On the Impact of Word Representation in Hate Speech and Offensive Language Detection and Explanation

Ruijia (Roger) Hu*†
Clemson University
Clemson, SC
roger.rj.hu@gmail.com

Wyatt Dorris*†
Clemson University
Clemson, SC
wyattadorris@gmail.com

Nishant Vishwamitra*
Clemson University
Clemson, SC
nvishwa@clemson.edu

Feng Luo
Clemson University
Clemson, SC
luofeng@clemson.edu

Matthew Costello
Clemson University
Clemson, SC
mjcoste@clemson.edu

ABSTRACT

Online hate speech and offensive language have been widely recognized as critical social problems. To defend against this problem, several recent works have emerged that focus on the detection and explanation of hate speech and offensive language using machine learning approaches. Although these approaches are quite effective in the detection and explanation of hate speech and offensive language samples, they do not explore the impact of the representation of such samples. In this work, we introduce a novel, pronunciation-based representation of hate speech and offensive language samples to enable its detection with high accuracy. To demonstrate the effectiveness of our pronunciation-based representation, we extend an existing hate-speech and offensive language defense model based on deep Long Short-term Memory (LSTM) neural networks by using our pronunciation-based representation of hate speech and offensive language samples to train this model. Our work finds that the pronunciation-based presentation significantly reduces noise in the datasets and enhances the overall performance of the existing model.

CCS CONCEPTS

• **Security and privacy → Social network security and privacy**; **Social aspects of security and privacy**; • **Social and professional topics → Hate speech**.

ACM Reference Format:
Ruijia (Roger) Hu, Wyatt Dorris, Nishant Vishwamitra, Feng Luo, and Matthew Costello. 2020. On the Impact of Word Representation in Hate Speech and Offensive Language Detection and Explanation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy (CODASPY '20), March 16–18, 2020, New Orleans, LA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3374664.3379535

*Three authors contributed equally to the paper.
†Intern from D.W. Daniel High School, Central, SC, USA.

CODASPY '20, March 16–18, 2020, New Orleans, LA, USA

ACM ISBN 978-1-4503-7107-0/20/03…$15.00
https://doi.org/10.1145/3374664.3379535

1 INTRODUCTION

The proliferation of Online Social Networks (OSNs) has changed ways for information sharing. Users of OSNs can share information instantaneously with a large number of people all over the world. However, one of the biggest issues of this increased information sharing is their inherent potential to engender *hate speech* and *offensive language* that have been widely recognized as serious social problems. A Pew Research Center study [4] has recently reported that "roughly four-in-ten Americans have personally experienced online harassment, and 63% consider it a major problem". In our work, we consider hate speech to be abusive speech directed towards a particular group of people [7]. In addition, we consider offensive language as the presence of offensive or curse words in online comments [14]. Instances of hate speech may appear in all kinds of OSN platforms. These instances can have severe negative impact on users worldwide.

A solution to the problem of hate speech and offensive language is their detection using machine learning models [5, 6, 8, 11, 12] to root them out from the OSNs. Such machine learning models are typically trained on large datasets [2] of hate speech and offensive language, and these models learn the specific hate and offensive words, including their correlations, to make quite accurate predictions. For example, a logistic regression-based model is introduced in [2] which performs reasonably well on offensive language detection. However, it may not be able to accurately detect hate speech (40% of hate speech mis-classified). Therefore, the limitation in performance of this classifier may not allow a practical use in OSNs. A major limitation with all these models is that they do not attempt to represent the hate speech and offensive language samples in the training dataset in an efficient manner.

On analysis of the hate speech and offensive language datasets used by these existing models, we observed the language in their datasets is not regular language, but consists of an overwhelming number of samples with spelling and lexical errors, making the dataset very *noisy* in nature. We found that since most OSNs do not have any type of spell checks or language constraints, users often use this kind of language to communicate online. Since the existing machine learning-based detection models discussed above for hate

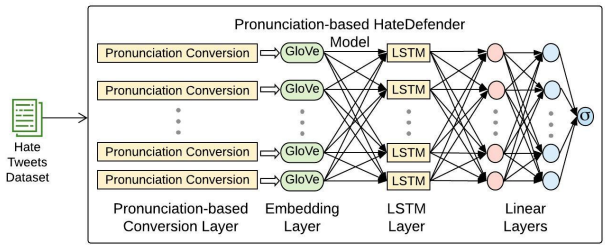

Figure 1: Overview of `Pronunciation-based HateDefender` System.

speech and offensive language detection, and the existing word embedding models (such as GloVe [10] and Word2Vec [9]) are trained on such noisy datasets, their efficiency in handling hate speech and offensive language detection tasks is severely impacted. In this work, we introduce a novel, pronunciation-based representation of hate speech and offensive language samples to train an existing deep learning-based detection model called `HateDefender` [3] and show that our representation significantly reduces the noise in the dataset, which improves the detection capability of the detection model by a large extent.

2 OUR APPROACH

2.1 Overview

The goal of our work is to study the impact of our pronunciation-based representation of hate speech and offensive language on existing detection models. The overview of our system is depicted in Figure 1. In the first phase of our system, we use our pronunciation-based conversion layer to convert the noisy hate speech and offensive language data to pronunciation-based representations. Next, these representations are used to train the detection model, that consists of a GloVe-based embedding layer, fully connected Long Short-term Memory (LSTM) layer and fully connected linear layers. Finally, we make a prediction from a Softmax layer for each sample to determine whether it is hate speech, offensive language or neither of the above.

2.2 Dataset

In our work, we used an existing annotated dataset [2] consisting of tweets labeled as hate, offensive and neither to train our pronunciation-based model. We first pre-process our dataset using the following methodology. To reduce noise in the dataset, mentions to other Twitter users (such as "@DonaldJTrump") were replaced with the identifier "<TwitterUser>". Next, links in the text were replaced with an identifier "<link>". By replacing the terms with identifiers, the deep learning model can still learn important features in relation to people, mentions, and links without the random noise. To further filter out the noise, we lower cased all samples in the dataset. Finally, we tokenized all the samples in the dataset.

2.3 System Design

2.3.1 Pronunciation-based Representation of Hate Speech and Offensive Language. In our analysis of the existing dataset of hate speech and offensive language, we found that the dataset has a lot of noisy samples. The Table 1 depicts some samples from the dataset.

Sample #	Noisy Sample	Pronunciation-based Sample
1	bi**h ni***a miss me with it	b'ItS n'Ig@ m'Is m,i: wID It
2	Hey pu**y you still there	h'eI p'Usi ju: st'Il D'e@
3	I'm sorry I fu**ed yo bi**h... But she really dig a ni**a	Ims'0ri; aI f'Vkt j'oU b'ItS b,Vt Si: r'i@lI d'Ig a# n'Ig@
4	Let's kill cra**er babies!	l'Ets k'Il kr'ak3 b'eIbIz
5	Ni**as be pressed for pu**y, Eeeeeen nothin	n'Ig@z bi: pr'Est fO@ p'Usi 'i:;i:;,i:n n'0TIn

Table 1: Dataset samples and their pronunciation-based representations.

From Table 1, we can observe that the input samples have a lot of noise. For example misspelled words (e.g. "yo"), repeated letters (e.g. "ni**a" and "ni***a") and grammatical errors (e.g. "you still there"). Such noisy dataset can have severe impact on the performance of existing detectors. In our work, we mitigate the negative impact of such noisy samples using a pronunciation-based representation method, where we express the words in the original samples in terms of phonemes [13]. Since the phonemes for noise such as misspelled words and repeated words are same (e.g. Table 1, samples 1 and 5, "ni**a", "ni***a" and "ni**ah" have same phoneme "n'Ig@"), this helps remove a lot of such noise from the training dataset. Thus, a model trained on less noisy pronunciation-based representations is more robust and with higher detection accuracy.

2.3.2 Hate and Offensive Language Detection. The goal of our `Pronunciation-based HateDefender` system is to accurately detect hate or offensive language in an input sample, using their pronunciation-based representations to improve detection accuracy. `Pronunciation-based HateDefender` system contains a detection model based on LSTM units. Our hate detection model is depicted in Figure 1. Our model consists of a deep LSTM network consisting of LSTM units and Linear layers. Specifically, we use bi-directional LSTMs in our detection model. We first preprocess the input samples (Section 2.2) and convert them to pronunciation-based representation (Figure 1, "Pronunciation Conversion"). Next, we transform the pronunciatio-based representations into 100-dimensional GloVe representations [10] (Figure 1, "GloVe"). This allows the LSTM network to associate similar words and make better predictions. Next, we train a bi-directional LSTM layer (Figure 1, "LSTM"), with the input sample embeddings generated in the GloVe layer. We use word representations to increase accuracy in the association and training process and decrease overall noisy samples.

From the LSTM layers, we get an output from each time step of the input sample. However, in the detection model, we only consider the output from the last time step . This output is then passed through linear (fully connected) layers so that the model can utilize the linear layer's trainable parameters to make more accurate predictions. Finally, we output a detection score according to Equation 1 (Figure 1, "Linear Layers").

Metric	HateDefender	Pronunciation-based HateDefender
Accuracy	90.82	91.69
Precision	60.56	61.87
Recall	64.71	65.10

Table 2: Evaluation of hate speech detection for HateDefender [3] and Pronunciation-based HateDefender.

Metric	HateDefender	Pronunciation-based HateDefender
Accuracy	89.10	90.77
Precision	83.82	84.54
Recall	84.23	84.66

Table 3: Evaluation of offensive language detection for HateDefender [3] and Pronunciation-based HateDefender.

The Equation 1 outputs a probability for each category in our dataset (hate, offensive and neither). We consider the final category to be the category that has the highest probability.

$$Detection\ Score = Softmax(h_t) \quad (1)$$

In our work, we use a one-versus-rest, "ensemble" framework where a separate detection model is trained for each category and the category label with the highest predicted probability across all detection models is assigned to each input sample. We discuss the performance of our ensemble model in further detail in the Section 3.

3 IMPLEMENTATION AND EVALUATION

3.1 Implementation

We use the open source software, eSpeak [1], to generate the pronunciation-based representations of the dataset samples. Our detection model is implemented as a one layer, bi-directional LSTM. We use 100-dimensional Glove [10] as word embedding model for our input samples for the purpose of transfer learning. We use one fully connected layer to improve prediction accuracy in our detection model. We train both of our models jointly using 5-fold cross validation, with 80% of the dataset for training and 20% for test. The dataset is converted to phonemes using espeak, which removes noise since words are broken down into their sounds. We use Adam optimizer with a learning rate of 0.0001 and we use Cross Entropy Loss as the loss function. For training the hate model, we use class weighting in the loss computation to mitigate the class imbalance in the dataset. We have used the PyTorch framework to train both our models. After training, we use the weights of the trained model for computing the partial derivatives involved in word salience computation.

3.2 Detection Model Evaluation

We use the test dataset to report our evaluation results of the detection model in our `pronunciation-based HateDefender`. We discuss the accuracy, precision, and recall of the pronunciation-based hate and offensive model, in comparison with the baseline model, without pronunciation-based representations. Table 2 and Table 3 depicts the performance of the pronunciation-based detection model on the accuracy, precision, and recall metrics.

Overall, from Tables 2 and 3, our pronunciation-based model performs reasonably better than the existing model [3] on both hate and offensive detection (average accuracy of 91.69% and 90.77% on hate and offensive language respectively). Higher percentages of accuracy in both hate and offensive language may indicate that the cost of an incorrect prediction may be quite low.

The higher value of hate and offensive language detection models could be attributed to the noise-removal effect of the pronunciation-based representations used in the pronunciation-based `HateDefender` model.

Moving further, we can also observe that the precision and recall metrics on both hate speech and offensive language detection is improved in the pronunciation-based model. This could indicate that in addition to removing significant noise, pronunciation-based representations can also make a model learn the difference between hate speech and offensive language better.

4 CONCLUSION

In this work, we have introduced a novel representation technique for hate speech and offensive language detection, based on pronunciation-based representations. This new representation can considerably improve the detection models that are trained on noisy datasets, by removing noise from the datasets. In our evaluation, it was observed that the model with the pronunciation-based presentations outperformed the model that is trained on the noisy datasets in terms of accuracy, precision, and recall metrics.

REFERENCES

[1] 2020. eSpeak text to speech. http://espeak.sourceforge.net/.
[2] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media.*
[3] Wyatt Dorris, Ruijia (Roger) Hu, Nishant Vishwamitra, Feng Luo, and Matthew Constello. 2020. Towards Automatic Detection and Explanation of Hate Speech and Offensive Language. In *Proceedings of the 2020 ACM International Workshop on International Workshop on Security and Privacy Analytics.* 1–1.
[4] Maeve Duggan. 2017. Online harassment 2017. (2017).
[5] Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding. 2018. Peer to peer hate: Hate speech instigators and their targets. In *Twelfth International AAAI Conference on Web and Social Media.*
[6] Claudia I Flores-Saviaga, Brian C Keegan, and Saiph Savage. 2018. Mobilizing the trump train: Understanding collective action in a political trolling community. In *Twelfth International AAAI Conference on Web and Social Media.*
[7] 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. In *Twelfth International AAAI Conference on Web and Social Media.*
[8] Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science.* ACM, 173–182.
[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[10] 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).* 1532–1543.
[11] Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth International AAAI Conference on Web and Social Media.*
[12] Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Tenth International AAAI Conference on Web and Social Media.*
[13] George L Trager and Bernard Bloch. 1941. The syllabic phonemes of English. *Language* (1941), 223–246.
[14] Wikipedia. 2019 (accessed November 18, 2019). *Profanity.* https://en.wikipedia.org/wiki/Profanity.