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

We extend our two step approach which we presented in the proposal. We have spent time exploring different algorithms for both these stages. We seem to have settled on to a methodology, which we explain in this section.

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, TF-IDF and Naive Bayes 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 Input Brand Vocabulary

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 way 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: Extend Vocabulary with Similar Words using Glove Model

```
$Vocabulary \leftarrow [”guns”, ”rifle”, ”weapon”, ”nra”, ”handgun”, ”firearm”]$;
$model \leftarrow train\_glove($*glove.6B.300d.w2vformat.tx*$)$;
**GetGloveWords** $(Vocabulary, model)$
    $extended\_vocab \leftarrow Vocabulary$;
    **foreach** *word* $w \in Vocabulary$ **do**
        $similar\_words \leftarrow model.get\_similar\_words($word=w, $topN = 10)$;
        $extended\_vocab.extend($similar_words$)$;
    **end**
    **return** $extended\_vocab$;
```

Once the vocabulary is extended, we use this extended vocabulary to identify articles that have the most occurrences of these words. In the Proposal, we had used a similar algorithm to calculate our baseline but with the vocabulary as the raw input set of words. Following is a figure that shows how the expanded vocabulary behaves for a brand like NRA and input vocabulary = *["guns", "weapons", "nra"]*.

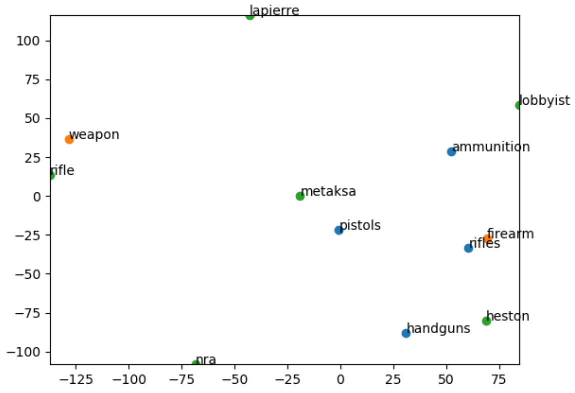

Figure 1: Similar Words obtained from GloVe Learning for *NRA*

We use this extended vocabulary to come up with an impact score by simply counting frequencies of these words. Our algorithm that performs this to come up with an impact score for each article in the category is given below. The score dictionary contains the impact score calculated for each article in the set of articles.

Algorithm 2: Calculate Impact Score for each article from a list of articles, given extended vocabulary

```
CalculateVocabImpactScore $(V, D)$
    inputs : A list of words that form ExtendedVocabulary V; a list of articles D
    output: A dictionary representing impact scores for all articles in D
    $score \leftarrow \emptyset$;
    $RangeD \leftarrow range(0, len(D))$;
    foreach index $i \in RangeD$ do
        score[i] = 0 ;
        freq = [] ;
        /* iterate over all words in $i_{th}$ article to count frequency */
        foreach word $w \in D[i]$ do
            freq[word] += 1 ;
        end
        /* iterate over all words in V to build a score for article $i$ */
        foreach word $w \in V$ do
            score[i] += freq[w] ;
        end
    end
    score.Normalize() ;
    return $score$;
```

We have detailed our results of the above score dictionary in the Results section.

2.2.3 Relevant words using TF-IDF

Term Frequency-Inverse Document Frequency commonly known as TF-IDF can be used to find the frequency of vocabulary words in a document to perform sentiment analysis or extract relevancy from a document.

This algorihm has 2 algorithms working together.

1. Term Frequency

$$tf(t,d) == \frac{noofoccurrencesoftermindoc}{totalnumberofallwordsindocument}$$

Before using this algorithm, we first do some preprocessing of the doc as follows:

- Remove all stop words.(ex: in, the,are it)
- Convert all words to lowercase.

Now using the term frequency alone will end up giving us words that are not unique (for ex: repeated words, should not add up to the relevance of the document)

2. Inverse document frequency

This gives us the uniqueness of a word:

$$idf(t,d) == log(\frac{nooftimesthetermappears}{noofdocumentscontainingtheword})$$

The final step is to multiply the two together to get the TFIDF score which would give us the impact score.

3 Implementation

Following are some important points to mention about implementation details:

- *Application* Currently we have a python application that performs the training of the supervised algorithms and data transformations. We are working on a single *guns* category. We also use a static set of vocabulary mentioned above, assuming we got that as input from a fictitious National Rifles Association (NRA) client
- *GloVe training* We trained GloVe on the *Wikipedia 2014 + Gigaword 5* dataset mentioned on the project's website (*https://nlp.stanford.edu/projects/glove/*)
- *Compute* We noticed that running GloVe on a laptop is not fast. Going forward we will use the compute credits provided to us.

4 Preliminary Results

We will outline the impact scores we identified using both the methods below. We are yet to come up with a justifiable way to combine these scores to present a single final score:

4.1 Similar Words using GloVe

Article Index	Impact Score
5893	0.021
692	0.017
6763	0.016
7274	0.014
326	0.014
3613	0.014
3831	0.013
1665	0.011
3781	0.011
2135	0.010

If we compare it with the impact scores presented in the Proposal, we notice that articles have shuffled up a bit. Here are some points to consider:

1. Article index 7274 was at the top, now it is in 4th place, whereas 5893 is at the top now.
2. Article 5893 is a long article about 2nd ammendment, which is extremely relevant to NRA while 7274 is an article about gun buy backs, which is also fairly relevant.
3. Article 2135 has remained at the 10th place.

5 Future Work

This impact score, we feel, still does not capture all the richness we can provide using natural language processing algorithms. We intend to improve in the following areas:

1. *Combining Scores from Different algorithms*: TF-IDF and GloVe, both gave a varying set of articles that each thought were more impactful. We would like to investigate good strategies to combine these and present a final score.
2. *Aspect based Sentiment Analysis*: ABSA is another methodology that literature showed is a good way of identifying important articles. Add aspect based sentiment analysis where we will come up with some aspects which add more meaning to the impact score.
3. *Expand the number of categories*: Generalizing our work to focus on more categories
4. *Better Trained Models*: Currently the GloVe model we use is trained from the wikipedia dataset. We intend to see if training it on a more relevant dataset creates a better set of similar words that we can use to calculate impact scores

6 Appendix