Opinion mining using unsupervised learning techniques.

Name:	**Dinesh Adhithya**
Registration No./Roll No.:	18097
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	October 29, 2021
Date of Submission:	November 20, 2021

1 Introduction

The task of this work is to use unsupervised learning techniques to perform opinion mining.The data set contains 38 opinions for the question ”What qualities do you think are necessary to be the prime minister of India?” . The gold standard qualities is also provided which has been curated by experts and we use WordNet and Word embedding models such as CBOW and Skip-gram models trained on twitter and Wikipedia data sets, along with that use clustering methods along with this to find opinion from the data set given.

2 Methods

2.1 WordNet based methods

WordNet [1] is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.The words were extracted from the data set provided , cleaned and stop words discarded.These words form the vocabulary.The Noun and Adjective words were alone filtered using WordNet as a single word could be used in various parts of speech and adjective and nouns would describe the qualities of a prime minister rather than a verb or pronoun.[2]

2.1.1 Word Frequency

All words related to words from our vocabulary were found using WordNet and added to our vocabulary and simple frequency analysis of the words in the new vocabulary were found.The following words occurred more than 5 times in the new vocabulary.

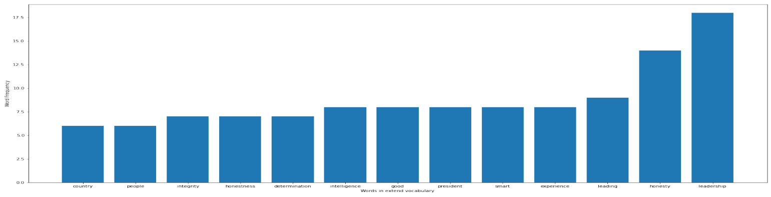

Figure 1: Most frequent words in the extended vocabulary

2.2 Cluster Frequency

2.2.1 Algorithm

1. The words in the vocabulary are filtered , only nouns and adjectives are retained and rest of the words are removed as our objective is to find qualities of a prime minister.We sort the values in the vocabulary in order of decreasing frequency of terms and remove duplicates.

2. Then using sysnets function and lemma function in WordNet we find all derivationally related words of the words in the vocabulary and add it to the vocabulary.

3. We find similar words from the vocabulary and put them in a cluster using various measures of similarity such as wu palmer's similarity based on a threshold (around 0.9).

4. Empty clusters are ignored , the clusters with same words are then combined.

5. The clusters are then stored in decreasing order of frequency with corresponding labels.

2.2.2 Cluster labels frequency

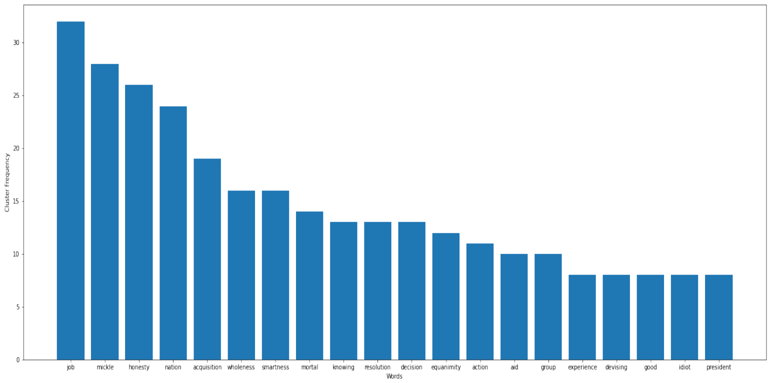

Figure 2: Most frequent cluster labels in the extended vocabulary

2.2.3 Model Prediction

Words classifier as important by the model are : job , mickle , honesty , nation , acquisition , wholeness , smartness , mortal , knowing , knowing , resolution , decision , equanimity , action , aid , group , experience , devising , good , idiot and president.

2.3 Network Models

2.3.1 PageRank

PageRank is an algorithm used by Google Search to rank web pages in their search engine results. A network was build with each word in vocabulary as a node , latter edges were added with each edge weight being assigned to the wu Palmer similarity between those two nodes.[3]
The following were the 15 most important words: skill intelligence mess mickle understanding negotiator plenty Brobdingnagian hatful state crucial sight Max_Born kindness stack.

2.3.2 Degree centrality

For all those words whose wu Palmer similarity is more than a threshold (set at 0.9) , a edge is added between those words and by using degree centrality , degree of a node in a network is the no. of edges connected to it. Essentially we look for words which have large number of connections with other nodes , higher would be its importance.
The 15 most important words happened to be : skill intelligence mess mickle understanding negotiator plenty Brobdingnagian hatful state crucial sight Max_Born kindness stack.

2.4 Clustering Techniques

We construct a distance matrix , based on wup Palmer similarity metric where each element in the matrix of row i and column j happens to be:

distance_matrix[i][j]=1-wu_palmer_similarity[vocabulary[i]][vocabulary[j]]
Using these pre-computed distance matrix we use different clustering techniques and try to find cluster centers and find important words in the data set.

2.4.1 Agglomerative Clustering

We cluster similar words together to form larger clusters build on smaller clusters based on pre-computed distance matrix.
Finding the most important words in the found clusters were : body_politic management political morality retard goal committedness international trying trusty thought controlling politically sooner background

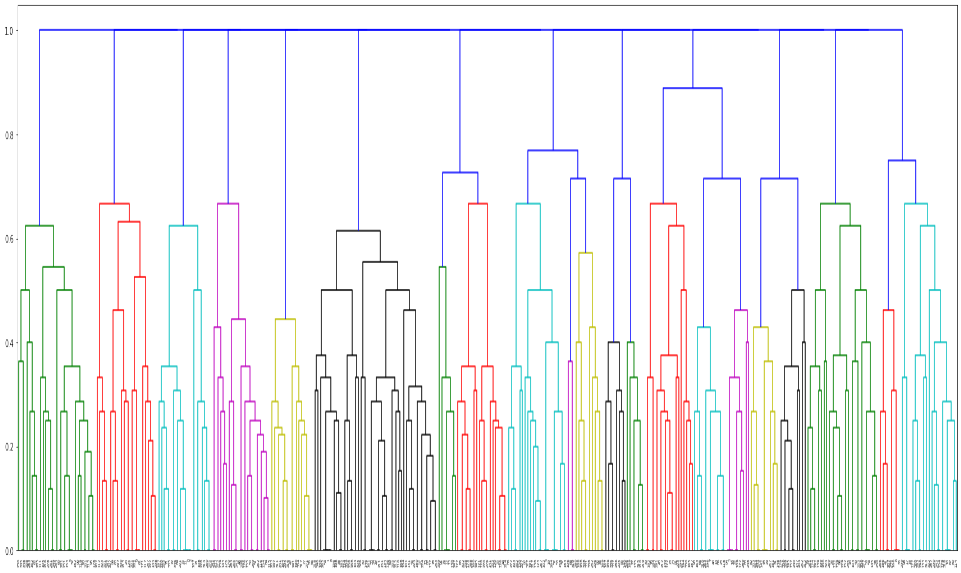

Figure 3: Dendogram plot for agglomerative clustering

2.4.2 Affinity Propagation

This technique is based on the concept of message passing and doesn't require the number of clusters to be set , which is ideal for our work to find the most relevant opinions in our vocabulary.
The most important words identified by the model was : turn_over class acquisition logical direction silver passel resolution friendly passionateness acquirement decisiveness oriented sight leader carry_through morality trying

2.4.3 Ward hierarchical Clustering

This method makes uses of Ward's method for hierarchical clustering , this method defines an objective function and merges clusters based on that objective function's criteria.

The most important words identified by the model was : prayer finding job command appeal promises promise fashioning devising acquisition stay leadership tally determination making deal solving appeals serve run entreaty

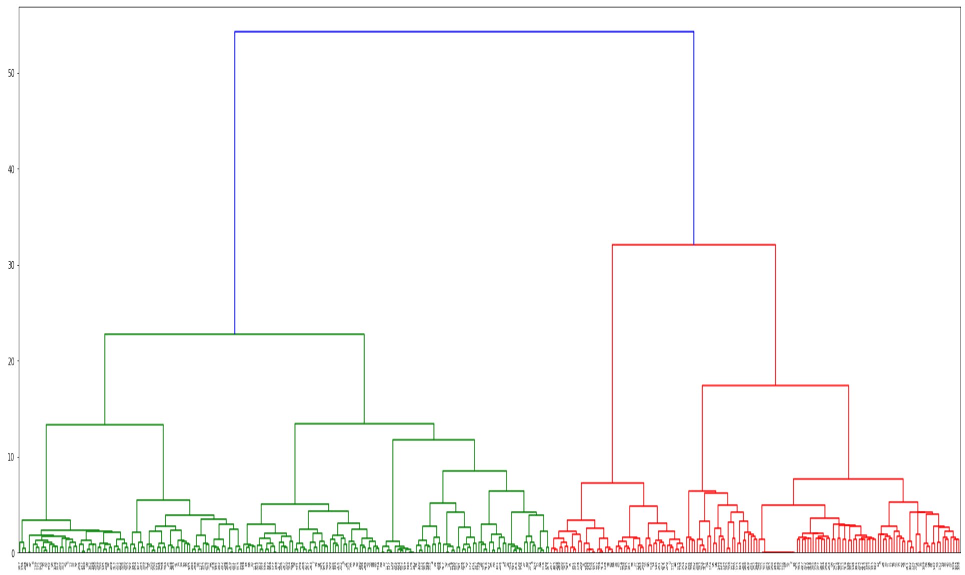

Figure 4: Dendogram plot for Ward hierarchical clustering

2.4.4 KMeans Clustering

The method needs the number of clusters (k) to be defined earlier and clusters the data points based on euclidean distance between 2 points. We find the best number of clusters using the elbow method and find the nearest point to the various cluster centers to be selected as significant opinions from the vocabulary.

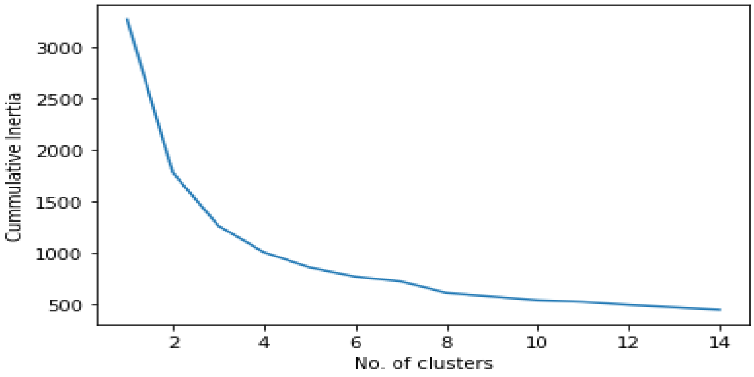

Figure 5: Performance of KMeans method for various no. of clusters

The most significant words obtained from the model were : nerve-racking eloquent honorable vast stressful selfless Brobdingnagian honorable peaceful level-headed

2.5 Word Embedding Methods

2.5.1 Twitter embeddings

The twitter based embeddings was used , each word’s vector was found from the pre trained embedding. The words which aren’t present in the embeddings were discarded.Using KMeans clustering , best no. of cluster were found using elbow method and words close to cluster centers were found.

The following words were found as important opinions words from the vocabulary:
commercialize_duloxetine_hydrochloride bicommunal_bizonal_federation
knowlegeable together implantable_neurostimulation_devices
grassroots_volcanogenic_massive_sulphide isno kindness_gentleness

roofers_painters exprience compro_mise roti_kapda_aur_makaan dishonesty_incompetence hospice_palliative unimpeachable_honesty udner idealized_womanhood thepolitical compromise

2.5.2 Wikipedia and gigaword embeddings

The Wikipedia and gigaword embeddings was used for each word in vocabulary , the embedding vector was obtained.Then KMeans clustering was used to group the vectors and points close to the centers are selected.

The following words were found as important opinions words from the vocabulary:
sort come sure well means kind better rest way thoughtfulness boldness fearlessness amiable fun-loving hot-tempered self-confident open-minded particular

3 Evaluation Criteria

We develop a evaluation criteria using the most important words predicted by the model and the gold standard labels curated by experts and compare them by wup Palmer similarity defined in WordNet.

Model prediction (MP) $\leftarrow [set\ of\ words\ predicted\ by\ model]$
$Goldstandard(GS) \leftarrow [words\ curated\ by\ experts]$
$Similarity\_Score = \sum(i \in MP,\ maximum(j \in GS, return\ wup\_Palmer\_similarity(i,j)))/length(MP)$

We use this similarity score to compare various models and their relative performances.The essential idea of this performance metric is to find the closest word for each word in the predicted words list from the gold standard words. The index of similarity is the wu-palmer similarity as the score for words similar to each other such as leader and leadership should be a given a similarity index of 1 whereas for words leader and captain , which are different words we want a low score and we also get that from wu-palmer similarity metric.Then the sum of all these similarity index is found and normalized by dividing by number of words in predicted list , this would ensure the output is between 0 and 1.

4 Analysis of Results

Using twitter embeddings seems pretty useless , as it tends to associate to words from the twitter language , and this suffers bad predictions. The Wikipedia based embeddings performs better than twitter embeddings but still performs poorly at a score of 0.326 . The kmeans performs poorly as thought as the centroid of a cluster wouldn't understand lexical and grammatical relationships between words , its score it 0.0 . The network methods such as degree and pagerank similarity also perform decently at 0.529 and are able to identify the most important words in the vocabulary by giving weightage to words with larger degree . Although presence of some words makes no sense such as maxborn , Brobdingnagian , stack are present and negative words such as hateful and mess are also predicted . Although these adjectives might be opinion of a fraction of people against the prime minister they aren't the qualities expected from a prime minister they are opinions of a fraction of people and in a opinion mining task they cant be discarded. The three best performing models happen to be cluster frequency , word frequency and ward hierarchical clustering techniques. These models performed very well and predicted opinions such as morality , determination , decision making , equanimity but also words such as idiot and president . The word president doesn't have much relation with the qualities of prime minister but WordNet is a manually made lexical database and is prone to such errors . Also we find the word idiot 's cluster had words such as retard and fool present in them which indicates negative opinions of people on a prime minister. The clustering technique found clusters for each word from vocabulary , discarded the empty ones and combined the ones with overlap.Then frequency of clusters were found , there were 89 clusters and no. of clusters were found such that they maximize the similarity score defined above.

Table 1: Table showing performance of the various models

Model name	Model performance (similarity score)
cluster frequency	0.545
word frequency	0.659
PageRank centrality	0.529
Degree centrality	0.529
Agglomerative clustering	0.421
Affinity propagation	0.388
Ward hierarchical clustering	0.746
KMeans clustering	0.000
Twitter embedding	0.000
Wikipedia embeddings	0.326

5 Discussions and Conclusion

The best performing models were obtained using WordNet based methods , the major advantage of using it is the lexical organization of words and to a good sense of capable to predicting similar words.Whereas the disadvantages include lack of context , doesn't provide a clear distinction between atomic and non-atomic lexical units.In simpler words its hard to distinguish between simple words and multi-words units.Frequency data of words arent available as the frequency of sad and unhappy words could have different frequencies , meaning the less common words isn't known. Better embeddings data set could be made developed in the future , to make best use of embeddings technique. The WordNet models performs well from extracting opinions in small sentences , whereas in larger sentences methods based on parts of speech model can be used to find words with significant positions in terms of opinion can be found. The WordNet does perform well with its lexical structure of words , can be used for opinion mining .

The above models were implemented in jupyter notebooks [4] using the scikit- learn package [5] and NLTK package[6] in python language. Stack exchange has also served as a good resource for debugging Latex and Python scripts. All the codes used for this work can be found at the GitHub repository. Any changes you wish to suggest can be reported on GitHub itself.

References

[1] George A. Miller. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM*, 38:39–41, 1995.

[2] Pavel Smrž. Using wordnet for opinion mining. In *Proceedings of the Third International WordNet Conference, GWC 2006*, pages 333–335. Masaryk University, 2006.

[3] Andrea Esuli and Fabrizio Sebastiani. Pageranking wordnet synsets: An application to opinion mining. 01 2007.

[4] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.