Stanford University

CS221 Final Project - Impact Lens

Nimisha Tandon
Naman Muley
Shaila Balaraddi
December 13, 2019

1 Introduction

A lot of information on the internet like news, social media posts etc that can impact brands. Directly, in the form of positive or negative PR and indirectly in that their product strategy could be potentially informed from these. Identifying such news is extremely useful for large brands or analytics divisions to get ahead of the PR cycle and formulate an early response.

We attempt to identify information that can be potentially impactful to a brand. We decided to tackle this problem by using classification algorithms and impact analysis methods. This project focuses on using classification to break down the documents into various categories and then look through the lens of a specific brand to calculate an impact score of each document. We present a stack rank of such documents back to the brand.

2 Method Overview

Approach This project can be broken down into two stages: a) Classification and b) Impact analysis. Both these operations will deploy ML techniques. The system will take as input some domain data which represents the brand. It will also present to the user certain categories or news groups that the brand is interested in.

3 Classification

3.0.1 Stochastic Gradient Descent with TFIDF Vectorizer

For classification of the documents into categories, we used the SGD classifier with TfIdf vectorizer. SGD is a simple and efficient optimization algorithm for learning of linear classifiers. SGD randomly chooses training data, gradually decreases the learning rate, and penalizes data points which deviate significantly from what's predicted. Since we had a large dataset to go through, SGD was the optimal and simplest algorithm to use.

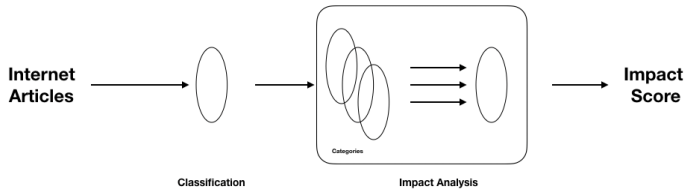

Figure 2.1: Process Overview

3.0.2 Feature Extracton process

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document.

The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some words appear more frequently in general.

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

3.0.3 Model fitting

We then fit the model with preprocessed training data and labels. We feed the model with training data and their categories to train it to accurately classify the documents into respective categories.

Classify We then run classification of test data on this trained model to get dataset categorized into various topics.

4 Impact score analysis

4.0.1 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.

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"]*.

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

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 $extended\_vocab$;

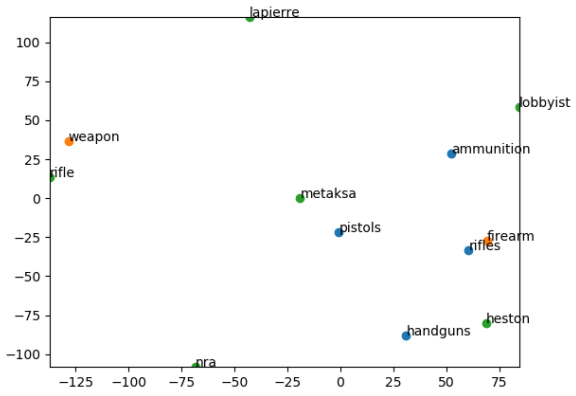

Figure 4.1: Similar Words obtained from GloVe Learning for *NRA*

in the category is given below. The score dictionary contains the impact score calculated for each article in the set of articles.

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

4.0.2 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 pre processing 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)

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 $score$;

2. Inverse document frequency

   This gives us the uniqueness of a word:

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

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

5 Data

We used the following datasets for various learning algorithms.

1. ***20-News:*** The newsgroup dataset contains 18000 news posts on 20 topics split. We used this data set to train the classifier and to come up with fundamental categories for the news articles.

   The categories provided by this dataset seemed to work just fine for our purposes. Following are the categories that this data set provides. To showcase our system and it's functionalities, we created a fictitious client NRA and chose the *talk.politics.guns* category to work upon in detail.

   a) comp.graphics
   b) comp.os.ms-windows.misc
   c) comp.sys.ibm.pc.hardware
   d) comp.sys.mac.hardware
   e) comp.windows.x

f) misc.forsale
g) rec.autos
h) rec.motorcycles
i) rec.sport.baseball
j) rec.sport.hockey
k) talk.politics.misc
l) talk.politics.guns
m) talk.politics.mideast
n) sci.crypt
o) sci.electronics
p) sci.med
q) sci.space
r) talk.religion
s) alt.atheism
t) soc.religion.christian

We used the 20 News group to also assess accuracy of our classifiers. The test data split out of this dataset was used to assess the accuracy of the classifiers.

2. ***New York Times articles:*** A set of 6179 articles from New York Times archive of December 2018. These provide a perfect testing dataset to see how our system classifies articles into categories and provides impact scores for each.

6 Results

We wanted to showcase a few sets of results from our experiments.

6.1 Classification

- Naive Bayes vs. Stochastic Gradient Descent The SGD classifier did slightly better than the Naive Bayes classifier. <Input table>
- Loss function comparison <Talk about comparison of various loss functions>

6.2 Enhancing Vocabulary with GloVe embeddings

6.3 New York Times results

The NYT archive of 2018 December provided a great real world ground for our complete end-to-end testing.

- Classification The SGD classified 189 articles into the *talk.politics.guns* category. Articles were on variety of topcis like gun shootings, poaching in africa, drug cartels around the world, police aggression in US etc. All of these were rightly classified.

- Impact Scores

  We found the impact scores change after enhancing the vocabulary with GloVe embeddings. With the raw input vocabulary, the articles that came in the top 10 index were these:

  After the use of GloVe embeddings, with the vocabulary that included the positive GloVe embeddings, we found the articles change impact scores. New articles came into the top 10, while those that were present changed their positions.

7 Conclusion

8 References