Safe Space

Andreas Zurhaar, Sandu Zuza [1]

May 26, 2019

Abstract

The purpose of this paper is to show how negative or hurtful comments can be censored and rewritten into positive ones using natural language processing techniques.Three different techniques of increasing complexity are explored and their resulting censorship capabilities are measured. This includes a "Simple Censorship," "Naive Vectorization," and "Smart Vectorization." Experiments were performed to fine tune these processes, and the conclusions based on this research are delved into. However, Smart Vectorization had the best overall results compared with the other two techniques. Possible improvements on our research are also included.

1 Introduction

Peoples interactions with negative comments have increased over the years, especially in a world where everyone is on social media. In 2014, a Pew Research Center study found that about 66 percent of internet users who have experienced online harassment said their most recent incident occurred on a social networking site or app [2]. A lot of these comments could be automatically filtered out and altered to be positive in order to ensure that people have a more positive experience online. That is the inspiration for this research, and this is why we have produced natural language processing techniques in order to mark negative comments and then transform them into positive comments.

In this paper these techniques are introduced and explained. Following this, results of the various experiments conducted on these techniques will be showcased and examined. These experiments revolve around how changes to the censorship techniques can improve or degrade the accuracy of the outputted censored comment. A section that includes ways to further the research is also included, finally ending with a conclusion of the results of everything discussed.

2 Corpora

In order to train any machine learning algorithms or perform any sentiment analysis, A few corpora (databases of words) need to be used for both labeling and for examples to train. For this project three different corpora were used. One is a database of all swear words (INSERT TITLE AND CITATION HERE) for help in labeling bad words. Another corpus is just examples of positive words(INSERT TITLE AND CITATION HERE) that is used for labeling positive words. The final corpus is from a Stanford project [1] and has 1.6 million sentiment labeled example tweets to extract and use.

3 Censorship Techniques

This section explains the mechanics and goals of each natural language technique used for censoring negative comments and replacing them with positive ones.

3.1 Simple Censorship

This module takes in a sentence and it replaces the swear words with "*". This is accomplished with the help of a database populated with negative and offensive words. Every word in the sentence is cross referenced with the offensive words and if there is a match that offensive word is replaced.

3.2 Naive Vectorization

Naive Vectorization takes in a sentence as an argument and locates all negative words. A bigram that is labeled "negative" is made from the found negative word and the word after it in the sentence. This bigram is then vectorized using Word2Vec, which is a python package that converts words into vectors with float numbers as values. This "negative" bigram's magnitude is compared to all other vectorized bigrams created from the source corpus of 1.6 million comments, and is replaced with the bigram that has the closest values that also has a positive label. Thus the negative words are removed with a positive bigram as a replacement.

3.3 Smart Vectorization

Smart Vectorization module is similar to Naive Vectorization, as it also takes in a sentence compares two vec-

[1] Universiteit Maastricht, Department of Computer Science, P.O. Box 616, 6200 MD Maastricht, The Netherlands

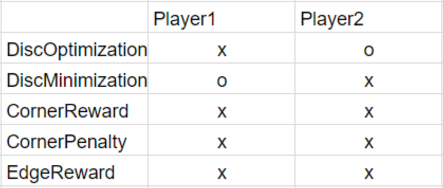
torized bigrams. The key difference is that the ”negative” bigram from the inputted sentence starts with the word before the negative word. This ”negative” bigram is then cross referenced with ”positive” bigrams created from the source corpus that also start with the same word. Then the magnitude of the second words of both bigrams are compared. The ”positive” bigram whose second word has the closest magnitude to the second word of the ”negative” bigram replaces the ”negative” bigram. This helps the censored sentence keep some semblance of meaning from the orignial sentence.

4 Experiments

To see how the different implemented censorship techniques can be improved different experiments were ran. All of them were designed to help the overall improvement of censorship as well as maintaining a similar context of the uncensored comments, as well as gain insight to why they work the way they do.

4.1 Word Stemming Experiments

The first step in creating a competitive algorithm that is able to win a game is to obtain a strategically .



	Player1	Player2
DiscOptimization	x	o
DiscMinimization	o	x
CornerReward	x	x
CornerPenalty	x	x
EdgeReward	x	x

Figure 2:

Overall Winner: player 2
Win Percentage: 59%

5 Conclusions and Insights

The results of our experiments demonstrate that as expected, Alpha Beta outperforms the other two implemented algorithms in this journal. Alpha Betas ability to predict future moves gives it an advantage over Greedy and also Monte Carlo. Othello being a deterministic game makes Monte Carlo perform worse as a result of its random factor.

6 Further Research

Talk about how our research could be extended or furthered.

References

[1] Bhayani, R. and Huang, L (2009). Twitter sentiment classification using distant supervision. *Stanford*, Vol. 1, p. 12.

[2] Duggan, Maeve (2014). Online harassment. *Pew Research Center*, Vol. 1, p. 1.