Detecting Cyberbullying in Online Forums

Hailey (Han Bit) Yoon
Department of Computer Science
Stanford University
hbyoon@stanford.edu

Abstract

Social networking platforms allow anyone to share one's opinions and views on various topics. Cyberbullies abuse this fact to attack another user/group online by posting offensive comments. This research presents a tool that can closely monitor and detect insults, sexual harassment, and other negative statements by utilizing Long-Short-Term-Memory (LSTM) and the optimization techniques such as data augmentation, hyperparameter tuning, and dropout.

1 Introduction

Social networking platforms have made it easy for us to communicate easily with family, friends, and others. It allows one to freely share one's opinions and beliefs on a public forum. These shared posts may be about interests, local affairs, events, politics or religion, people, and a wide variety of other topics. Unfortunately, some users encounter cyberbullying after sharing their opinion online. One could be bullied for one's religious or political beliefs, race or skin color, body image, if you have a mental or physical disability or for no apparent reason whatsoever. The effects of cyberbullying can result in mild distress to the most extreme cases like self-harm and suicide [3]. Therefore, offensive comments must be closely monitored and notify the administrator to take appropriate action.

This project aims to develop a deep learning architecture for text classification in terms of online hate speech, comment attacking someone based on his or her race, religion, ethnic origin, sexual orientation, and more. It mainly focuses on detecting offensive comments written in English. This project's approach employs a neural network solution composed of different models using Long-Short-Term-Memory (LSTM). The optimization techniques such as data augmentation, hyperparameter tuning, and dropout are used to achieve more accurate and less biased models.

2 Related Work

In "Detecting Online Hate Speech Using Context Aware Model" by Lei Gao and Ruihong Huang, they presented the importance of utilizing context information for online hate speech to improve hate speech detection accuracy. Gao and Huang had to gather data manually because all the publicly available hate speech annotated datasets do not contain context information. Once they trained their models with a dataset with more clear background information, it was easier for the algorithm to determine if the comment includes a vicious insult towards someone [4]. This method avoids mistakes, such as classifying a sarcastic joke as an insult based on the context. Also, "Detecting Offensive Language in Tweets Using Deep Learning" by Georgios Pitsilis, Heri Ramampiaro, and

Helge Langseth successfully distinguished racism and sexism messages within tweeter posts by utilizing an ensemble of Recurrent Neural Network (RNN) classifiers with various features associate with user-related information [2]. Despite these enhanced models in hate-speech classification, I believe the offensive comments detection can be improved by including new slangs and cultural expressions to determine what is offensive and not.

3 Dataset and Features

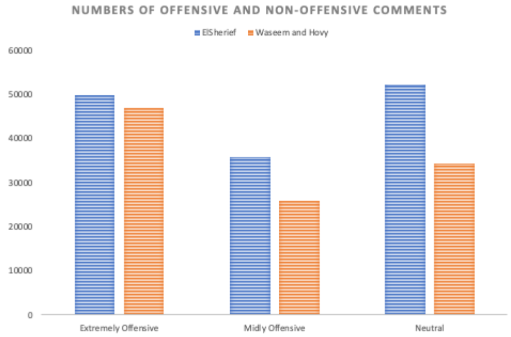

Figure 1

This research uses a collection of more than 28K hate speech tweets, which is made available by Mai ElSherief, Shirin Nilizadeh, Dana Nguyen, Giovanni Vigna, and Elizabeth Belding (2018); it also uses additional hate speech tweet database created by Waseem and Hovy (2016). These twitter datasets are available as CSV files; the files are named by the topic of offensive comment (Class, Sexual Orientation (sexorient), Disability, Gender, Ethnicity (ethn), Nationality (nation), Religion (rel), Archaic). As shown in Figure 1, both datasets are labeled in three features: Extremely Offensive, Mildly Offensive, Neutral. These are assigned a unique variable representing a user's tendency towards posting Extremely Offensive, Mildly Offensive, and Neutral content.

4 Methods

The first method used in this experiment was pure LSTM. However, the pure LSTM's runtime was slow, and its accuracy rate was low. Inspired by the work by Pitsilis, Ramampiaro, and Langseth (2018), the experiment combined various LSTM models, which improved the performance of offensive comment detections. Then, the project utilized CNN LSTM for faster execution; the final approach was to use a bidirectional LSTM with CNN for a shorter runtime and higher accuracy. Its pipeline is shown in Figure 2.

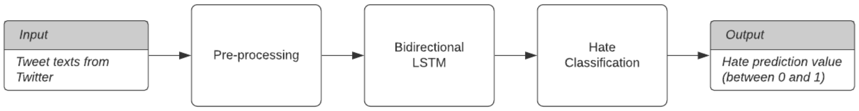

Figure 2

Figure 2 illustrates the pipeline of this project. The program first took the input of tweet texts from Twitter, and then it began pre-processing. Here, the pre-processing phase labeled each data based on the comment's offensive level. The pre-processed input entered bidirectional LSTM for training and hate classification, and then it generated output prediction into three features introduced before: (1) extremely offensive, (2) mildly offensive, or (3) neutral. The codes for this can be found at `https://github.com/hay318/Offensive_Comment_Detection`.

5 Experiments / Results / Discussion

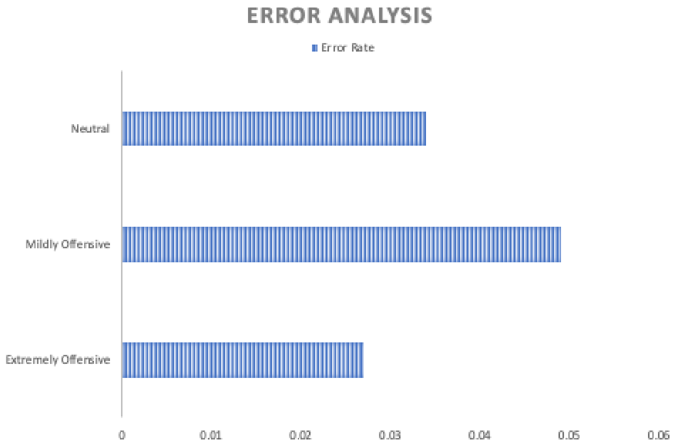

Figure 3

5.1 Predicting Offensive Speech in Two datasets

This study measured the prediction performances of offensive comment detection against two baselines, shown in Figure 1. I made several observations. First of all, the bidirectional LSTM with CNN outperforms the baselines on the Waseem and Hovy dataset. However, it underperforms them on the ElSherief dataset. I suspect this is caused by the baseline's external information (user information and metadata). By manually inspecting some failed predictions, I noticed that it is particularly challenging even for a human to determine if one's being sarcastic or genuinely making that comment without any background information. Figure 3 shows how detecting "Mildly Offensive" has the highest error rate; it is more difficult to determine what is offensive but not in an extreme way.

5.2 Determining hateful contents

I noticed that offensive slang words or modern jokes are correctly scored higher. However, I observed that word comments related to racism are misclassified because of the data sampling bias. Some racist tweet texts use Islamic or Muslim related terminology, which is falsely classified as racist comments. On the other hand, Asian related racist jokes got labeled neutral due to the language barrier. For instance, one would spell out offensive Chinese words using English alphabets and include in one's comment, and this would be classified as neutral because the offensive keywords spelled (spelled out Asian words) got labeled neutral because the offensive part is not written in English.

6 Conclusion / Future Work

With the increasing number of social media users, it is crucial to maintain an online environment where everyone feels comfortable and safe. The project successfully built an algorithm that detects cyberbullying (offensive comments) in online forums using a bidirectional LSTM with CNN. The current model supports English words only, but it would be interesting to build various language modules for this tool; I can imagine how it will be structured differently, given how each language has its own culture and standard of offensive comments. Applying the current algorithm to Spanish tweets will be my next goal, then eventually expand it to different languages such as Japanese and Arabic.

References

[1] C. Khatri, B. Hedayatnia, R. Goel, A. Venkatesh, R. Gabriel, and A. Mandal, “Detecting Offensive Content in Open-domain Conversations using Two Stage Semi-supervision,” thesis, 2018.

[2] G. K. Pitsilis, H. Ramampiaro, and H. Langseth, “Detecting Offensive Language in Tweets Using Deep Learning,” thesis, Trondheim, Norway, 2018.

[3] Get Safe Online. [Online]. Available: https://www.getsafeonline.org/social-networking/online-abuse/. [Accessed: 31-Oct-2020].

[4] L. Gao and R. Huang, “Detecting Online Hate Speech Using Context Aware Models,” thesis, 2018.