Text Summarization

Kotabagi, Aditi Mahabaleshwar
Erik Jonsson School of Engineering
University of Texas at Dallas
Dallas, U.S.A
aditi.kotabagi@utdallas.edu

Enumudi Venkatesha, Nagarjun
Erik Jonsson School of Engineering
University of Texas at Dallas
Dallas, U.S.A
nagarjun.enumudivenkatesha@utdallas.edu

Kumar, Thanuja Shailendra
Erik Jonsson School of Engineering
University of Texas at Dallas
Dallas, U.S.A
thanuja.kumar@utdallas.edu

***Abstract*—Text summarization is the technique of condensing a text document while preserving key details and the primary ideas of the original text. Automatic text summarization becomes a crucial tool for quickly and easily extracting precise information from enormous amounts of text. The two types of text summarizing techniques are extractive and abstractive. An extractive text summarization refers to the extraction of significant text from a text file or the source document. An abstractive text summarization is the process of creating a brief and succinct summary that encapsulates the key concepts of the source text. The generated summaries might include new words and phrases that aren't in the original text. In this paper, a method to perform an extractive text summarization on a document is demonstrated.**

Keywords—text summarization, N.L.P, latent semantic indexing

I. INTRODUCTION

Extractive text summarization has received a lot of attention recently due to the problem of information overload that has arisen in the age of the World Wide Web. When readers prefer shorter versions, they are swamped with lengthy text materials. In order to address this problem, we require a system that can instantly produce summaries from the provided source of text. The process of distilling a document's primary ideas takes time. Extracting only the relevant information from a big volume of data proves to be beneficial to people in all fields. This can be made possible by automatic text summarizing. The goal of this approach is to distill the lengthy text into concise summaries without distorting the underlying data and its meaning. Additionally, it guarantees that the summary has all of the crucial information.

There are two main methods for automatic text summarization- extractive method and abstractive method. The process of Extractive text summarization involves selecting the main phrases from the source text to condense it's meaning and generating them verbatim. As a result, they rely solely on sentence extraction from the original text and do not add or change texts. Abstractive summarization methods, on the other hand, aim to generate important content in a novel way. They interpret and analyze the text using advanced natural language techniques to create a new shorter text that conveys the most important information from the original text. The majority of text summarization research has recently been on extractive summarization, despite the fact that summaries written by humans are typically abstractive. When compared to automatic abstractive text summarization methods strictly extractive summarization methods frequently yield better results. This is because, in contrast to data-driven approaches like sentence extraction, abstractive summarization methods deal with issues like semantic representation, inference, and verbal fluency that are more challenging. In actuality, there isn't a fully abstractive summarization system available right now. To create the abstract of the text, existing abstractive summarization methods frequently use an extractive preprocessing component.

In this paper, we concentrate on extractive summarization techniques and provide a brief overview of some of the most popular methods in this area. Extractive summarization consists of three main tasks

First, Construction of an intermediate representation of the input text. There are two categories for construction of an intermediate representation of the input text: topic representation and indicator representation. Topic representation interprets the topic(s) covered in the text by transforming it into an intermediate representation. The methods used for this are classified as frequency-driven approaches, topic word approaches, topic word approaches, latent semantic analysis, and Bayesian topic models based on how complex they are. Every sentence is described by indicator representation as a formal property(indicators) that are significant, such as sentence length, place in the document, the presence of specific phrases, etc.

Second, each sentence is given an importance score when the intermediate representation is constructed. According to topic representation techniques, a sentence's score reflects how well it describes some of the most significant textual themes. The data from various weighted indicators is aggregated to create the score in indicator representation.

Third, to create the summary, the summarizer algorithm picks the top k sentences. To choose the most important sentences, some methods use greedy algorithms. Other methods turn sentence selection into an optimization problem, choosing a group of sentences while keeping in the mind the requirement that the collection should minimize redundancy and maximize overall importance and coherency

The method used in this paper to construct the intermediate representation of the input text is the Latent Semantic Analysis (LSA). Latent Semantic analysis is a technique in natural language processing, in particular distributional semantics, of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms [1]. LSA works on the assumption that the words that are close in meaning will occur in similar pieces of text called the distributional hypothesis. First, Singular value decomposition

(SVD), a mathematical technique is used to condense a large piece of text into a matrix containing word counts per document where rows represent unique words and column represent each document. This technique reduces the number of rows while maintaining the similarity structure among columns. The cosine similarity between all the columns is then used to compare the documents. Documents with values close to 1 are considered to be highly similar, while those with values near 0 are considered to be very different. If the diagonal matrix is multiplied with weights with the topic sentence matrix, the result will reflect the measure of similarity a phrase represents a topic.

II. RELATED WORK

The first work in the field of Automatic text summarization was published in 1957 by Hans Peter Luhn [2] and is used as a statistical approach. In the year 2015, there was a substantial increase in research. By the year 2016, term frequency-inverse document frequency was in use. By 2016, the most effective method for summarizing multiple documents was pattern-based summarization. The combination of LSA and non-negative matrix factorization outperformed the pattern-based summarization in the year 2017 [1]. By the year 2019, machine learning methods dominated and augmented the extractive summarization of single documents, even though they did not completely replace previous approaches, but were frequently integrated with them. In the recent times, the studies and research are moving more towards real-time and abstractive summarization.

III. DATASET

Our dataset consists of news articles issued by BBC News. This dataset for extractive text summarization has four hundred and seventeen political news articles of BBC from the year 2004 to 2005 [3]. This dataset was created using a dataset used for data categorization that consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005 used in the paper of D. Greene and P. Cunningham. ICML 2006; whose all rights, including copyright, in the content of the original articles are owned by the BBC [4]. For each article, five summaries are provided and the first clause of the text of articles is the respective title.

IV. THEORITICAL AND CONCEPTUAL STUDY OF TECHNIQUE

Topic modeling is an unsupervised method of discovering hidden topics represented in text or documents. Topic modeling is very useful in implementing Search Engine Optimization (SEO) techniques which can be achieved by incorporating document clustering of similar topics.

Latent Semantic Analysis (LSA) is one of the prominent fundamental techniques used in Topic modeling. The concept of LSA is very similar to that of cosine similarity. LSA is used in a wide range of applications, including text classification, dimension reduction, and text summarization. Text summarizing is the process of condensing long texts into coherent, fluent summaries by extracting only the most significant topics present in the text. According to International Data Corporation (IDC), the amount of digital data that is transmitted globally each year will rise from 4.4 zettabytes in 2013 to 180 zettabytes in 2025. Due to this exponential increase in data, traditional methods of data storage and access have become very challenging and ineffective. Text summarization helps in reducing the reading time, enables faster information search, and increases the amount of information that can fit in a storage by eliminating redundant information.

The primary goal of this paper is to provide an insight of how LSA can be used to summarize the huge amount of text present in texts or documents. The LSA algorithm's fundamental premise is that words with similar semantics will appear in paragraphs of text that are similar to one another. These techniques have been used by us to summarize news articles and provide insights to end users. This saves a great amount of time while conveying the same information and assisting in understanding news about different topics.

V. EXPERIMENTAL STUDY

Fig. 1 illustrates the model for the automatic text summarization. It consists of three stages: preprocessing of text, Latent Semantic Analysis (LSA) processing and Sentence Extraction. User will input an unstructured source text to the preprocessing unit.

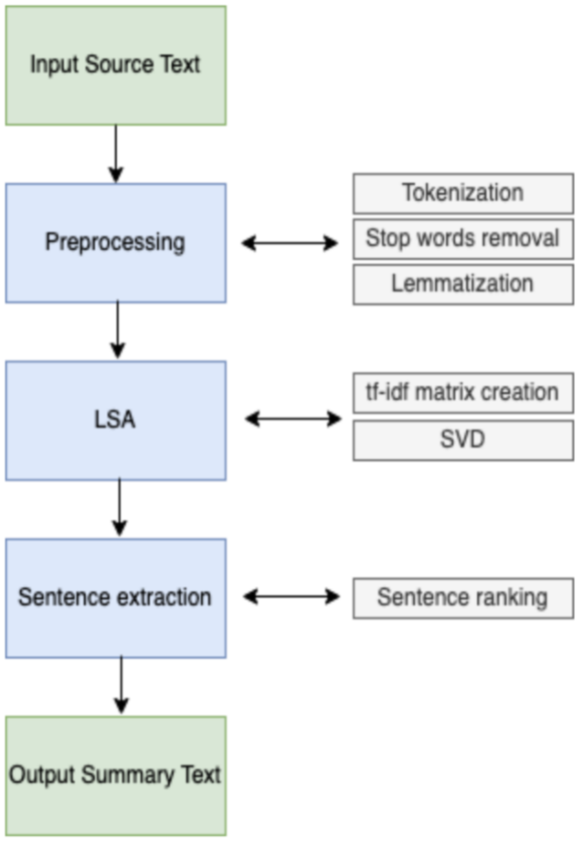

Figure 1: Automatic text summarization using LSA flow chart

1. *Preprocessing*

The input text document goes through multiple stages including tokenization, stop words removal and lemmatization in the preprocessing component.

Tokenization is the process of breaking down a phrase, sentence, paragraph, or even an entire text document into simpler components, like individual words or phrases. These simpler components are called tokens. The tokens may take the form of words, numbers, or punctuation. By identifying word boundaries, tokens are produced by breaking larger chunks of text into smaller ones. Word boundaries are the places where one-word ends, and the next word begins. These tokens are used in the subsequent steps to remove the stop words.

One of the preprocessing steps that is most frequently utilized across many NLP applications is stop word removal. The idea is to exclude words that appear frequently throughout all of the corpus's documents. Pronouns and articles are typically categorized as stop words. These words are not significantly discriminative because they have little relevance in some NLP tasks like information retrieval and classification. The low-level information in our text is eliminated by deleting these stop words, which helps the vital information stand out more. In other words, we may say that the model we train for our task does not exhibit any negative effects as a result of the removal of such phrases. As there are fewer tokens involved in training, the removal of stop words obviously reduces the size of the dataset and, consequently, the training time.

Lemmatization is one of the text pre-processing methods most frequently used in machine learning and Natural Language Processing (NLP). The objective of lemmatization is to identify the lemma form for each input word. The input string "I ate an apple," for instance, will be lemmatized into "I eat an apple." There are numerous real-world uses for this kind of word normalization. The aim is to remove inflectional suffixes and prefixes to bring out the word's standardized form.

2. *LSA*

Once the preprocessing stage is completed LSA algorithm is applied on the preprocessed text. The hidden semantic structures of phrases and sentences are extracted using the algebraic-statistical method known as LSA. The LSA seeks to ascertain how a document's concepts and phrases relate to one another. LSA consists of two steps: tf-idf matrix creation and Singular Value Decomposition (SVD)[10].

No.	**Sentences from input text**
S1	A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery.
S2	French musician Jean-Michel Jarre is to perform at a concert in Copenhagen.
S3	They each received an athletic scholarship to the school
S4	The ability of science and technology to improve human life is known to us.
S5	Rapid population growth is a serious problem in our country
…	…
Sn	Google and Meta funding for Canadian Publishers.

Table 1: Sentences from input text

TF-IDF is the short form for term frequency-inverse document frequency. It is a metric that quantifies the significance of a word in a corpus or collection of documents. [1] In information retrieval, text mining, and user modeling searches, tf-idf is frequently employed as a weighting factor. Overall, given the fact that some words are used more often than others, the value of tf-idf increases with the number of occurrences of the word in the document and is replaced by the number of documents in the corpus containing the term NumPy is used to multiply the TF matrix by the IDF vector to get the TF-IDF matrix. Here the words represent the rows of the matrix and sentences represent the columns of the matrix

$$A = \begin{array}{r c} & \begin{array}{ccccc} S_1 & S_2 & S_3 & \cdots & S_{25} \end{array} \\ \begin{array}{r} \text{Christmas} \\ \text{French} \\ \text{Athletic} \\ \text{Championship} \\ \text{Coast} \\ \text{Technology} \\ \cdots \\ \text{Population} \end{array} & \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \end{array}$$

Figure 2: TF-IDF matrix

Singular Value Decomposition (SVD) is based on a theorem from linear algebra in which the values in input matrix 'M' of size m x n are decomposed into three matrices. The values in input matrix 'M' represent the weight of each word in each phrase. As the complexity of SVD is strongly correlated with its size, the input text is preprocessed to decrease the dimensionality of matrix 'M'. We employ the frequency of words occurring in sentences out of all the numerous methods for filling the cell values of matrix 'M' [2], including binary representation, Tf-idf, log entropy, root type, and modified Tf-idf. The majority of words only appear in a few phrases at most, making 'M' a sparse matrix with an extremely low percentage of non-zero values. SVD breaks down 'M' into three new matrices, 'U' and 'V' as shown in the formula [9].

$$M = U \Sigma V^T \qquad (1)$$

$\boldsymbol{U}$ is a $\boldsymbol{m \times m}$ complex unitary matrix

Σ *is a* $\boldsymbol{m \times n}$ rectangular diagonal matrix with non-negative diagonal numbers on the diagonal

$\mathbf{V}$ is a $\boldsymbol{n \times n}$ complex unitary matrix. V^T is its transpose

Here the terms and sentences can be viewed and represented in a semantic space, which can be illustrated as a scatter plot. The terms or tokens which are frequently occurred in many sentences, or the document as whole tend to form a cluster, representing important concepts. The difficult part is figuring out how many concepts should be chosen in the document based on the interrelationship of words and sentences in the LSA model. The idea is that concept clustering will make it easier to condense the key concepts from documents.

3. *Sentence Extraction*

The final step is to extract the sentences and to generate the summary. The results from LSA model are used to pick the top sentences to generate the summary text. Every sentence will be assigned a measure called Length. Length is the sum of products of corresponding singular vector diagonal values and right singular vector elements [7].

$$\mathbf{Length_i} = \mathbf{sqrt}\left(\sum_{i=1}^{n} (\mathbf{v_{ij}} * \mathbf{S_{jj}})\right)$$

The sentences are ranked based on the length values in descending order. For our experimental study the total number of sentences are divided by a constant value 5 and the result is stored in a variable ‘k’. The top ‘k’ sentences with the highest sentence scores were chosen to form a summary.

VI. RESULTS

For evaluation we use the Rouge evaluation approach. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for assessing machine translations and automatic text summarization. We evaluate by comparing the summary produced by the model and a reference summary (typically human produced).

For example, if the model produced summary is “The dog is sleeping under the table” and the human produced summary is “The dog is sound asleep beneath the desk”. There are four words that overlap between the system summary and reference summary if we consider only the individual words are considered. This metric wouldn’t be much of a use and would not convey important quantitative information. To get a good. We can calculate the precision and recall utilizing the overlap to obtain a good quantitative metrics. Recall refers to the measure of how much similar the model produced summary and the reference summary is.

$$Recall = \frac{Number\ of\ overlapping\ words}{Total\ number\ of\ words\ in\ reference\ summary}$$

Recall will just tell us the measure of same words between the two summaries. However, recall includes stop words, and the system summary may include many such stop words making the summary verbose.

We use precision to overcome the drawback of recall. Precision gives the measure of how much of the system produced summary is relevant or useful.

$$Precision = \frac{Number\ of\ overlapping\ words}{Total\ number\ of\ words\ in\ system\ summary}$$

When trying to generate brief summaries, the precision factor becomes vital. Hence, it is always preferable to compute the precision and recall first and then calculate the F-Measure.

The F-score or F-measure is a test accuracy metric. It is calculated using the test's precision and recall, where precision is the number of true positive results divided by the total number of positive results, including those that were incorrectly identified, and recall is the number of true positive results divided by the total number of samples that should have been identified as positive. [6]. We can say that F-score is a model's accuracy on a dataset.

$$F-score = 2\,\frac{Precision * Recall}{Precision + Recall}$$

We have used three rouge metrics: Rouge 1, Rouge 2, Rouge L. Rouge 1 considers only unigrams as overlapping words to calculate precision, recall, and F-score. Rouge 2 considers bigrams as overlapping words to calculate precision, recall, and F-score. Rouge L measures the precision, recall and F-score considering the longest common subsequence as overlapping words [8].

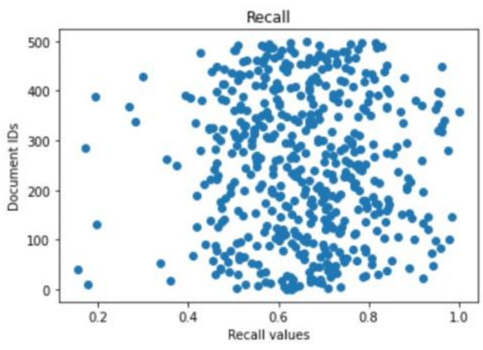

Figure 3: Scatter plot of recall values for 500 documents

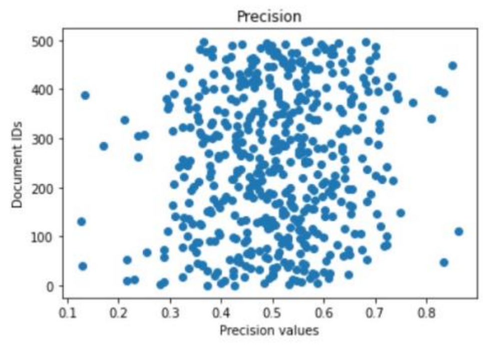

Figure 4: Scatter plot of precision values for 500 documents

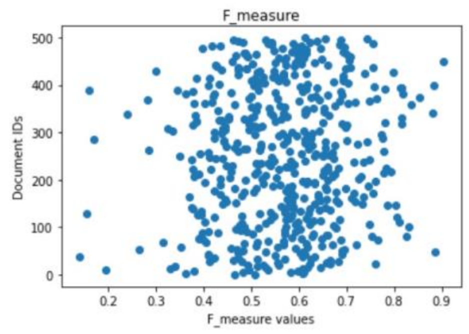

Figure 3: Scatter plot of F-measure values for 500 documents

No. of Documents (N)	Reduction Ratio	Ranking Ratio	Recall (avg)	Recall (max)	Precision (avg)	Precision (max)	F_measure (avg)
2225	1/1	n/3	0.687385773904 9624	0.987385773904 9624	0.526193487654245 98576	0.85	0.593458912542398
50	1/1	n/3	0.4836525436014816	0.9636363636363636	0.43950721756010624	0.76	0.4580338998476718
500	1/1	n/3	0.5323922919085837	0.9636363636363636	0.476613020524510 66	0.8173076923076923	0.4995411457493 1274
50	1/2	n/3	0.4798308635202222	0.9636363636363636	0.43103115058403924	0.76	0.4515084896095 2084
500	1/2	n/3	0.533234013746757	0.9636363636363636	0.47446230611318724	0.8173076923076923	0.4985470316187467
50	1/1	n/4	0.4826306434859164	0.9523809523809523	0.3512405153161629	0.74	0.4029609974729818
500	1/1	n/4	0.5576884372356341	1.0	0.397853142990424 46	0.7752808988764045	0.4598508654961 6633

50	1/2	n/5	0.47786109108652	1.0	0.287141417548447	0.74	0.3530240816538 3736
500	1/2	n/5	0.5714248938228866	1.0	0.345760867375858 87	0.7752808988764045	0.4256598844065 0526

Table 2: log report of the Automatic text summarization

VII. CONCLUSION

In conclusion, to implement the automatic text summarization using extractive methodology on the BBC news articles. First, the input text was preprocessed by removing the stop words and lemmatized. The output from the preprocessing stage is then passed to the LSA algorithm, where SVD components are calculated. Based on the SVD values, we rank each sentence by calculating its length. Finally, we select the top ranked sentences to create the summary. For evaluation of our model, we used Rouge methodology, where we used Rouge 1, Rouge 2 and Rouge L as our evaluation metrics.

REFERENCES

[1] https://en.wikipedia.org/wiki/Automatic_summarization

[2] Luhn, Hans Peter (1957). "A Statistical Approach to Mechanized Encoding and Searching of Literary Information" (PDF). IBM Journal of Research and Development. 1 (4): 309–317. doi:10.1147/rd.14.0309. [3]—do not use "Ref.

[3] https://www.kaggle.com/datasets/pariza/bbc-news-summary

[4] http://mlg.ucd.ie/datasets/bbc.html

[5] M. G. Ozsoy, I. Cicekli and F. N. Alpaslan:Text Summarization of Turkish Texts using Latent Semantic Analysis. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 869–876, Beijing, August 2010

[6] https://en.wikipedia.org/wiki/F-score

[7] http://parishodhpu.com/gallery/30-november-2203.pdf

[8] https://www.freecodecamp.org/news/what-is-rouge-and-how-it-works-for-evaluation-of-summaries-e059fb8ac840/

[9] O. -M. Foong, S. -P. Yong and F. -A. Jaid, "Text Summarization Using Latent Semantic Analysis Model in Mobile Android Platform," 2015 9th Asia Modelling Symposium (AMS), 2015, pp. 35-39, doi: 10.1109/AMS.2015.15.

[10] K. Merchant and Y. Pande, "NLP Based Latent Semantic Analysis for Legal Text Summarization," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1803-1807, doi: 10.1109/ICACCI.2018.8554831.