Document Summarization using NLP Techniques

Laxmi Niharika Epuri
lxe210008

Yuvasree Dhaya
yxd200003

Venkata Hema Madhuri Vaddella
vxv200052

Sharan Kumar Satish
sxs200317

***Abstract* — Large amounts of structured, unstructured data and semi-structured are referred to as "big data" and it is three dimensional because of the increase in Volume, Velocity and Variety of Data. This rapid increase made it difficult to analyze the data and extract meaningful insights from it. Using Natural Language Processing (NLP) techniques data can be mined for predictive analytics and various other applications such as Chat Bots, Forecasting Demand, Fraud Detection etc. Businesses utilize bigdata analytics to enhance operations, to deliver better customer service, and to develop individualized marketing to make faster business decisions. One such big data application is text summarization that summarizes the original text and generates a meaningful and concise summary. With growth in Digital Media, Text Summarization has become more popular as it saves the time and effort to read the entire article. In this project we implemented Document Summarization using Text Rank algorithm and examined the outcomes obtained.**

Keywords — Document summarization, textrank algorithm, Cosine Similarity, NLP, ROUGE, metrics, and analysis.

I. INTRODUCTION AND BACKGROUND WORK

The technique of extracting a concise text from original text, while maintaining the semantics of the text, is known as summarization. Reading the summary cuts down the amount of time needed to read the entire article content. One of the most interesting problems in natural language processing is document summarization. With the increase in the number of publications, books, and online media; document summarization has become more popular. Due to the abundance of textual data available these days, the demand for text summarization is surging. Text summarization is divided into two subcategories — Extractive Summarization and Abstractive Summarization.
Extractive Summarization: It creates the summary by taking important and significant sentences from the input document. Its main aim is to shorten the article length mainly for the user's convenience.

Abstractive Summarization: It usually involves using NLP techniques and is more challenging than traditional approaches to do document summarization. Here the main goal is to get a relevant summary without worrying about the frequency of the words or the same sentences from the source document. Many researchers and companies are now a days moving towards this approach by using deep learning models and NLP.

II. IMPLEMENTATION OF THE ALGORITHM

In this project, we used TextRank Algorithm for generating the summary. It is an extractive text summarization technique. TextRank algorithm uses PageRank algorithm for summarizing the document. To understand PageRank algorithm let us consider the following example. Let's assume we have 4 web pages like A, B, C and D with hyperlinks. There could be few pages that do not have any inbound links/references. If a web page is referred by many links, then it has higher importance or significance in the network and has higher PageRank score.

TextRank Algorithm Explanation: For ranking phrases within a document, text rank algorithm is used, whereas in page rank we rank web pages. We rank phrases based on the similarity matrix which is constructed based on similarity between two sentences. The probability of a web page transition in page rank is equivalent to similarity score between two texts in the text rank algorithm. Here, we have a square matrix for similarity scores.

TextRank algorithm: It starts with fetching the entire articles content from a public location. Then, for each article, preprocessing is performed where we split the article into sentences, remove stop words and tokenize. For each sentence we find the vector representation using word embedding and then this is used for calculating the similarity between the sentences. We used cosine similarity method for similarity matrix construction. Next, we found the highly ranked sentences using page rank algorithm for summary generation. Finally, to evaluate the relevance of the generated and original summary, rouge metrics are used.

III. DATASET

The input dataset for this project is BBC News summary. It is a zip file with summaries and articles collected across several topics like tech, politics, entertainment, sport etc. The author of this dataset is Mr. Pariza Sharif. The dataset has approximately 2478 articles and from 2004 to 2005. It is 33 MB size dataset.

IV. RESULTS AND ANALYSIS

We use rouge_score and rouge_metric libraries for calculating the performance of the algorithm. ROUGE is widely used for assessing the performance of summarizing texts and documents. It has various metrics and utilities to produce results with different metrics, it checks the generated summary and the original summary for similarities and outputs a score which represents how well the algorithm performed.
In general, there are three ROGUE levels of granularity to evaluate summary:

ROGUE-L - is the Longest Common Subsequence (LCS) of both the generated and original summaries. It uses sequences of words that occur int both the summaries /texts.

ROGUE-1 - is used to check the similarity of words taken one at a time between both the summaries. It is also known as the unigram.

ROUGE-2 - 1s used to check the similarity of words taken two at a time between both the summaries. It is also known as the bigram

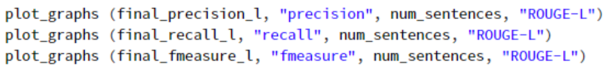
A few criteria can be used to evaluate your summarization system, such as:

Precision, which is calculated as the ratio of the count of key sentences to the count of summary sentences.

Recall is determined by dividing the total number of significant sentences that were present by the number that could be successfully recalled.

Fmeasure is defined as the total words in the generated summary divided by the total words present in the original summary.

After generating the summary and calculating the metrics, we plot graphs for Precision, Recall and Fmeasure based on the number of sentences in text for ROUGE-L, ROUGE-1 and ROUGE-2.



```
plot_graphs (final_precision_l, "precision", num_sentences, "ROUGE-L")
plot_graphs (final_recall_l, "recall", num_sentences, "ROUGE-L")
plot_graphs (final_fmeasure_l, "fmeasure", num_sentences, "ROUGE-L")
```

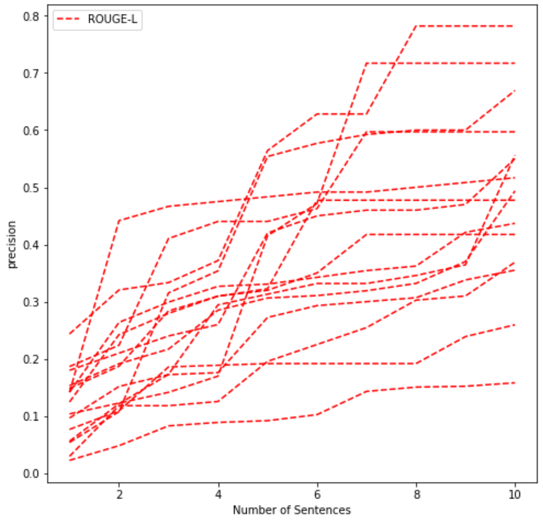


Figure. 1 rouge-L graph for precision

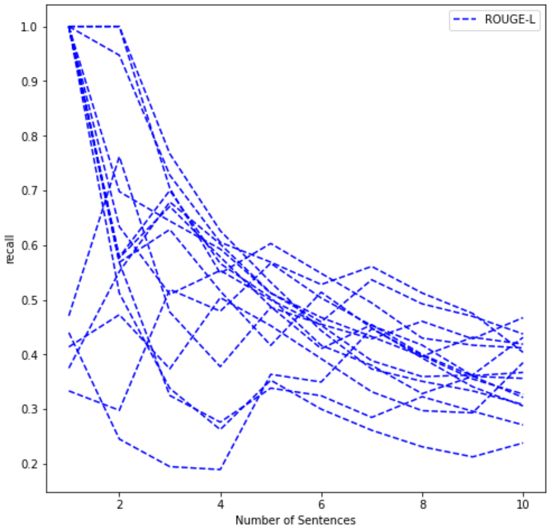


Figure. 2. Rouge-L graph for Recall

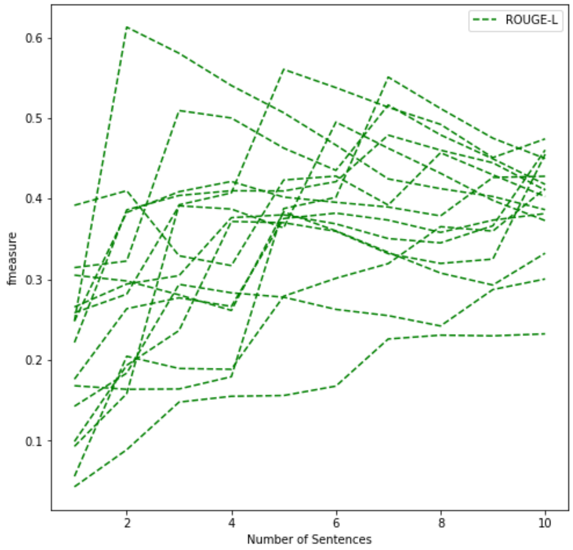
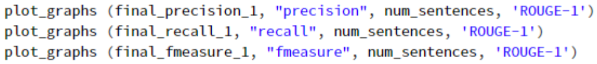


Figure. 3. Rouge L graph for Fmeasure



```
plot_graphs (final_precision_1, "precision", num_sentences, 'ROUGE-1')
plot_graphs (final_recall_1, "recall", num_sentences, 'ROUGE-1')
plot_graphs (final_fmeasure_1, "fmeasure", num_sentences, 'ROUGE-1')
```

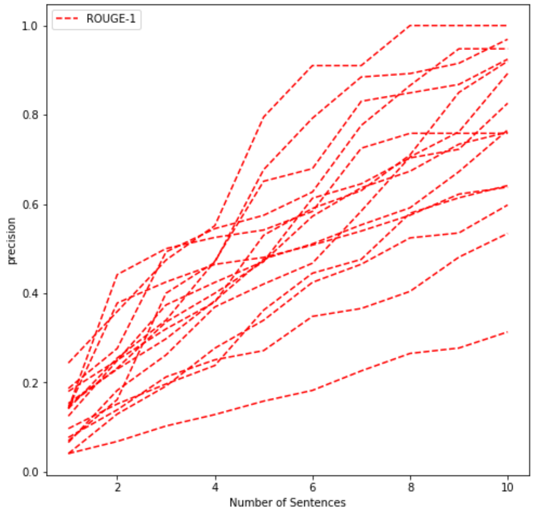


Figure. 4. Rouge-1 graph for Precision

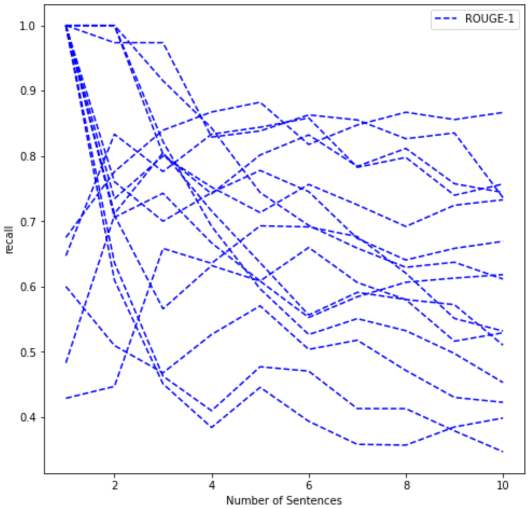


Figure. 5. Rouge-1 graph for Recall

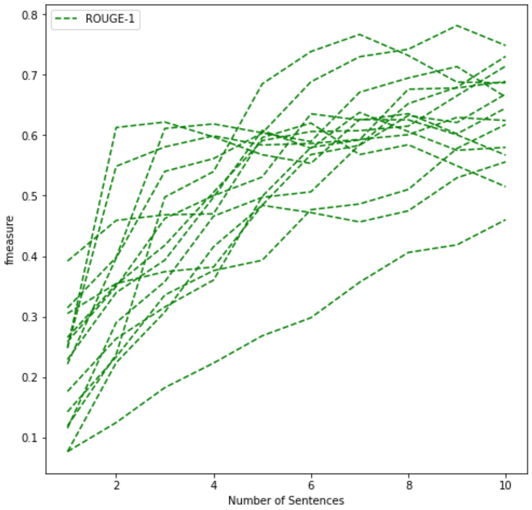


Figure. 6. Rouge-1 graph fmeasure

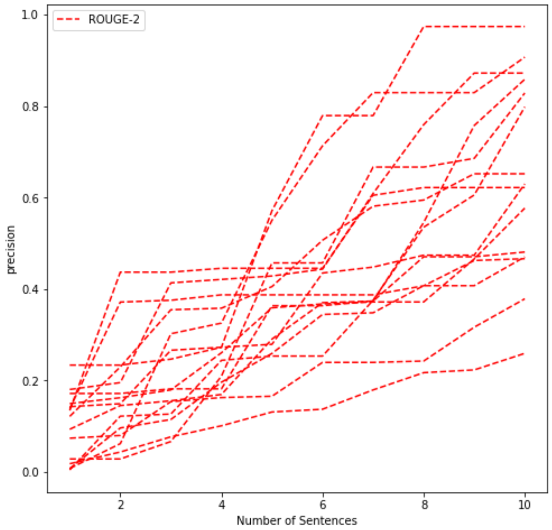


Figure. 7. Rouge-2 graph for Precision

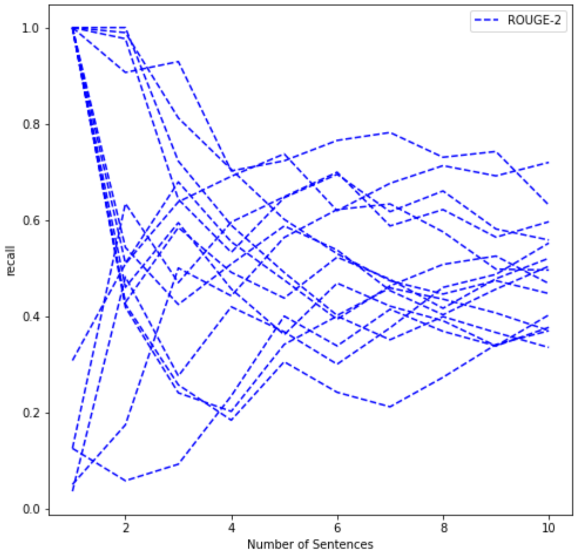


Figure 8: Rouge-2 graph for Fmeasure

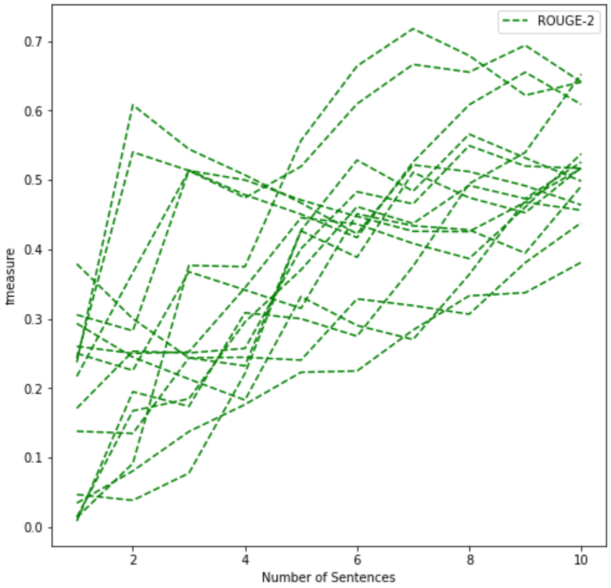


Figure. 9 Rouge-2 graph for Fmeasure

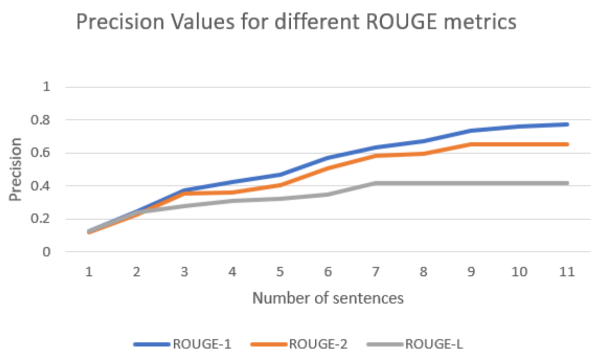


Figure. 10 graph for Precision values for different ROUGE metrics

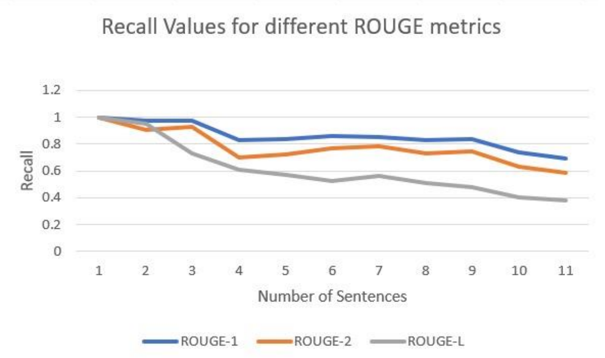


Figure. 11 graph for Recall values for different ROUGE metrics

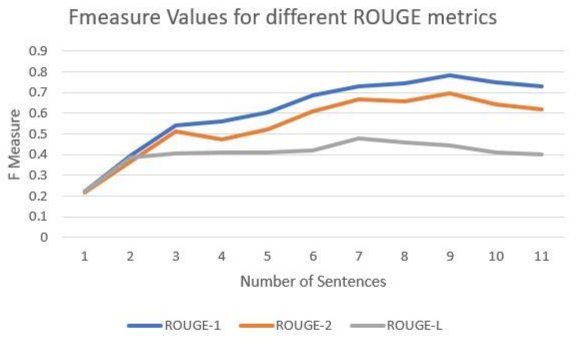
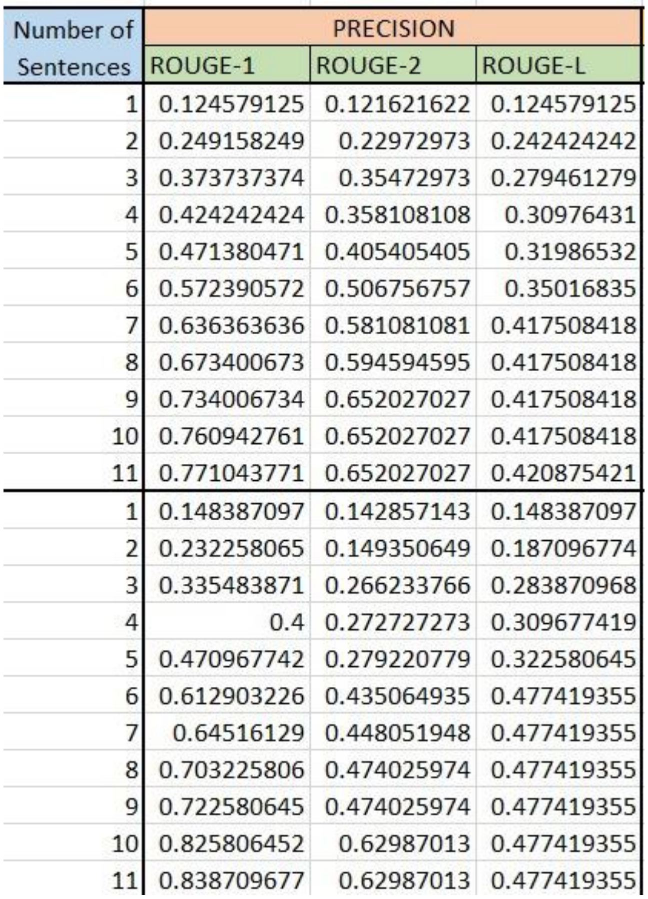
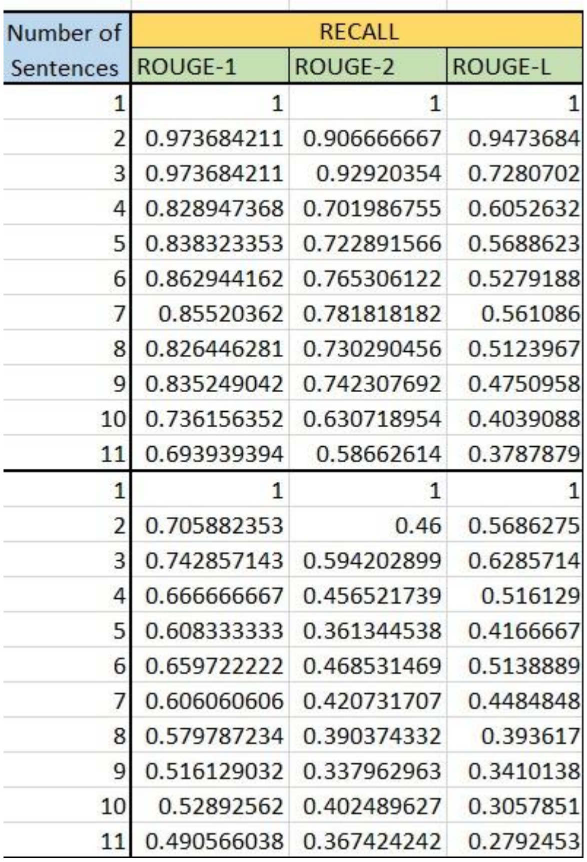


Figure. 12 graph for Recall values for different ROUGE metrics



Number of Sentences	PRECISION		
	ROUGE-1	ROUGE-2	ROUGE-L
1	0.124579125	0.121621622	0.124579125
2	0.249158249	0.22972973	0.242424242
3	0.373737374	0.35472973	0.279461279
4	0.424242424	0.358108108	0.30976431
5	0.471380471	0.405405405	0.31986532
6	0.572390572	0.506756757	0.35016835
7	0.636363636	0.581081081	0.417508418
8	0.673400673	0.594594595	0.417508418
9	0.734006734	0.652027027	0.417508418
10	0.760942761	0.652027027	0.417508418
11	0.771043771	0.652027027	0.420875421
1	0.148387097	0.142857143	0.148387097
2	0.232258065	0.149350649	0.187096774
3	0.335483871	0.266233766	0.283870968
4	0.4	0.272727273	0.309677419
5	0.470967742	0.279220779	0.322580645
6	0.612903226	0.435064935	0.477419355
7	0.64516129	0.448051948	0.477419355
8	0.703225806	0.474025974	0.477419355
9	0.722580645	0.474025974	0.477419355
10	0.825806452	0.62987013	0.477419355
11	0.838709677	0.62987013	0.477419355

Figure. 13 Summary Table for Precision



Number of Sentences	RECALL		
	ROUGE-1	ROUGE-2	ROUGE-L
1	1	1	1
2	0.973684211	0.906666667	0.9473684
3	0.973684211	0.92920354	0.7280702
4	0.828947368	0.701986755	0.6052632
5	0.838323353	0.722891566	0.5688623
6	0.862944162	0.765306122	0.5279188
7	0.85520362	0.781818182	0.561086
8	0.826446281	0.730290456	0.5123967
9	0.835249042	0.742307692	0.4750958
10	0.736156352	0.630718954	0.4039088
11	0.693939394	0.58662614	0.3787879
1	1	1	1
2	0.705882353	0.46	0.5686275
3	0.742857143	0.594202899	0.6285714
4	0.666666667	0.456521739	0.516129
5	0.608333333	0.361344538	0.4166667
6	0.659722222	0.468531469	0.5138889
7	0.606060606	0.420731707	0.4484848
8	0.579787234	0.390374332	0.393617
9	0.516129032	0.337962963	0.3410138
10	0.52892562	0.402489627	0.3057851
11	0.490566038	0.367424242	0.2792453

Figure. 14 Summary Table for Recall

Observations:

1. As we gradually increase the number of sentences of the summary the value for Recall tends to decrease as the possibility of having the same words increases.
2. As we increase the sentences in the summary that we generate, Precision values increase because adding more words and lines to the summary generates summary that is quite similar to the original summary in the dataset.
3. Results obtained through ROUGE-2 I.e., bigram, are lesser than ROUGE-1 I.e., unigram since finding two consecutive words in both the reference summary and algorithm generated summary is very less in terms of Recall, FMeasure, Precision.
4. ROUGE-1 results for precision, recall and fmeasure tend to be better since they just find matches with single words in both the reference summary and the summary produced by the algorithm.
5. FMeasure or the F1 score calculated, and it tends to increase upto 9 sentences and reduces as the number of sentences increase.
6. Precision values and the recall values vary for all the models as the number of sentences increase.
7. Precision values for ROUGE-L increase upto 4 and flatten as we increase further.

V. CONCLUSION AND FUTURE WORK

Research for text summarization still goes on for accurate description of evaluating a summary from the content of the documents. Precision and accuracy can be increased using Recurrent neural networks. There are many applications of text summarization that impacted human life. Particularly an app called InShorts which runs through a lot of data and produces a short summary of the entire news article in short paragraphs of about 43 to 56 words. Chat Bots assist with customer service and interaction, Grammarly to help correcting the grammar mistakes in any document, Survey's that help in company's feedback and much more. Hence, Text Summarization plays an important role in the field of Data Science and scientists are still working on improving the metrics.

Future work: Textrank algorithm might skip the vital keywords due to its lower frequency of occurrence. We can solve this problem by considering semantic similarity of the words, we plan to use different word embedding techniques and evaluate its impact.

VI. REFERENCES

[1] Md. Majharul Haque , Suraiya Pervin , and Zerina Begum, "Literature Review of Automatic Single Document Text Summarization Using NLP".

[2] Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, Zheng Chen, "Document Summarization using Conditional Random Fields".

[3] Ziqiang Cao, Wenjie Li, Sujian Li, Furu Wei, "Improving Multi-Document Summarization via Text Classification".

[4] Kaiz Merchant, Yash Pande, "NLP Based Latent Semantic Analysis for Legal Text Summarization".

[5] https://www.analyticsvidhya.com/blog/2020/07/top-10-applications-of-natural-language-processing-nlp/