ECE 225 - Text Summarization using Text Rank Algorithm

Meher, Gitika
A53266547

Tyagi, Aditi
A53279284

1 Introduction

Automatic text summarization is the task of producing a brief summary while preserving key information of a huge text corpus. Text summarization can broadly be divided into two categories — Extractive Summarization and Abstractive Summarization. Extractive Summarization relies on identifying the right sentences for summarization and Abstractive Summarization uses advanced NLP techniques to generate an entirely new summary. For the purposes of this project, we implemented the Text Rank Algorithm which is an extractive approach.

Text rank algorithm can generate a summary of text from multiple resources. For the implementation of extractive summarization, we will be using probability techniques to give scores to sentences and extract the most important sentences based on the highest score. Our main purpose is to summarize the different categories containing the most important representative sentences and visualize results based on the summaries.

2 Text Summarization - Significance

Text Summarization is one of those applications of Natural Language Processing (NLP) which is bound to have a huge impact on our lives. With growing digital media and ever growing publishing reading entire articles or books just to see whether they will be useful for a particular task becomes easy with Text Summarization techniques. Automatic Text Summarization is one of the most challenging and interesting problems in the field of Natural Language Processing (NLP). It can be useful while generating a meaningful summary of text from multiple resources such as books, news articles, blog posts, research papers, emails, and tweets.

3 Dataset

To explore the same, we will be working with BBC News Summary Kaggle dataset [1]. This dataset was created using a exisiting dataset for catrgorization project used in D. Greene and P. Cunningham's paper "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering". The dataset is split into different news categories with around 400 - 500 articles from 2004 to 2005. For the purposes of this project, we will be exploring the "Business", "Politics" and "Tech" categories of the BBC news dataset.

4 Data Pre-Processing

To implement the text rank algorithm for this project, we first had to process the dataset given. The following steps were taken to generate sentence vectors:

Conference on Neural Information Processing Systems (NeurIPS), .

1. Split multiple paragraphs into separate sentences. 2. Using stop words from nltk package and removing special characters, we cleaned the sentences extracted from previous step. All the sentences were then lower cased. 3. The words from the clean sentences were then tokenized. Each word from the clean sentences was given a token value based on its embedded wording coefficient. 4. In the end, word was converted to vector representation using glove and finally sentence to vector representation by averaging all the individual word vectors in a sentence.

5 Algorithm and Implementation

TextRank [2]is a general purpose, graph based ranking algorithm for NLP. Graph-based ranking algorithms are a way for deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. When one vertex links to another one, it is basically casting a vote for that vertex. The higher the number of votes cast for a vertex, the higher the importance of that vertex. TextRank finds how similar each sentence is to all other sentences in the text. The most important sentence is the one that is most similar to all the others. Cosine similarity is used as the similarity metric. As mentioned in the data pre-processing section, all sentences are represented as vectors and cosine similarity between two sentence vectors is define as follows: where A and B are sentence vectors representations of two sentences. The cosine similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

values range between -1 and 1 ; -1 representing completely dissimilar sentences and 1 representing completely similar sentences as a semantic distance.

Text Rank is a modification of the Page Rank algorithm applied to text. In order to rank different sentences in the corpus, we would have to compute the similarity score of all sentences with respect to all other sentences. To capture the similarities, we create a square matrix M, having n rows and n columns, where n is the number of sentences. An example with 4 sentences is shown below. This matrix is initialized to the similarity values computed using cosine similarity. All diagonal

M =

	w1	w2	w3	w4
w1				
w2				
w3				
w4				

elements are assigned 0 as this is the similarity between same sentences and it would be otherwise computed as 1. Finally, the values in this matrix will be updated in an iterative fashion to arrive at the sentence rankings. The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation. Finally, a certain number of top-ranked sentences form the final summary. Every vertex's text rank score is calculated iteratively using the update equation below. WS(V_i) represents the text rank score of sentence V_i.The parameter d is a damping factor which can be set between 0 and 1.V_j are all the sentences that have a positive value w_ij for the edge value between sentences V_i and V_j. The damping factor d is included to prevent

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

sinks (i.e. pages with no outgoing links) from absorbing the text rank scores of those sentences connected to the sinks.

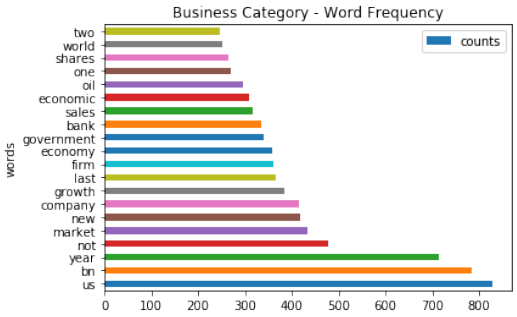

Figure 1: Frequency Count of Business Category

Figure 2: Word Cloud of Business Category

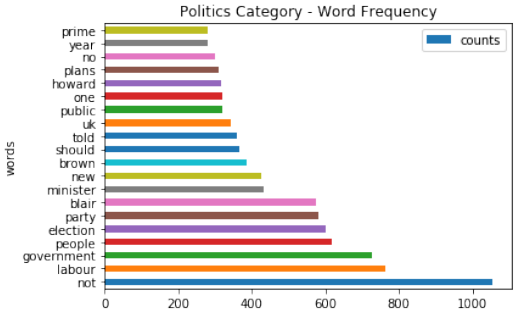

Figure 3: Frequency Count of Politics Category

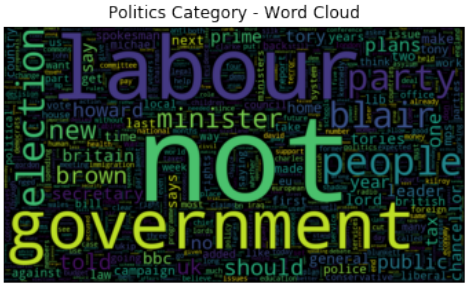

Figure 4: Word Cloud of Politics Category

6 Results and Visualization

For this project, we will be looking at frequency count of repeating words in most important sentences. Moreover, we will be generating the similarity matrix correlation plot to compare the probabilities of similar sentences. Lastly, we will be generating network graph to visually see the most important sentences as nodes using networkx.

The following are the visualization results from the 3 categories of interest:

7 Insights Gained

Using the text rank algorithm for news articles summarization, we were able to extract the most important words and sentences. The most frequently repeated words as seen in the frequency count histograms, are the extracted from the most important sentences from the articles. Words like "Billion" and "US" in the business category are important. Generic sentences with these words in them are of higher importance and have a direct relationship with the results of the gephi and frequency histogram

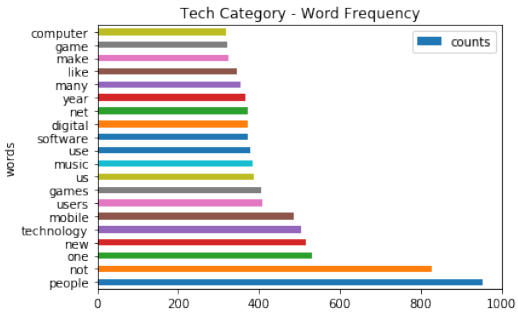

Figure 5: Frequency Count of Tech Category

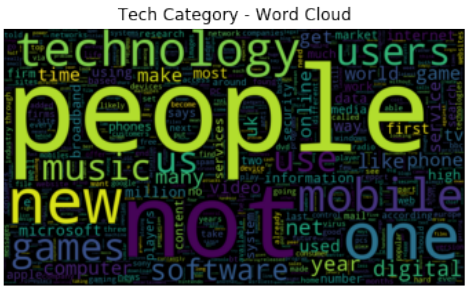

Figure 6: Word Cloud of Tech Category

Figure 7: Gephi plot for Business Category

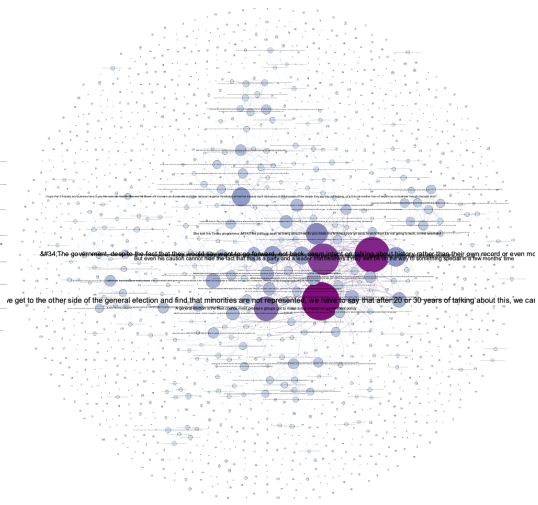

Figure 8: Gephi Plot for Politics Category

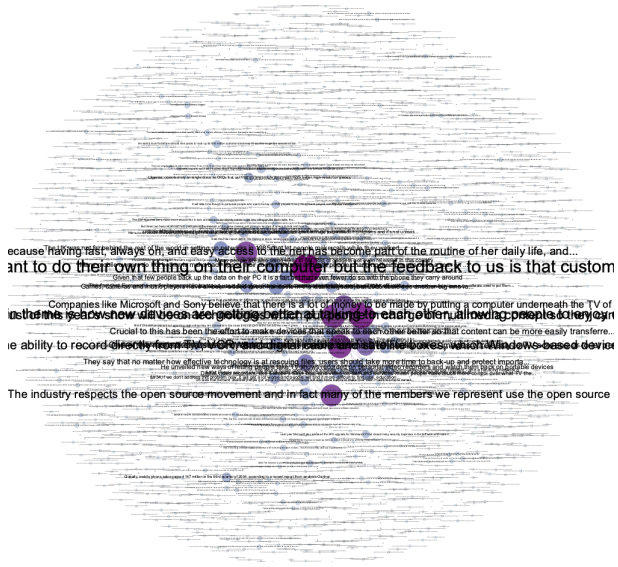

Figure 9: Gephi plot for Tech

plots. Similarly, words like "Labour", "Election", "government" are the most frequent words in the politics category. In the year of 2004 to 2005, elections were the hot topic in the political domain of news and hence these words and sentences (as seen from gephi plot) are the most recurring. Lastly, words like "mobile", "games", "music", "Microsoft" and so on are the most important tech words from the 2005 to 2005 time frame. With new evolving technologies in the mobile, music and digital fields, these words are sentences took dominance over the news articles.

For this project, we used Gephi visualization tool to plot the gml graph of our network. The nodes are sentences. The most important sentences in the nodes with highest degree centrality are noted by darker colors and bigger radius. Those sentences have most recurring words in them and are common in most of the articles from the relative categories. By plotting the network, we were able to define a direct relationship between the frequent words and most important sentences.

References

[1] Bbc news. `https://www.kaggle.com/pariza/bbc-news-summary`, 2016. Accessed: 2019-03-09.

[2] Text rank. https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/, 2017.