Analysis of Text Summarization using LSA and TextRank

Vivek Kapa
ECSS
University of Texas at. Dallas
Dallas,U.S.A
VXK200011@utdallas.edu

Vinutha Kapa
ECSS
University of Texas at Dallas
Dallas, U.S.A
VXK200014@utdallas.edu

Saipriya Nimmagadda
ECSS
University of Texas at Dallas
Dallas, U.S.A
SXN200007@utdallas.edu

Yash Niraj Majmudar
ECSS
University of Texas at Dallas
Dallas, U.S.A
YNM210000@utdallas.edu

Abstract - Time is very important to man. To reduce the amount of time by reading long texts and notes for research, selection processes and other purposes, document summarization has been put to practice. The task of extracting important and crucial information from single or multiple sources and creating a mini document for a specific purpose is called summarization of documents. These techniques are of two types 1) Extractive 2) Abstractive techniques. This report tries to show the analysis of performing extractive text summarization using two most known techniques called Latent Semantic Analysis (LSA) and TextRank algorithm. Recall Oriented Understudy for Gisting Evaluation will be put into use to analyze the performances between these two techniques on a CNN daily mail dataset. For analysis, ROUGE, Recall and precision are calculated and compared. Lastly, a conclusion on which among the two techniques perform better on the given dataset is provided based on the analysis.

Keywords — LSA Algorithm, TextRank algorithm, ROUGE

I. INTRODUCTION

Big data has 3Vs: Velocity, Variety, Volume. These V's make us familiar with, data is getting produced at what speed, amount of data generated, variety of data that is present. Also, with vast amount of data present in cloud platforms, data can now be accessed anywhere, anytime and by anyone. Now the real problem lies in choosing right content from array full of resources. For various reasons this very large text should be summarized in such that way only the important information is emphasized, and the user can easily skim the article more effectively than the reading it entirely. Text summarization is a technique of obtaining the keys and phrases from native content without human intervention. Text summarization has the following advantages:

- Reduces the time spent by the user on the data
- Reduces the space in the storage
- Summaries make text selection process simple and easy
- Automatic summarization techniques are less biased when compared to human summarizers

Main purpose of this project is to develop an algorithm that will do text summarization of the given input. Here, Latent Semantic Analysis (LSA) and TextRank algorithm are used to develop the algorithm and then comparison of the results generated by both the models is done using evaluation metrics. To provide summary of the given input paragraphs, LSA utilizes single value decomposition technique to figure out the hidden connection between terms and concepts. One of the major advantages of using LSA is that it uses reduced representation which helps in removing some 'noise' from the data.

We have used nltk, python, boto3, sklearn, bs4, pandas and some other associated libraries in Databricks community edition for the development of both of our algorithms. Our project will be useful for getting accurate text summarizations from a given input. We will also compare the accuracy of LSA and TextRank at the end of this paper.

This paper has a total of six sections, a brief INTRODUCTION as to what this paper will talk about is mentioned in section I. Section II BACKGROUND WORK gives an overview on the literature study, where various models that were used for text summarization are talked and discussed about. This project primarily discusses extractive output-based summarization algorithms such as LSA and TextRank.

THEORETICAL AND CONCEPTUAL STUDY, Section III of this paper, gives an overview about how algorithms like LSA and TextRank, graph-based approaches work. Brief working of ROUGE, Recall, Precision and F-measure is also provided. DATASET AND IMPLEMENTATION, Section IV of this paper, talks about our dataset and text prepossessing using tokenizers. In Section V, RESULTS AND ANALYSIS, we show the results generated by each algorithm in a table. Finally, Section VI, CONCLUSION AND FUTURE WORK, provides a conclusion on which algorithm is better and talks about the scope of future work of this project.

II. BACKGROUND WORK

A brief history about the evolution of techniques in text summarization has been studied. We can see the use of NLP from world war -II where a Machine Translation was used for translating Russian into English language and vice-versa. And till date, we can see great evolution in the way text summarization has been performed [1]. An evolution has been made from generating simple extracts from single document (usually in English language) to generating more sophisticated extracts and summaries from multiple documents in various languages.

From the above developed techniques, two of the most famous techniques are LSA [2] and TextRank. LSA is an algebraic + statistical approach that is used to extract hidden structures between terms of a document. LSA is a combination of NLP with unsupervised learning. LSA is known for extracting these hidden features which usually cannot be mentioned directly. LSA helps to reduce the dimensionality of the original text by

using reduced representation. One of the major applications of LSA is in its assistance for performing prior art searched for patients.

TextRank algorithm on the other hand is an unsupervised learning method that will find similarity between sentences while doing automatic text summarization. It is believed that TextRank was inspired from Page Rank. Every sentence is converted into a vector representation. Similarity matrix is used for storing the similarities between sentence vectors. Using this matrix, a graph is formed, where sentences are vertices and edges are used for similarity scores. Top ranked sentences are picked and which for the summary.

One of the major drawbacks of TextRank is that it tends to neglect the information about the structure and context. In [3], an alternative to overcome these shorting is done by using highly scalable iTextRank that takes into consideration more of the semantic and structural information of the input. This method has high accuracy and low recall when compared to TextRank.

III. THEORTICAL AND CONCEPTUAL STUDY

A. *Output-based summerization*

- *Abstractive*: In this technique, the text is understood at a comprehensive level. Summarization is made by paraphrasing the original text and generating new sentences from it, instead of fetching important sentences from the original dataset. New sentences constructed like this will provide a condensed version of the native text. Abstractive summarization is very similar to the summarization done by humans. This way of summarization provides the user with concise, non-redundant and information rich summaries. Abstractive summary is more like a summary written with pen.

- *Extractive*: In this technique, terms and expressions are picked up and added into extracted summary. Topmost relevant sub-sentences from the original text are chosen using algorithms which are used to rank relevance between terms or expressions. This technique basically identifies salient information extracts it to form a concise summary. Extractive summaries basically work as a highlighter of the original text.

B. *Content-based summerization*

- *Query-based method*: In this technique, given a search query, a summary of the document gets generated that basically answers the search query. One of the many applications of query-based summarization is search engines and chatbots.

- *Generic method*: In this technique, the user does not have any prior knowledge about the topic. The access to the native document is also not given to user. Generic summarization emits a broad synopsis of the input dataset by providing the input text's gist that is as clear as possible.

C. *Number-based summerization*

- *Single document*: In this technique, one document is given as input and a paragraph is generated retaining all the relevant terms from the input.
- *Multiple document*: In this technique, from more than one document a summery is produces that keeps similar or relevant information and filters out all the differences between the documents to overcome the redundancy factor. Here all the documents are summarized to produce one single short piece of summary.

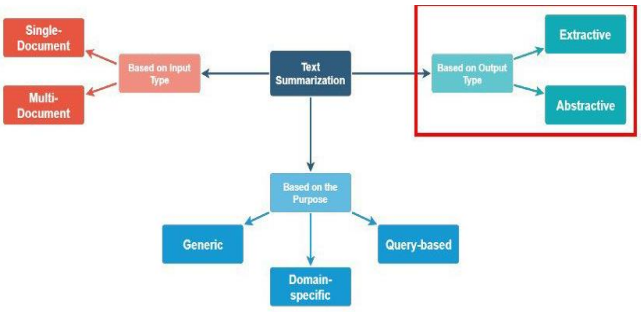


Fig: Types of text summarizations

D. *Latent Semantic Analysis(LSA)*

LSA [4], an extractive summarization approach, that uses unsupervised learning to understand hidden connection between terms and expressions and produces the report summary. The phrases containing linguistically significant terms are selected by applying singular value decomposition (S.V.D) to the term document frequency grid [5].

With the application of the algebraic technique, S.V.D, a sizable portion of the input data is represented as a grid with the count of various words present in data. The grid is designed in such a way that each column represents a document, and each row represents the distinct words present in the that document. For words in sentence t, present in the document, B_y stands for the t's weighted words as a result we can get a vector $B = [B_1, B_2..., B_n]$. B is a l x m matrix in a text that contains l words and m phrases. The S.V.D of a l x m matrix B, where l > m, is defined as a matrix where R is a m x m orthogonal matrix, Q is a l x m column-orthonormal matrix, and I is a n x n diagonal matrix. Multiplying three additional matrices, namely Q, and RT, will resemble the initial feature grid of LSA.

Steps followed using SVD are:

1. Decompose document M into single distinct sentence, and create S and initialize t=1
2. Next step is to form the terms by sentences matrix B for M
3. SVD is performed on B, to obtain singular value matrix μ, and right singular vector matrix R^T.
4. t^{th} right singular vector from matrix μ is selected.
5. Sentence having largest value with t^{th} right singular vector is added to summary.
6. If (the predefined number t is reached):
   Terminate operation
   Else:
   t = t+1
   go to step 4

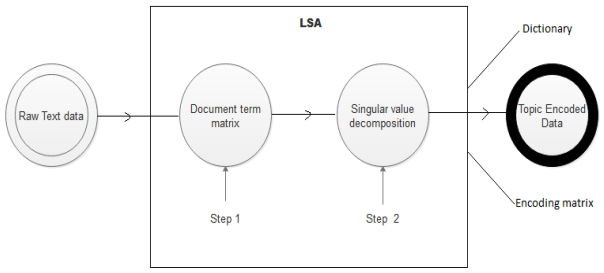


Fig. LSA processing

E. *Recall-Oriented Understudy for Gisting Evaluation (ROUGE)*

ROUGE [6] is used to analyze performance in automatic text summarization of texts and machine translations. The way ROUGE works is that it tries to compare the automatically produces summary against a reference summary. These reference summaries are usually produced by humans. Using the overlapping words, we calculate the precision and recall values. This is done to get a good quantitative value. ROUGE is used to get linguistic parallelism. It does not consider words that have same meaning but different spelling.

In ROUGEs language:
Recall is about the amount of produced summary that is caught by the reference summary
Recall is:

$$\frac{number\_of\_overlapping\_words}{total\_words\_in\_reference\_summary}$$

Precision on the other hand shows the user what degree of the retrieved summary is applicable.
Precision =

$$\frac{number\_of\_overlapping\_words}{total\_words\_in\_system\_summary}$$

To get more concise summary, we can join precision with f-measure. F1 score is:

$$2 * \frac{precision * recall}{precision + recall}$$

The following types of ROUGE take granularity of texts that will get compared between reference summary and automatically produced summary.

ROUGE-N calculates n-gram overlap. Example: ROUGE-1 provides details about the unigrams overlapping and ROUGE-2 provides details about the bigrams overlapping These are calculated between the reference and automatically produces summary.

ROUGE-L measures LCS (longest common subsequence) that gives back the level order of the sentance.

ROUGE-S is also called skip-gram concurrence. It considers any two words consecutively in a data while allowing gaps between them.

F. *TextRank Algorithm*

TextRank Algorithm is inspired by PageRank. Page rank is used primarily for ranking the web pages that are produced because of online search queries.

Each item in this method is presented as a collection of papers that are hyperlinked, which act as a weight. These weights indicate how important it is in relation to the other documents in the set. The rank score is determined by considering both the inlinks and the outlinks from each content. The document with maximum incoming and outgoing connections is picked up as the page with highest rank. There exists a cutoff for repetitions, it is assumed the as the number increases, the precision also increases. The top scoring phrases are merged into the abstract depending on phase ranks. PageRank provides the best models with a specific number of phrases.

In PageRank algorithm and TextRank algorithm similarity is that we use sentences in place of web pages, similarity between two sentences is same as transition probability between two web pages and both PageRank for TextRank store the similarities in a n*n matrix

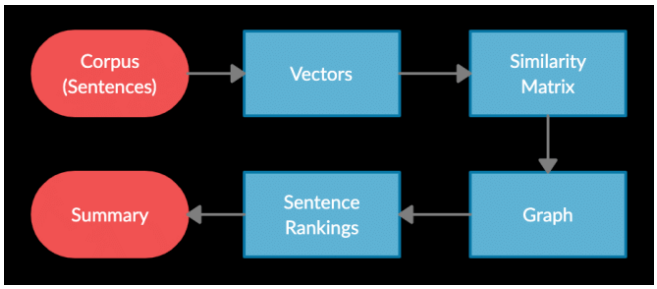


Fig: Working of text rank algorithm

IV. DATASET AND IMPLEMENTATION

Both the algorithms, LSA and TextRank are performed on the same dataset that is present inside '.story' format. We have used non-anonymized dataset [7] of cnn-dailymail. We are using a modified version of it, which is suitable for summarization. The dataset has numerous articles from both CNN and daily mail, and we are majorly using the CNN. story files. Each file consists of the highlights of the story as well. There are about 287,113 instances of the train, 11,490 instances of test, and 13368 instances of validation data in the dataset and the average token count being 781 for the articles and 56 for the highlights. For the time being, we have randomly selected a few articles out of them and stored them in AWS S3 which are proposed to be used as input for models The dataset file was stored in AWS S3 bucket and can be found at [8]

A. *TextRank Implemntation*

1. Firstly, import all the required libraries
2. Use GloVe which is an unsupervised algorithm that can be used for producing the word embedding and its fundamental premise is to extrapolate the link between the words using statistics. Taking the input for the article and the summary from the AWS S3.

3. Apply punkt sentence tokenizer on the article
4. Remove stop words and get cosine similarity
5. The sentences having higher ranks make summary, so we sort the list accordingly in descending order

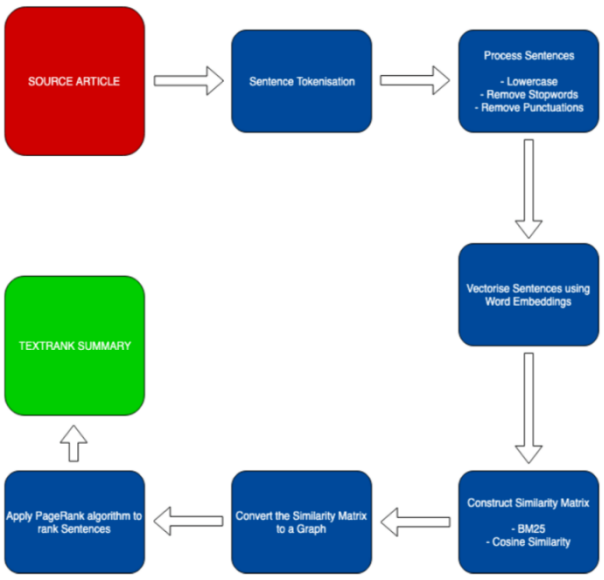


Figure 3.3: TextRank - Implementation process

B. Latent Semantic Analysis (LSA) Implementation

1. Firstly, import all the required libraries
2. Get dataset from the amazon s3 bucket
3. Purify the data by removing the missing values, special characters, stop -words from the NLTK library
4. Calculate SVD
5. Generate output generated by utilizing the diagonal matric and then keywords are used to form the phrase

V. Observations and Results

ROUGE-L, ROUGE-1 and ROUGE-2 have been calculated along with recall, precision, and f-measure for different test cases for each algorithm and is presented below.

Values calculated for TextRank algorithm

DataColumn	ROUGE-L	ROUGE-1	ROUGE-2
1	P:0.006578947368421052 R:0.1 F:0.012345679012345678	P:0.006578947368421052 R:0.1 F:0.012345679012345678	P: 0.0 R:0.0 F: 0.0
2	P:0.039473684210526314 R:0.17647058823529413 F:0.06451612903225806	P:0.05921052631578947 R:0.2647058823529412 F:0.0967741935483871	P:0.01324503311258278 1 R:0.06060606060606061 F:0.021739130434782608
3	P:0.046052631578947366 R:0.14285714285714285 F:0.06965174129353234	P:0.07894736842105263 R:0.24489795918367346 F:0.11940298507462688	P:0.01324503311258278 1 R:0.04166666666666664 F:0.020100502512562814
4	P:0.052631578947368 42 R:0.11940298507462686 F:0.07305936073059360	P:0.10526315789473684 R:0.23880597014925373 F:0.14611872146118721	P:0.01324503311258278 1 R:0.03030303030303034 F:0.01843317972350230 4
5	P:0.065789473684210 52 R:0.11494252873563218 F:0.08368200836820083	P:0.11184210526315 79 R:0.19540229885057472 F:0.14225941422594143	P:0.01324503311258278 1 R:0.02325581395348837 2 F:0.01687763713080168 6

Values calculated for LSA algorithm:

DataColumn	ROUGE-L	ROUGE-1	ROUGE-2
1	P: 0.046052631578947366 R: 0.4117647058823529 F: 0.08284023668639053	P: 0.05921052631578947 R: 0.5294117647058824 F: 0.10650887573964497	P:0.01986754966887413 R: 0.1875 F:0.03592814371257485
2	P: 0.059210526315789 47 R: 0.31034482758620 69 F: 0.09944751381215468	P: 0.09868421052631579 R: 0.51724137931034 49 F: 0.16574585635359115	P:0.02649006622516556 3 R:0.14285714285714285 F: 0.04469273743016766
3	P: 0.092105263157894 73 R: 0.21212121212121213 F: 0.12844036697247707	P: 0.15131578947368 42 R: 0.34848484848484 85 F: 0.21100917431192662	P: 0.033112582781456956 R: 0.07692307692307693 F: 0.04629629629629630
4	P: 0.092105263157894 73	P: 0.18421052631578946	P: 0.039735099337748346 R:

	R: 0.181818181818181818182 F: 0.122270742358078 6	R: 0.363636363636365 F: 0.244541484716157 2	0.0789473684210526 3 F: 0.0528634361233480 1
5	P: 0.125 R: 0.202127659574468 07 F: 0.154471544715447 13	P: 0.223684210526315 8 R: 0.361702127659574 5 F: 0.276422764227642 34	P: 0.0529801324503311 26 R: 0.0860215053763440 9 F: 0.0655737704918032 8

Summary generated by TextRank Algorithm:

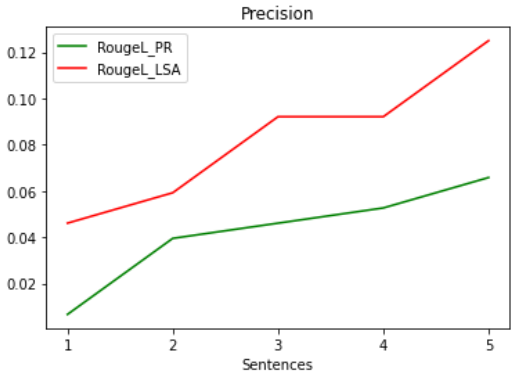
"majority people brought Rouhani power 2012, hoping would ease sanctions. Iran's first Supreme Leader made bitter peace Iraq's Saddam Hussein Nazila Fathi: current leader Ayatollah Khamenei trying put Iran right track dies? Whether Khamenei seriously ill not, many believe advanced age, needs put country straight path death. Iranian politicians divided moderates led President Hassan Rouhani, wants develop economy hardliners, view deal threat regime's ideology. Iran finalize deal June United States P5+1 partners, international investors, including Americans, would able invest Iran first time decades."

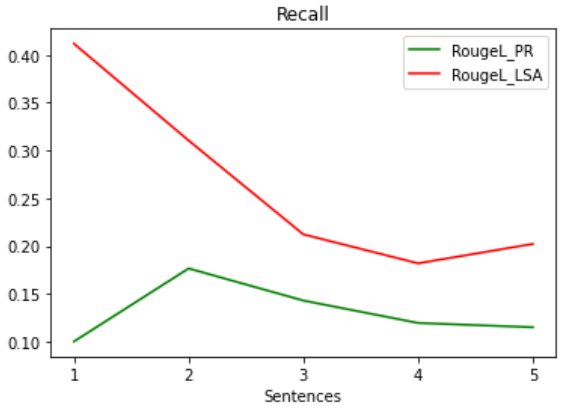
Summary generated by LSA model:

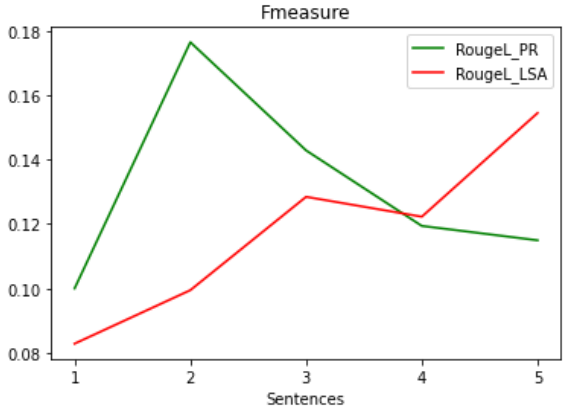
Like the war, Iran's defiance to halt its controversial nuclear program has defined Khameni's era. He has defied the West in the face of increasing economic pressure. Iran claims that its program is peaceful, but Khamenei's refusal to end uranium enr ichment activities -- a process that can lead to making nuclear fuel as well as nu clear bombs -- has landed the country und er crippling sanctions. The standoff with the West has stretched longer than the wa r.The agreement reached in Lausanne, Swit zerland, however, marks a new chapter in the history of the country

Graphs comparing ROUGE values between LSA and TextRank:

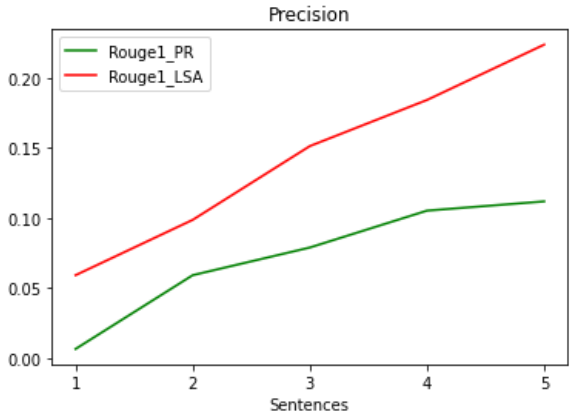
A. ROUGE-L values

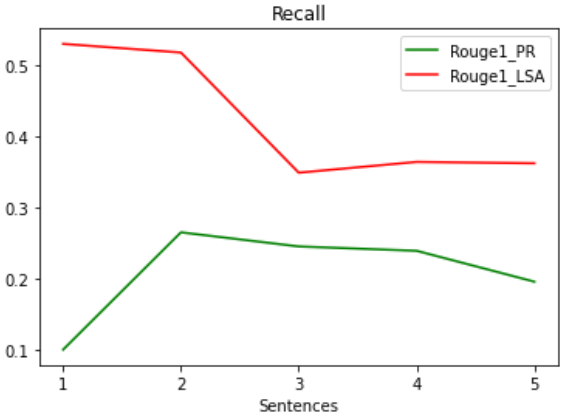




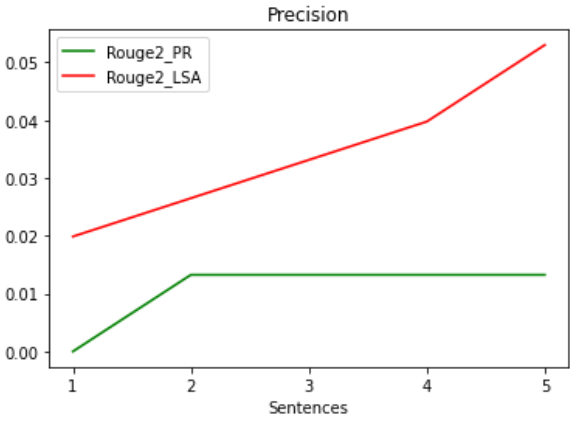


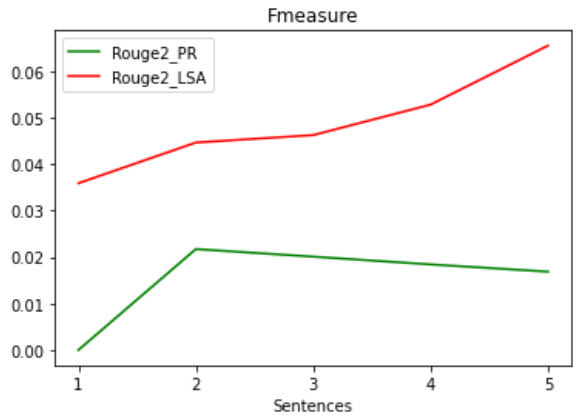
B. ROUGE-1 values





C. ROUGE-2 values





VI. Conclusion and Future Work

We have successfully implemented two extractive algorithms LSA and TextRank. In our experiment, from the above graphs and tables, based on ROUGE recall, precision and F1 score values we can see that, for ROUGE-L, ROUGE-1 and ROUGE-2, we can see that precision and F1-measure is high for LSA when compared to TextRank. To analyze and compare the extracted summary, a mix of precision and F1 measure gives a best result. This proves that LSA does better summarization than TextRank for our database.

In both the models, the original data was preprocessed using tokenizers, stop words and other libraries. It was understood how LSA selects top N documents depending on the weight. It was also understood how NLP using t^{th} singular vector, finds the most relevant sentence. Corresponding to their values these vectors are presented in descending order. It was also understood that we get minimum redundancy in LSA because these singular values are independent as a result, we get minimum redundancy. TextRank and its process has also been studied.

This project can be further extended by implementing another version of text Rank as mentioned earlier in the introduction called iTextRank and then comparing its results with LSA. ROUGE has its own disadvantages like ROUGE does not consider words that have same meaning but different spelling. As a result, we can always use other evaluation metrics like cosine similarity and then compare the results.

VII. References

[1] Lloret, Elena. "Text summarization: an overview." *Paper supported by the Spanish Government under the project TEXT-MESS (TIN2006-15265-C06-01)* (2008).

[2] Ozsoy, M. G., Alpaslan, F. N., & Cicekli, I. (2011). Text summarization using Latent Semantic Analysis. Journal of Information Science, 37(4), 405–417. https://doi.org/10.1177/0165551511408848

[3] Yu, Shanshan, Jindian Su, Pengfei Li, and Hao Wang. "Towards high performance text mining: a TextRank-based method for automatic text summarization." *International Journal of Grid and High Performance Computing (IJGHPC)* 8, no. 2 (2016): 58-75.

[4] Evangelopoulos, Nicholas, Xiaoni Zhang, and Victor R. Prybutok. "Latent semantic analysis: five

methodological recommendations." *European Journal of Information Systems* 21, no. 1 (2012): 70-86.

[5] Y. Gong, X. Liu: Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, United States 2001, pp. 19-25

[6] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[7] https://www.tensorflow.org/datasets/catalog/cnn_dailymail

[8] https://vinnubigdata.s3.us-west-1.amazonaws.com/Project/story/0e0bdf77a264313fa9c706e2a02ce33b42e6494d.story

[9] https://www.kdnuggets.com/2019/01/approaches-text-summarization-overview.html

[10] https://www.sciencedirect.com/science/article/abs/pii/S0957417421003870?via%3Dihub