold cross validation, obtaining the average of 5 tries; we divide the dataset randomly by a ratio of 85:5:10, respectively. In order to evaluate the overall performance, we calculate the weighted average of precision, recall, and F1 scores of all four labels, "*normal*", "*spam*", "*hateful*", and "*abusive*".

4.4 Empirical Results

As shown in Table 2, neural network models are more accurate than feature engineering based models (i.e. NB, SVM, etc.) except for the LR model—the best LR model has the same F1 score as the best CNN model.

Among traditional machine learning models, the most accurate in classifying abusive language is the LR model followed by ensemble models such as GBT and RF. Character-level representations improve F1 scores of SVM and RF classifiers, but they have no positive effect on other models.

For neural network models, RNN with LTC modules have the highest accuracy score, but there are no significant improvements from its baseline model and its attention-added model. Similarly, HybridCNN does not improve the baseline CNN model. For both CNN and RNN models, character-level features significantly decrease the accuracy of classification.

The use of context tweets generally have little effect on baseline models, however they noticeably improve the scores of several metrics. For instance, CNN with context tweets score the highest recall and F1 for "*hateful*" labels, and RNN models with context tweets have the highest recall for "*abusive*" tweets.

5 Discussion and Conclusion

While character-level features are known to improve the accuracy of neural network models (Badjatiya et al., 2017), they reduce classification accuracy for *Hate and Abusive Speech on Twitter*. We conclude this is because of the lack of labeled data as well as the significant imbalance among the different labels. Unlike neural network models, character-level features in traditional machine learning classifiers have positive results because we have trained the models only with the most significant character elements using TF-IDF values.

Variants of neural network models also suffer from data insufficiency. However, these models show positive performances on *"spam"* (14%) and *"hateful"* (4%) tweets—the lower distributed labels. The highest F1 score for *"spam"* is from the RNN-LTC model (0.551), and the highest for *"hateful"* is CNN with context tweets (0.309). Since each variant model excels in different metrics, we expect to see additional improvements with the use of ensemble models of these variants in future works.

In this paper, we report the baseline accuracy of different learning models as well as their variants on the recently introduced dataset, *Hate and Abusive Speech on Twitter*. Experimental results show that bidirectional GRU networks with LTC provide the most accurate results in detecting abusive language. Additionally, we present the possibility of using ensemble models of variant models and features for further improvements.

Acknowledgments

K. Jung is with the Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul, Korea. This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT) (No. 2016M3C4A7952632), the Technology Innovation Program (10073144) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

We would also like to thank Yongkeun Hwang and Ji Ho Park for helpful discussions and their valuable insights.

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.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Peter Burnap and Matthew Leighton Williams. 2014. Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making.

Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference*, pages 13–22. ACM.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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*, pages 29–30. ACM.

Maeve Duggan. 2017. Online harassment 2017. Pew Research Center; accessed 5-July-2018.

Antigoni 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. In *Proceedings of the International AAAI Conference on Web and Social Media.*

Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, et al. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 229–233. ACM.

Minlie Huang, Yujie Cao, and Chao Dong. 2016. Modeling rich contexts for sentiment classification with lstm. *arXiv preprint arXiv:1605.01478*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Janette Lehmann, Bruno Gonçalves, José J Ramasco, and Ciro Cattuto. 2012. Dynamical classes of collective attention in twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 251–260. ACM.

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.

Shafi Musaddique. 2017. Artist stencils hate speech tweets outside twitter hq to highlight failure to deal with offensive messages. Independent; accessed 5-July-2018.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Canadian Conference on Artificial Intelligence*, pages 16–27. Springer.

Adi Robertson. 2017. Facebook explains why it's bad at catching hate speech. The Verge; accessed 5-July-2018.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Toby Segaran and Jeff Hammerbacher. 2009. *Beautiful Data: The Stories Behind Elegant Data Solutions*. O'Reilly Media.

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *AAAI/IAAI*, pages 1058–1065.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 189–198.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26. Association for Computational Linguistics.

Guang Xiang, Bin Fan, Ling Wang, Jason Hong, and Carolyn Rose. 2012. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1980–1984. ACM.

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.

Seunghyun Yoon, Joongbo Shin, and Kyomin Jung. 2018. Learning to rank question-answer pairs using hierarchical recurrent encoder with latent topic clustering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1575–1584.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.