

CS221: Project Progress Report

Nimisha Tandon

nimisha@stanford.edu

Shaila Balaraddi

shailaab@stanford.edu

Naman Muley

ngmuley@stanford.edu

November 15, 2019

1 Introduction

We are tackling the problem of identifying articles that could be potentially dangerous to a brand. Our approach has two steps. First, we take in a set of articles and classify them into categories. In the final application, this could be done by pulling articles from the internet daily. Once these categories are identified, in the second step, we create an impact score for the brand in question. The articles with top impact scores can be provided to the client as having high potential for impact to the brand.

This problem has a rich application of various algorithms for natural language processing. Since proposing this project, we have explored a few algorithms to create a better sense of what an impact score can be. We looked into utilizing unsupervised learning algorithms like GloVe and supervised algorithm implementations like Naive Bayes. These have given us better impact scores. Further more, it feels like we could perform some more fine tuning to come up with better models to bring a richer understanding of the *impact score*

2 Methodology

2.1 Classification

In making progress for the project, we decided to give more weight to the impact score analysis section, since that's the more novel component of the solution approach. Our classification currently still uses a simple Stochastic Gradient Descent classifier. We found it's accuracy to be nearly 82 percent and thought we can improve upon that in the later parts of the project.

We still did look at using the Naive Bayes algorithm to perform classification, since that's what literature reported is it's primary use. It performed slightly better at classification than SGD but not by much.

2.2 Impact Score

Impact score needs to be a number that represents how relevant or impactful will a particular article be to the brand. Hence, we extended our previous approach of having a vocabulary of words that are relevant to the brand and applying a combination GloVe analysis and TF-IDF to come up with a score of how closely does the article's content relate to the dictionary relevant to the brand.

2.2.1 Vocabulary of a brand as an input

It was possible to decipher a set of words that are deemed contextually more dominant and important for a category like *guns* by various methodologies including TF-IDF itself. But we believe finding relevancy to a brand should allow some input bias from the brand. For example, a brand may want to find articles relevant to a subset of their very specific products and those may not be popularly present in the testing dataset.

We allow a list of words to act as our vocabulary for finding articles with highest potential impact. For example, for a brand like NRA, we take the following set of words as a vocabulary:

["guns", "shooting", "weapon", "nra", "handgun", "assault", "rifle", "america", "firearm"]

2.2.2 Similar Words using GloVe learning

A measure of how relevant an article is to a brand is to know how many of semantically or linguistically similar words occur in that article. For each of the words in the vocabulary, we get a list of *similar words* using the GloVe learning algorithm [1]. We then use a simple TF algorithm to understand how commonly do these similar words occur in an article.

Algorithm 1: Impact Score using similar words from GloVe

```
initialization;  
while While condition do  
    instructions;  
    if condition then  
        instructions1;  
        instructions2;  
    else  
        instructions3;  
    end  
end
```

2.2.3 Relevant words using TF-IDF

2.2.4 Get Stack Rank of Articles

3 Implementation

Talk about how the models are trained, what data sets did we use for these models. how are the data sets relevant. Future work may involve using and creating better data sets

4 Preliminary Results

Show similar words, maybe a fancy representation of similar words and important words as found by GloVe and TF-IDF respectively

5 Future Work

1. Get better training data for the algorithms 2. Have the application fetch testing data from today's newsfeed