A Report on Unsupervised Opinion Mining using NLP Techniques

Name:	**Rishabh Bhonsle**
Registration No./Roll No.:	17217
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	October 29, 2021
Date of Submission:	November 20, 2021

1 Introduction

Opinion mining (sometimes known as sentiment analysis or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine. [Source: Wikipedia]

Different approaches and techniques are used to classify the sentiment of texts. Word embedding is one of the effective methods that represent aspects of word meaning and help to improve sentiment classification accuracy. The aim of this project is to develop a method to find major/significant opinions from given data to identify qualities of a Prime Minister using Wordnet and Word2Vec (BOW or Skipgram model or both) to accomplish this task.

2 Methods

The methods used in this analysis can be classified into two categories based on the basis of the approach used:

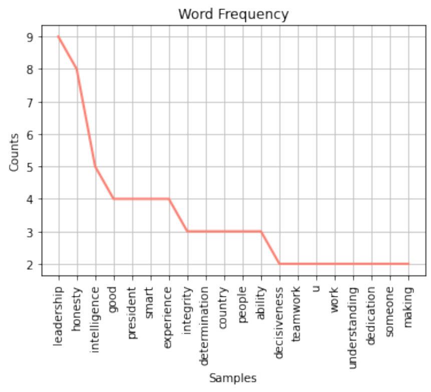
2.1 Wordnet Based Approach

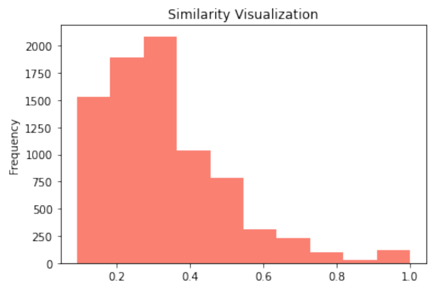
The experiment numbers refer to those in their implementations found at - Wordnet Implementation

2.1.1 Pre-Processing

We begin by tokenizing our raw text using the Regular Expressions Tokenizer Module of NLTK. We don't apply lemmatization here as it doesn't have a significant effect on the tokens generated. As qualities can be described using nouns and adjectives, the words are further filtered to include only nouns, adjectives and satellite adjectives using Wordnet. Here we use the synsets interface belonging to NLTK for word property look-ups. These words are then filtered to exclude stop words. These words are further filtered by replacing words signifying negative sentiments by their antonyms. Negative words might not be associated with qualities of a PM and might be used in a negative context like "not bad" where "bad" has a negative sentiment. We thus create our vocabulary with each word having a frequency tag and a part of speech tag.

Arranging the words of our vocabulary by decreasing order of their frequencies already starts to surface a few qualities (fig on the left). We also visualize a comparison of frequencies of word similarities between two words of a document measured by a similarity score (here, WuPalmer Similarity) to get an idea of the "similaritiness" of our vocabulary (fig on the right)





2.1.2 Experiment 1 - WuPamer Similarity

1. Consider a word of the vocabulary and sum its similarity scores with other words of the document.
2. The average of this sum is the document similarity score we assign to the word.
3. Repeat this process for every word of the document and arrange the words in decreasing order of similarity.
4. The top words should thus be the most contextual words associated with our vocabulary and thus the given data.

2.1.3 Experiment 2, 3 and 4 - Synonyms, Derived Forms and tf-syn-der

1. In each of these experiments, consider a word of the vocabulary and associate with it, its synonyms or derived forms using lemmas on the synsets of the words using Wordnet.
2. In case of synonyms, we associate with each word, an average of the sum of similarities (here WuPalmer Similarity) of the synonyms of that word with the synonyms of the rest of the words along with the words themselves (syn-sim score) from our vocabulary.
3. This serves as an extended vocabulary clustering of similar words. The experiment thus skip the ”clustering and then ranking the clusters” process to identify words significant to highest ranked clusters and rather directly compute them as the most contextual words with respect to this extended vocabulary.
4. The above two steps are repeated for derivational forms of words as well to obtain the derivational forms similarity scores (der-sim score).
5. We then incorporate term frequency of a word in our vocabulary, its extended synonym vocabulary and its extended derivational forms vocabulary to compute a score based on the product of tf values, syn-sym score and der-sim (tf-syn-der) to get an overall contextual significance for the words of our vocabulary associated with the given data.

2.1.4 Experiment 5 and 6 - KMeans and Extended Kmeans

1. In this experiment, the vocabulary is converted into a tf-idf vector using NLTK’s TfidfVectorizer() function.
2. This vector helps cluster the vocabulary words together on the basis of distances between word vectors using KMeans clustering.
3. The number of clusters is determined by the elbow method for finding the ’k’ value of the KMeans model.
4. The model is evaluated using a kmeans score from the scikit-learn library to estimate the quality of the clustering. The most contextual clusters of all clusters is then used for our model evaluation.
5. This experiment is then repeated on an extended vocabulary including synonyms, derivational forms and even word phrases. These word phrases are found by considering bi-grams and checking their phrase validity using Wordnet’s vocabulary.

2.1.5 Experiment 7 - WSD

1. Consider the meaning of the question - ”What qualities do you think are necessary to be the prime minister of India?”. This sentence is used to better obtain contextual words from the vocabulary pertaining to our document.
2. The experiment identifies nouns and adjectives from the question and their Wordnet definitions to obtain a reference context.
3. Then similarity scores (here WuPalmer Similarity) are measured between the words in this reference context and the vocabulry words.
4. The words with the highest scores should thus be the most contextual words in the given data that pertain to the meaning of the question in hand.

2.2 Word2Vec Based Approach

The Word2Vec approach is based on the BOW Model here. We leave the Skipgram model implementations for the same experiments as future work. The experiment numbers refer to those in their implementations found at - Word2Vec Implementation

2.2.1 Pre-Processing

We begin by tokenizing our raw text into sentences. We identify phrase words in our sentences and replace individual words with their respective phrase words. We thus create our vocabulary of sentences for the gensim Word2Vec model (experiments - A). We also import a pre-trained word embedding model - glove-wiki-gigaword-100, a 100 dimensional document vector trained on Wikipedia for better word embedding for our experiment (experiments -B). Even better models are available, however we consider this one due to lack of computational power on the available devices used. [Ref: GloVe]

2.2.2 Experiment A1, B1, B2, B3 and B4 - Similarities

Here we combine a list of experiments as they are somewhat similar to those performed using Wordnet but are re-done to portray the significance in change of model to Word2Vec.

1. The Word2Vec model is trained on the given data set.
2. The experiment identifies the words most similar to the word "qualities" just as an analysis to see the effectiveness of our model. (A1)
3. This experiment is repeated on the Word2Vec model trained on the Wikipedia corpus by choosing most similar words to the word in context context and also present in our given data. (B1)
4. This experiment is further expanded to consider all nouns and adjectives of the question words and phrase words and then considering most contextual words to the words in questions based on word embedding trained on Wikipedia corpus. (B2)
5. Another variant of this experiment is done by expanding the question context vocabulary by finding top 10 similar words of each word by computing similarities using the trained Word2Vec model. (B3)
6. Finally we run an experiment to find out the most contextual words from the given data set using the similarity model trained on the Wikipedia corpus. Each word in our vocabulary is measured with every other word in our data set and words predicted (top 10) to be similar to those. (B4)

2.2.3 Experiment A2 and B6 - KMeans

1. In this experiment we cluster the words of our sentences using KMeans.
2. Average silhouette values for each batch are considered for scoring each cluster and thus the cluster with the highest score is considered.
3. We thus obtain the most representative terms for that cluster based on centroids. (A2)
4. This experiment is repeated on the Word2Vec model trained on the Wikipedia corpus. Here, an approximation of KMeans called Mini-batch KMeans is used to make our algorithm computationally viable. We base this experiment on a pre-existing implementation of its functions. [Ref: mbkmeans] (B6)

2.2.4 Experiment B5 - Meta-Similarities

1. This experiment works on the principle that words associated with a word with CBOW/Skipgram Model would appear around the word.
2. Thus the most significant words are now found by comparing the similarity of words with these contextual words that are most likely to appear around the words that they are around of.
3. The words in the vocabulary are thus scored on the basis of their similarities to these contextual words and arranged in decreasing order to find out the most significant words.

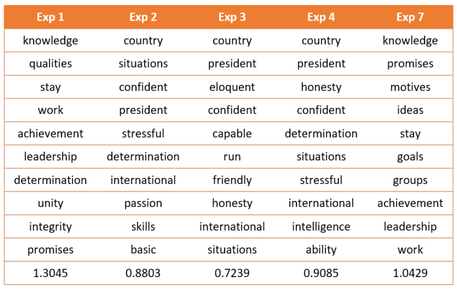
2.3 Score Evaluation

We evaluate the WuPalmer Similarity scores of the words of the result with themselves. We set this as a base score. We then evaluate generated or predicted words by our experiment in a similar way and compare these to the result words score. This is in no way an absolute measure, but gives a decent solution for comparison between models.

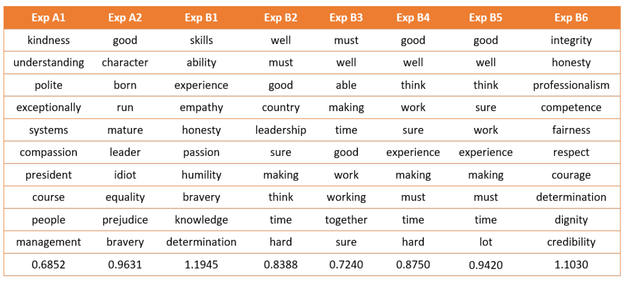
3 Experimental Analysis

In this section we present an analysis of our experiments. Each column represents the words generated in the experiment corresponding to the title of the column. The score for that experiment is written below each column. Note that we do not show experiments 5 and 6 of Wordnet here based on KMeans as their scores were quite low to be significant enough to consider their word clusters for scoring.

Thus we see that experiment 1 from WordNet and experiments B1 and B6 yield pretty good results. Result from B6 can be considered an alternative solution to the problem as the words predicted by the experiment can all be deemed as essential qualities of a Prime Minister.



Exp 1	Exp 2	Exp 3	Exp 4	Exp 7
knowledge	country	country	country	knowledge
qualities	situations	president	president	promises
stay	confident	eloquent	honesty	motives
work	president	confident	confident	ideas
achievement	stressful	capable	determination	stay
leadership	determination	run	situations	goals
determination	international	friendly	stressful	groups
unity	passion	honesty	international	achievement
integrity	skills	international	intelligence	leadership
promises	basic	situations	ability	work
1.3045	0.8803	0.7239	0.9085	1.0429



Exp A1	Exp A2	Exp B1	Exp B2	Exp B3	Exp B4	Exp B5	Exp B6
kindness	good	skills	well	must	good	good	integrity
understanding	character	ability	must	well	well	well	honesty
polite	born	experience	good	able	think	think	professionalism
exceptionally	run	empathy	country	making	work	sure	competence
systems	mature	honesty	leadership	time	sure	work	fairness
compassion	leader	passion	sure	good	experience	experience	respect
president	idiot	humility	making	work	making	making	courage
course	equality	bravery	think	working	must	must	determination
people	prejudice	knowledge	time	together	time	time	dignity
management	bravery	determination	hard	sure	hard	lot	credibility
0.6852	0.9631	1.1945	0.8388	0.7240	0.8750	0.9420	1.1030

Experiment 1 from Worndnet defines the most contextual words from the given corpus. Adding synonyms and derivative forms to the vocabulary and expanding it seems to impact it negatively. Also being trained on a small corpus, clustering doesn't seem to be very effective as seen by the failure of the elbow method to even pin-point a distinct "k" value. The WSD approach is novel, yet experimental. Maybe an addition of dictionary words of the data vocabulary apart from the question vocabulary to compare similarity scores might have helped.

Experiment B1 from the Word2Vec approach is similar to Experiment 1 from the Wordnet approach. Thus the explanation of its success remains the same. Experiments B2, B3 and B4 here are analogous to experiments 2 and 3 of wordnet. The expansion of vocabulary and training word embedding of a bigger corpus for similarity comparison doesn't seem to make a difference. However, these experiments on models of Word2Vec, even though failed, are an improvement over the models of Wordnet thus signifying a potential method for word embedding trained on a larger corpus. Experiment B5 was supposed to be an improvement over B4 using meta-similarities, however there is no significant difference in their result. Thus the assumption mentioned for performing the experiment does not hold. Finally experiment B6 based on Kmeans seems to be a significant success. As the model is now trained on a significantly larger (yet smaller than usual standards), the clusters built seem to be much more relevant for capturing the context of the data set given.

4 Discussions

4.1 Merits

1. The analysis project helped with looking for ways of manipulating text data for analysis and seeing how they relate in various ways.

2. Though a few new methods were explored apart from those done in the classroom lectures, it was a good practice to witness the manipulation of data in numerous forms first-hand merely from the few methods learnt.

4.2 Limitations

1. Due to limited access to computation power for immediate exploratory analysis by training over huge data sets like the Wikipedia corpus, certain compromises in the dimensionality of vectors generated for the data was limited. This caused the experiments to constraint their features from its full potential.

2. Secondly, the data set given consisting of only 38 sentences in itself is limited for highlighting words that contextualise it. A bigger data set would have improved the models' accuracies significantly.

4.3 Future Scope

1. Given enough computational power, word embeddings on larger corpuses with higher dimensionality could be considered. Not only that, but the models could then even be self trained to get better word embeddings to suit the purpose of the experiments.

2. Another suggestion for modification would be to use only those nouns which have adjective forms and are not just nouns themselves. This would eliminate nouns like "president" or "country" which keeps popping up in the results but is not something we are looking for.

3. The experiments based on the Word2Vec model only consider BOW models. The experiments could be repeated and explored on Skipgram models as well.

4. Lastly seeing the results on the experiments, it would be interesting to explore even more clustering algorithms apart from those done manually by similarity measures (which also could be explored futher apart from WuPalmer Similarity) and KMeans. Given enough computational power, even mini-batch KMeans could be generalised to KMeans for bigger data sets.