Extractive Text Summarization Using Text Rank And Tfidf-Vectorizer

Srushtiraj Patil
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India

Dhiraj mukane
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India

Rohant Narang
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India

Pavitra mandot
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India

Triveni fole
Department of Information Technology
Vishwakarma Institute of Technology
Pune, India

***Abstract*— meeting or text summarizing is the process of condensing valuable information from a meeting into a shorter, more digestible format. This can be a challenging task when dealing with thousands of words of information. There are two main approaches to summarization: abstractive and extractive. In this paper, we will break down the problem of meeting summarization into extractive and extractive components to produce a summarized paragraph. Our proposed solution involves converting recorded meetings, seminars, interviews, presentations, and conferences from audio to text format. We then apply natural language processing (NLP) models to the text to generate a summary that captures the key points of the meeting. Our approach is based on NLP and has an accuracy of 82 in compared to original descriptions.**

***Keywords*— natural language processing, Text summarization, Extractive summarization, Extractive summarization,**

I. Introduction

Summarization is the process of condensing information into a shorter, more meaningful format. In the age of the internet, there is an abundance of information available in various formats. Not everyone needs all the information from a single source, and often information is pieced together from multiple sources. This is where text summarization comes in handy. Text summarization involves converting information from various formats into text and then summarizing it using NLP techniques to produce a compact and impactful summary with fewer words.

One useful application of text summarization is to summarize long meetings to save time. To summarize a meeting, we can use NLP techniques to analyze the content and extract the key points. There are various techniques for summarizing text, including extractive, abstractive, multi- or single-document, generic, and query-based summarization. Extractive summarization involves generating a summary by selecting important sentences or phrases from the original text. Abstractive summarization, on the other hand, involves constructing a new summary by interpreting the text and generating new sentences. In this paper, we will focus on extractive and abstractive summarization techniques for meeting summarization.

- Abstractive Summarization

Abstractive summarization is an advanced natural language processing technique used for constructing a new summary by interpreting the text of a document. This technique generates a summary by covering the most important data points, even if they were not presented in the original document. Extractive text summarization, on the other hand, generates a summary by selecting a subset of sentences from the original document that covers the important topics. Single-document summarization generates a text summary from a single document, while multi-document summarization generates summaries from multiple documents.

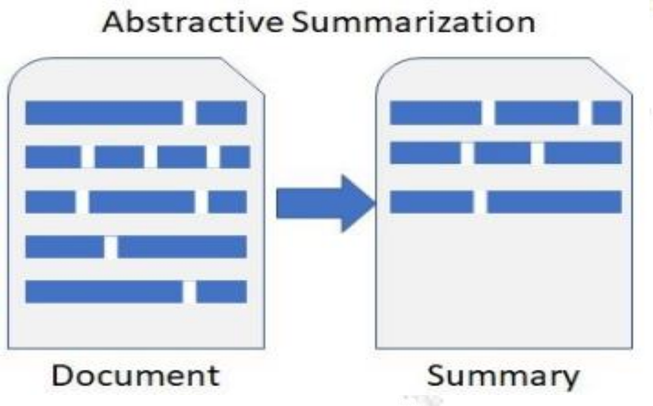


Fig 1. Abstractive summerization Concept

- Extractive Summarization

Extractive summarization is often used because it is easy to implement. It involves a binary classification problem on the input text, and its objective is to evaluate accuracy. The primary objective of extractive summarization is to identify the important information and present it in a comprehensive overview. This form of summarization is commonly seen in text summarization systems.

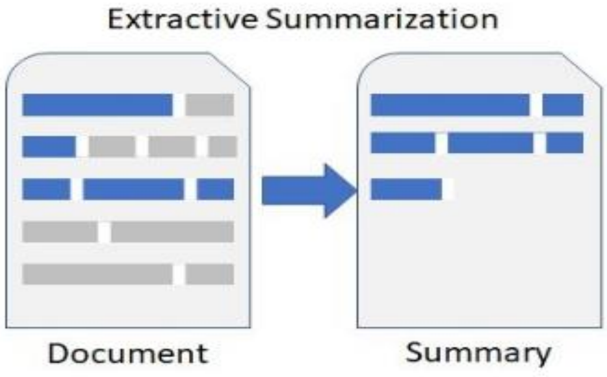


Fig 2. Extractive Summarization Concept

Automatic text summarization is done in three phases: data pre-processing phase, algorithmic processing phase, and post-processing phase. Most current automatic text summarization systems use extraction to generate summaries. Sentence extraction techniques are commonly used to generate extracted summaries. One of the methods of obtaining suitable sentences is to assign a numerical measure of a sentence to the summary, called the sentence score, and then select the best sentences to form a summary of the material. Data based on compression ratio is another important factor used in extraction methods to determine the ratio between the length of the summary and the source text. When the compression rate is between 5-30%, the quality of summary is considered acceptable.[7]

A. *Data Pre-processing*

Before summarizing a document, it's essential to pre-process it by cleaning and converting it into a more functional data format. This process includes the following steps:

1. Removal of noise data found in the document.
2. Word tokenization and sentence segmentation
3. Removal of punctuation marks
4. Removal of stop words such as "and" "an," "or" etc.
5. Removal of suffixes and prefixes
6. Word lemmatization transforms words to their base structure.

B. *Algorithmic Processing*

TextRank Algorithm

TextRank One of the most used algorithms for text summarization is the TextRank algorithm. The algorithm uses a graphical text processing ranking model to determine the most important sentences in a text. Here's how it works:

I. TextRank rates the importance of each sentence and then sorts them accordingly. The first sentence shown is considered the main idea of the text and can be understood as a summary. In the basic NLP method, all sentences in the input are vectorized, ranked, and the top three sentences are selected and returned.[1]

II. The next step involves converting sentences and all other words into vectors. The system finds the similarity coefficient for each sentence based on the words used and places the coefficients in a similarity matrix for each sentence. These similarities are placed in a network graph and sorted using the PageRank algorithm. The sentence closest to 1 is the most similar sentence, and it is selected as the main idea.

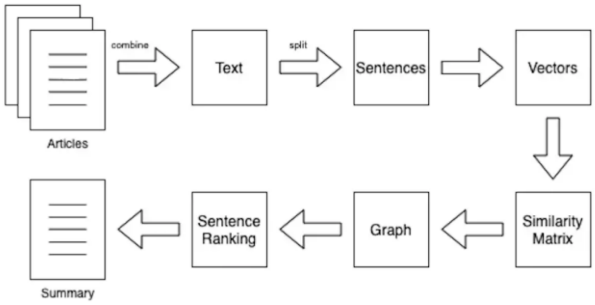
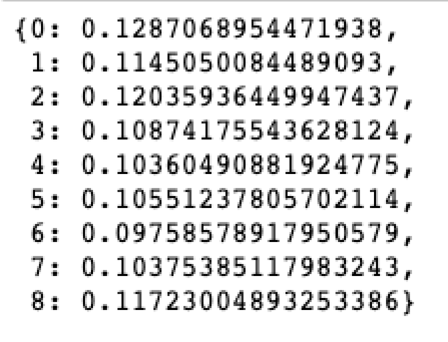


Fig 3. Work Principle of TextRank

Post-processing is another method of changing data to generate the target summary. This phase is optional in some models. One popular post-processing algorithm is the Luhn Summary algorithm, which is based on TF-IDF (Term Frequency Inverse Document Frequency) [9]. It is useful when rare words and stop words have no meaning. Sentence scoring is done based on TF-IDF, and the highest-ranked sentences appear in the summary.



```
{0: 0.1287068954471938,
 1: 0.1145050084489093,
 2: 0.12035936449947437,
 3: 0.10874175543628124,
 4: 0.10360490881924775,
 5: 0.10551237805702114,
 6: 0.09758578917950579,
 7: 0.10375385117983243,
 8: 0.11723004893253386}
```

Fig 4. similarity of sentences

In 2nd After the preprocessing, the Luhn Summary algorithm approach is used for post-processing to generate the summary. This approach is based on TF-IDF (Term Frequency Inverse Document Frequency), which is useful when rare and frequent words (stop words) have no meaning. Sentence scoring is done based on this approach, and the highest ranked sentences will appear in the summary.

In the Luhn Summary algorithm approach, the frequency of each word in the text is calculated, and the frequency of each word is then divided by the total number of words to obtain the frequency ratio of each word. The words with the highest frequency ratios are considered the most important words in the text. Stop words, which are commonly occurring words such as "and" and "the," are removed from the text to obtain more meaningful words.

After obtaining the most important words, the Luhn Summary algorithm approach calculates the relevance of each sentence to the text based on the frequency of important words in each sentence. The sentences with the highest relevance scores are selected for the summary.

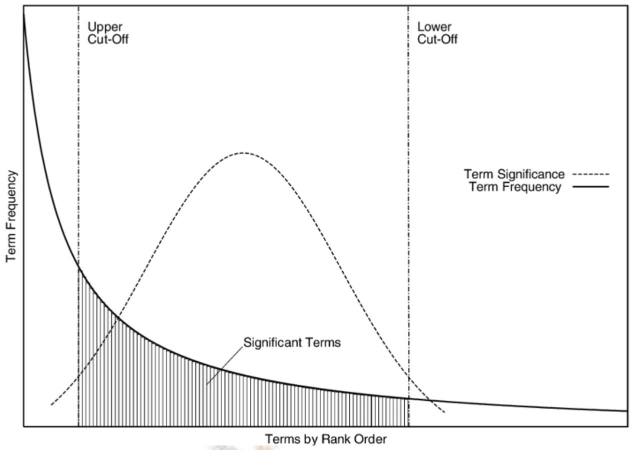


Fig 5. The area on the right means the most frequent items while the words on the left mean the less frequent items.

After obtaining the most important words, the Luhn Summary algorithm approach calculates the relevance of each sentence to the text based on the frequency of important words in each sentence. The sentences with the highest relevance scores are selected for the summary.

In summary, the Luhn Summary algorithm approach uses the TF-IDF approach for sentence scoring and relevance calculation. This approach is useful for generating summaries that contain the most important information from the original text.

II. Litreture Review

[1] The paper builds on previous research on text summarization, which is the task of producing a shorter version of one or more texts that preserves the essential information and meaning. Text summarization can be classified into extractive and abstractive methods. Extractive methods select important sentences or phrases from the original texts and concatenate them to form a summary. Abstractive methods generate new sentences that paraphrase or restate the original texts. The paper reviews some of the existing techniques and applications of text summarization, such as sentence extraction, recurrent neural networks, supervised learning, and unsupervised learning. The paper also discusses some of the challenges and limitations of text summarization, such as handling noisy data, preserving coherence and relevance, and dealing with domain-specific vocabulary and knowledge.

The paper “Applications and future of machine reading comprehension” [2]by Zhu provides an overview of the machine reading comprehension (MRC) task, which aims to enable machines to read, analyze and summarize text. The paper introduces the definition, challenges and applications of MRC, as well as the main approaches and models for solving MRC problems. The paper also discusses some of the current trends and future directions of MRC research, such as multimodal MRC, commonsense reasoning, explainable MRC and lifelong learning.

[3] The paper “Big Data Driven Natural Language Processing Research and Applications” by Gudivada, Rao and Raghavan explores the opportunities and challenges of applying natural language processing (NLP) techniques to big data analytics. The paper reviews some of the fundamental concepts and methods of NLP, such as tokenization, word vectors, linguistic tagging, language models and text summarization. The paper also presents some of the emerging applications and domains of NLP in big data scenarios, such as social media analytics, sentiment analysis, information extraction and question answering. The paper also identifies some of the open issues and research directions for advancing NLP in big data environments, such as scalability, parallelization, heterogeneity and quality.

[4] The paper “Performance Study on Extractive Text Summarization Using BERT Models” by Abdel-Salam and Rafea investigates the performance of different variants of BERT-based models for the extractive summarization task, which aims to select the most important sentences from a document to form a summary. The paper fine-tunes and evaluates three models: BERTSum, which uses the original BERT encoder; DistilBERTSum, which uses a distilled version of BERT with fewer parameters; and SqueezeBERTSum, which uses a compressed version of BERT with fewer operations. The paper compares the models on two datasets: CNN/Daily Mail and XSum, using ROUGE scores as the evaluation metric. The paper also analyzes the trade-off between model size, speed and performance. The paper finds that SqueezeBERTSum achieves competitive results with BERTSum while being significantly smaller and faster. The paper also suggests some future directions for improving extractive summarization using BERT-based models, such as incorporating domain knowledge, using pre-trained models on summarization data, and exploring other variants of BERT.

In the paper [5] N. Moratanch and S. Chitrakala, "A Survey on Extractive Text Summarization" The paper presents two types of level feature which are word level feature and sentence level feature. The paper also mentions the categorization of all extractive summarization methods. The paper distributes the weight between the supervised and unsupervised methods where each and every method is explained in detail and evaluated at the end with evaluation matrix.

The paper [6] describes a tool called Smart Meeting that can help people manage their meeting content more efficiently. The tool can automatically record, transcribe, summarize, and organize the content of meetings. It also has a feature that can recognize the attendees of in-person meetings. The tool has three main functions: transcription by ASR, transcript enrichment, and meeting summarization. The tool uses a hybrid ASR pipeline to transcribe the meeting audio. The transcript enrichment process consists of four steps: 1) voiceprint-based speaker diarization for each transcript, 2) speaker labeling for each separated utterance, 3) quality evaluation for each utterance, and 4) context selection and merging. These steps produce a refined transcript with speaker information and quality scores. For speaker diarization and identification, the tool employs a CNN with self-multi-head attention to segment and cluster the utterances based on speaker voiceprints. For meeting summarization, the tool uses WSNeuSummary, a supervised pre-training mechanism that can handle data scarcity. WSNeuSummary has two steps: pre-training with weak supervision and fine-tuning with limited labeled instances.

The tool adopts a transformer-based network that incorporates BERT for transcription purposes.

By combining key information from various utterances, authors [7] suggest a method for creating extractive summaries. The significant utterances within every segment are then identified using a neural network classifier after the conversation transcripts have first been divided into several subject parts. The crucial phrases are then placed together to create a short overview. The interdependence parses of the utterances in each segment are linked to construct a directed graph in the text generation step. To provide a one-sentence summary to every topic segment, the most insightful and is very well sub-graph from the integer linear programming (ILP) results is considered. Three steps make up the suggested method: Start by breaking up a lengthy text exchange into smaller text segment. Apply an extractive summarizer second, which pulls out key phrases from each segment. Finally, create a summary sentence by fusing all the utterances in a segment using an ILP-based technique. The final summary is created by appending each of the generated sentences. LCSeg and Bayesian unsupervised topic segmentation are 2 distinct text segmentation methods that the researchers examine. Significant utterances are chosen using the synthetic minority oversampling technique (SMOTE). The final stage involves linearizing the integer linear programming (ILP) problem's answer to create a phrase. The AMI Meeting corpus includes 139 meeting transcripts and the extractive and extractive summaries that would go with these. The suggested method may produce pertinent extractive summaries from meeting transcripts without any templates, according to trials on topic selection and readability.

The paper [8] compares different methods for sentiment analysis and text summarization. Sentiment analysis is a technique that uses machine learning to extract the feelings and emotions expressed in text. Some of the machine learning techniques used are Naive Bayes Classifier and Support Vector Machines (SVM). These techniques can identify the sentiments and emotions in textual data such as reviews of movies or products. Text summarization is a technique that uses natural language processing (NLP) and linguistic properties of sentences to select the most important words and sentences for the final summary.

This publication [9] offers an abstract perspective of the current research work scenario for text summarising. This study discusses the specifics of both the extractive and extractive approaches, the methodology employed, the results obtained, and the benefits and drawbacks of each strategy. The scholarly community and the commercial sector both value text summary. Compared to extractive summarising, extractive summarization produces more meaningful and suitable summaries, but it is a little more complex because it takes more learning and reasoning. Through the study, it was also discovered that relatively little work on Indian languages has used abstract methods; hence, there is much need for further investigation of these techniques for more accurate summary.

The paper [10] reviews the research done from 1958, of automatic text summarization. The paper shows the distribution of different techniques used during each year. The techniques such as extractive, extractive, domain, real time, optimization, multi document, single document. They also focused on the problems related to text summarization which are semantic, extraction , similarity, redundancy, ambiguity, scoring , optimization, word frequency, semantic analysis, noise, clustering, keyword, sentence ranking, feature, sentence scoring. The models used in text summarization in Fuzzy, apparitor, PSO, GA, TF- IDF, NLP, MArkv ,SVM , K -Means, ABC, TF, AE, Co-Rank, LSA, sentence scoring, deep learning. Evaluation of text summarization is done based on rough ,precision, recall, f-measure, BELU , METEOR, CR, and copy rate. SLR methods have been shown to provide a more structured, broader and more diverse overview, from trends / themes, datasets, preprocessing, characteristics, approach methods, problems, methods to evaluations. increase. Future work guides, trends / thematic relationships, issues and challenges for each theme, techniques and methods used are combined into one to facilitate research and reanalysis.

The major goal of the suggested strategy [11] is to automatically provide timestamps and descriptions for movies. Frames, feelings, and words all have a role in how the video is summarised. First, a summary of the video content is output together with the video clip itself, which displays in the frame. Second, the outputted summarization of the frames was combined with emotion and how it varies over time. Third, an abstract summary of the audio track is produced by the audio transcription into text. Finally, using natural language processing techniques, all summarizations (audio, video, and emotion) are combined. Tokenization, sentence segmentation, lemmatization & stemming, followed by extractive summarization are some of the techniques used. The experiment's results showed that, on average, 87 percent of participants thought the generated text did a good job of describing the movie.

The many approaches, tactics, and methodologies used in automatic text summarization are thoroughly reviewed in this study [12]. An autonomous summarization system's major goal is to generate a summary with the lowest number of duplication and related details in the shortest period of time. Future studies will go into greater detail on the most recent computing approaches available for extractive summarization jobs involving one or more documents. Extractive and extractive summarizers were applied in this case. The population, consumption, evolutionary measurements, and summary generator can all be used to categories the system's output.

SummCoder [13], an unsupervised framework for extracting text summarization neural network - based auto-encoders, was introduced by Joshi, Eduardo, Enrique, and Laura in 2019. According to their method, an extractive summarization problem was a problem of phrase selection from a document. The following three metrics were used to choose which sentences should be in the summary:

1. Sentence Content Relevance Metric.

2. Sentence Novelty Metric.

3. Sentence Position Relevance Metric.

This framework's ranking of the sentences using the three aforementioned criteria states the issue as follows: The sentence I is embedded into a vector VSi, which is the encoder representation calculated with the three metrics discussed earlier, and then decoded given a document D with N sentences D = (S1, S2..., SN).

The paper [14] introduces EdgeSumm, a graph-based framework that combines four extractive algorithms for text summarization. EdgeSumm uses graph-based, statistical-based, semantic-based, and centrality-based methods to select the most important sentences from the input document. EdgeSumm claims to achieve better performance than state-of-the-art systems in ROUGE-1 and ROUGE-2 metrics. Their proposed framework aims to provide a general solution for summarizing information from different domains. It first creates a text graph model based on the output of the pre-processing step. Then, it assigns weights to each node in the graph, which are based on the word frequency and other factors such as the word’s presence in the title

In The paper [15] presents MatchSum, a novel framework that treats extractive text summarization as a text matching problem. MatchSum uses a Siamese-BERT model to match a source document and candidate summaries (extracted from the original text) in a semantic space. MatchSum selects the candidate summary that is closest to the reference summary in that space as the output summary. The authors argue that this matching-based framework has not been fully explored yet. They conduct experiments on five datasets that demonstrate the effectiveness of the matching framework..

III. Proposed Methodology

The extractive summarization technique involves several steps to create a summary from the input text. The first step is to construct an intermediate representation of the input text. This representation highlights the important information of the text and serves as the basis for the evaluation of the sentences. There are different methods to create the intermediate representation, including frequency-based and topic-based approaches.

The frequency-based approach assigns weights to words based on their relevance to the topic. Words that are related to the topic are given a weight of 1, while words that are not relevant are given a weight of 0. The weights can also be continuous, depending on the implementation. One way to calculate the importance of each word is to use the word probability method. This method uses the frequency of the word in the input text to determine its importance. The probability of a word is given by its event frequency, f(w), divided by the total number of words in the input text, N.

After the weights have been assigned to each word, the next step is to evaluate the sentences based on the weights. The sentences that contain the most important words are considered more important than the sentences that contain less important words. One way to determine the importance of each sentence is to calculate the sentence score based on the sum of the weights of the words that appear in the sentence. The sentences with the highest scores are selected for the summary.

Another method for creating the intermediate representation is the topic-based approach. This approach focuses on expressing the subject of the text. There are different methods to obtain the topic representation, including latent semantic analysis and Bayesian models such as latent Dirichlet allocation (LDA).

The TF-IDF (Term Frequency Inverse Document Frequency) technique is an advancement of the word probability method. It works by assigning weights to the words based on their frequency in the document and the entire corpus. The weight of a word is high if it appears frequently in the document but infrequently in the corpus. Conversely, the weight of a word is low if it appears frequently in the corpus. The TF-IDF technique can also be used to assign low weight to words that are stop words or occur frequently in most documents.

In summary, the extractive summarization technique involves constructing an intermediate representation of the input text, evaluating the sentences based on the intermediate representation, and selecting the most important sentences for the summary. There are different methods to create the intermediate representation, including frequency-based and topic-based approaches. The TF-IDF technique is an advancement of the word probability method and can be used to assign weights to the words.

Text Summarization1 (Text rank)

A. Collect data.
B. Clean up data.
C. Algorithms to build word or
sentences)
D. Word frequency
E. Weighted frequency for each word
F. Calculate score for each sentence.
G. Select top sentences for summary.

You can modify the default setting of the sorting function according to your needs.

$$P(w) = \frac{f(w)}{N}$$

Word probability calculation

The parameters are:

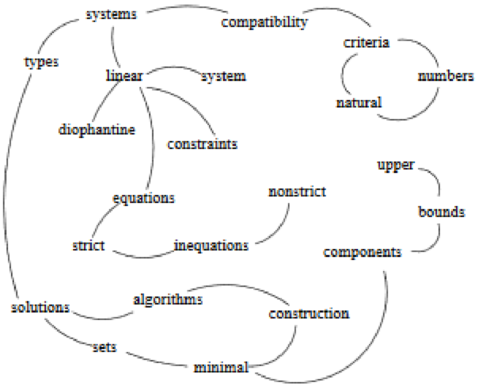
Ratio: It can take a value from 0 to 1. It represents the ratio of the summary to the original text.
wordcounts: It determines the number of words in the summary.

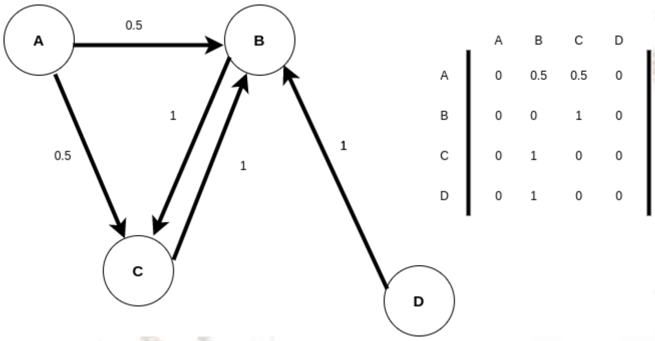
The following steps are taken in the Text Rank summary:
1. The document is preprocessed and segmented into sentences.
2. All sentences are still denoted by words and stop words are omitted.
3. The frequency of each word is calculated and normalized.
4. Each sentence is assigned a score by adding the normalized frequency values of its constituent words.
5. High-scoring sentences make up the summary.

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

Fig 6. Working of a page rank algorithm

W represents the weight factor. The text classification implementation consists of two different natural language processes:
Keyword extraction task, keyword, and phrase selection
Phrase extraction task, identifying sentences most important.

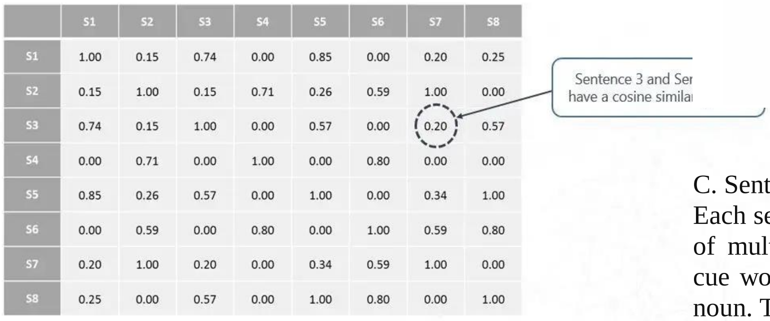




	A	B	C	D
A	0	0.5	0.5	0
B	0	0	1	0
C	0	1	0	0
D	0	1	0	0

Fig 7. Working of tof text algorithm

The text is represented by natural language and the parts of speech are tagged, and single words are added to the word graph as nodes. Then, if two words are identical, the corresponding nodes are connected by an edge. Co-occurrence words are used to measure similarity. If two words appear in a window of N words, where N ranges from 2 to 10, the two words are considered identical. The words with the highest number of significant incoming edges are selected as the most important keywords.



	S1	S2	S3	S4	S5	S6	S7	S8
S1	1.00	0.15	0.74	0.00	0.85	0.00	0.20	0.25
S2	0.15	1.00	0.15	0.71	0.26	0.59	1.00	0.00
S3	0.74	0.15	1.00	0.00	0.57	0.00	0.20	0.57
S4	0.00	0.71	0.00	1.00	0.00	0.80	0.00	0.00
S5	0.85	0.26	0.57	0.00	1.00	0.00	0.34	1.00
S6	0.00	0.59	0.00	0.80	0.00	1.00	0.59	0.80
S7	0.20	1.00	0.20	0.00	0.34	0.59	1.00	0.00
S8	0.25	0.00	0.57	0.00	1.00	0.80	0.00	1.00

Sentence 3 and Ser
have a cosine similar

Fig 8. The table above shows the cosine similarity matrix that is used to create a graph for the TextRank ranking algorithm.

Text Summarization2 (TIDF)

Text Summarization2 (TIDF) Text summarization approach consists of following stages:

A. Preprocessing

B. Feature Extraction

C. Sentence Scoring

A. Text Preprocessing There are four steps in preprocessing:

1. Segmentation: Document are split into sentences.

2. Removal of Stop words: Stop words are common words such as ‘a’ an’, the’ that provide less meaning and contain noise. The Stop words are stored in an array and those are predefined.

3. Word Stemming: converts every word into its root form by removing its prefix and suffix so that it can be used for comparison with other words.

B. Feature Extraction:

The text document is represented by set, D= {S1, S2, - - -, Sk}

TF-IDF score for a word can be calculated as

$$TF(w) = \frac{No.\ of\ times\ w\ appears\ in\ a\ document}{Total\ no.\ of\ terms\ in\ the\ document}$$

$$IDF(w) = log_e(\frac{Total\ no.\ of\ documents}{No.\ of\ documents\ with\ term\ w\ in\ it})$$

where, Si denotes a sentence contained in document D. The document belongs to feature extraction. The important word and sentence features to be used are decided. Title word, Sentence length, Sentence position, numerical data, Term weight, sentence similarity, existence of Thematic words and proper Nouns these features are used by feature extraction [15]

$$w_{i,j} = tf_{i,j} \times log_e(\frac{N}{df_i})$$

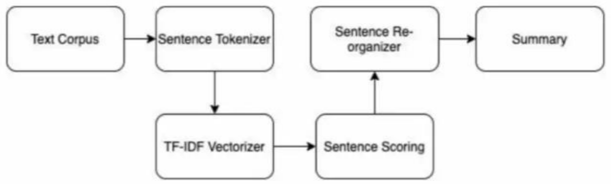


Fig 9. TF-IDF score along with its workflow.

C. Sentence Scoring:
Each sentence is scored by considering a linear combination of multiple features such as frequency, sentence position, cue words, similarity with title, sentence length and proper noun. The sentences are ranked according to their scores.

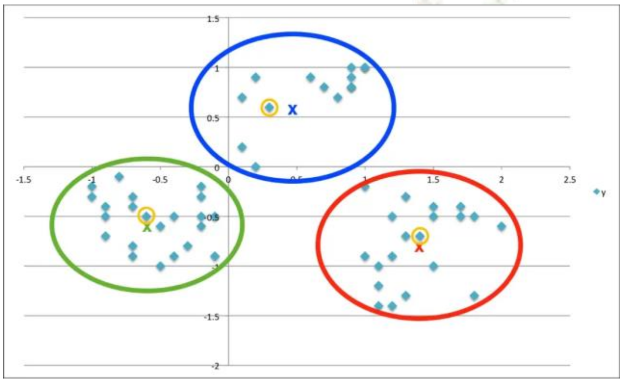


Fig 10. Each point represents a sentence in the vector space. The sentences circled in yellow represent the sentences that are closest to the cluster center and would be selected for the summary.

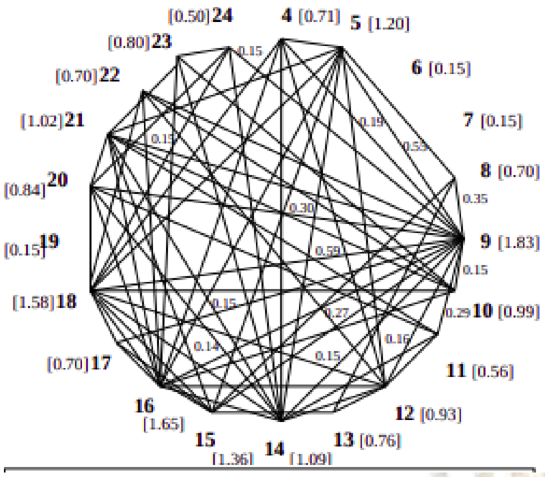


Fig 11. The similarity between two sentences is given by:

Where given two sentences Si and Sj, with a sentence being represented by the set of Ni words that appear in the sentence:

The most important sentences are obtained in the same way we did for Keyword extraction.

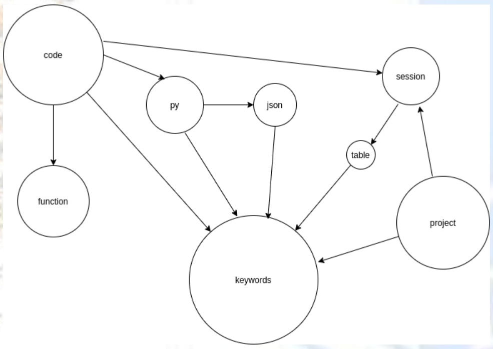
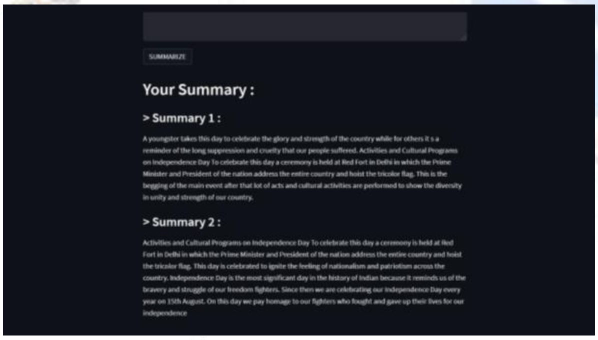


Fig 12. Keyword graph

IV. Result and Discussion



Your Summary :

> Summary 1 :

A youngster takes this day to celebrate the glory and strength of the country while for others it s a reminder of the long suppression and cruelty that our people suffered. Activities and Cultural Programs on Independence Day To celebrate this day a ceremony is held at Red Fort in Delhi in which the Prime Minister and President of the nation address the entire country and hoist the tricolor flag. This is the begging of the main event after that lot of acts and cultural activities are performed to show the diversity in unity and strength of our country.

> Summary 2 :

Activities and Cultural Programs on Independence Day To celebrate this day a ceremony is held at Red Fort in Delhi in which the Prime Minister and President of the nation address the entire country and hoist the tricolor flag. This day is celebrated to ignite the feeling of nationalism and patriotism across the country. Independence Day is the most significant day in the history of Indian because it reminds us of the bravery and struggle of our freedom fighters. Since then we are celebrating our Independence Day every year on 15th August. On this day we pay homage to our fighters who fought and gave up their lives for our independence

Fig 13. Result of summazriztion by website

Developing an efficient and accurate summarization system is an ongoing research challenge. One of the main difficulties is how to evaluate the quality of a summary. One of the popular methods for text summarization is TextRank, which does not require extensive linguistic knowledge or a domain- or language-specific annotated corpus, making it highly adaptable across domains,[4] genres or languages. Extractive methods are more commonly used than abstractive methods. Our algorithm shows better results than the output of the online summarizer. The rapid growth of technology and the use of the Internet have created an information overload. Text summarization can solve this problem by creating a useful summary of the document for the user. Therefore, it is necessary to develop a system that allows users to obtain concise summaries of documents.

A possible solution is to use either an extractive or an abstractive method. Extractive summaries are easier to create.

$$ROUGE-2=\frac{\sum_{S\in\{\text{RefSummaries}\}}\sum_{\text{bigrams } i\in S}\min(count(i,X),count(i,S))}{\sum_{S\in\{\text{RefSummaries}\}}\sum_{\text{bigrams } i\in S}count(i,S)}$$

Fig 14. Formula for calculating ROUGE-2 score.

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is a method of measuring the quality of a summary by comparing it with other human-made summaries as a reference. For model evaluation, there are several human-generated references and machine-generated candidate summaries. The intuition behind this is that if a model produces a good summary, it must have common parts that overlap with human references. It was proposed by Chin-Yew Lin, University of California.[3]

Summarization Approach	Dataset Type	News Dataset			
	ROUGE Type	ROUGE-L	ROUGE-SU4	ROUGE-1	ROUGE-2
TextRank	R_a (Avg_Recall)	0.803823	0.7707474	0.7902034	0.7603536
	P_a(Avg_Precision)	0.5686992	0.5494134	0.5607558	0.5405252
	F_a (Avg_F-Measure)	0.6520132	0.6232262	0.636767	0.6139468
TF IDF	R_a (Avg_Recall)	0.5352098	0.4290038	0.5030586	0.4150356
	P_a(Avg_Precision)	0.4541242	0.4151762	0.4675862	0.3959512
	F_a (Avg_F-Measure)	0.4868588	0.4161304	0.479677	0.4003722

Fig 15. calculating ROUGE-2 score

V. Conclusion

Extractive summarization remains a persistent challenging task for deep-learning Natural Language processing. This is especially true when the task is applied to a domain-specific corpus that differs from the pre-training, is extremely specialized, or encompasses a limited set of training data. The implemented system involves the BERT model and BART model which gives the summarization of text document. However, it is not human-level efficiency, the result is explainable and acceptable.

VI. Future Scope

- Remove the language barrier.

The system only takes input as English language and gives the summery in English text document. As the future implementation the system should take different language as input and given the summery in same language text document. For this the lexical data should be available online.

- Dive into the social application:

The system can take over in the social media to summarize the various things. The summarization can be done on comments, post or even twitter.

REFERENCES

[1] Jha, aryan, Temkar, S., &amp; Hegde, P. (2022). Business meeting summary generation using NLP - ITM-conferences.org. https://www.itm-conferences.org/. Retrieved 2022, from https://www.itmconferences.org/articles/itmconf/pdf/2022/04/itmconf_icacc2022_03063.pdf

[2] Zhu, C. (2021, March 26). Applications and future of machine reading comprehension. Machine Reading Comprehension. Retrieved June 19, 2022, from https://www.sciencedirect.com/science/article/pii/B9780323901185000084 V. N. Gudivada, D. Rao, and V. V. Raghavan, "Big Data Driven Natural Language Processing Research and Applications," Handbook of Statistics, 05-Aug-2015. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/B978044463492400009 55

[3] C. Zhu, "Applications and future of machine reading comprehension," Machine Reading Comprehension, 26-Mar-2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323901185000084.

[4] Abdel-Salam, S.; Rafea, A. "Performance Study on Extractive Text Summarization Using BERT Models."Information, 2022, 13, 67. https://doi.org/10.3390/info13020067

[5] N.Moratanch and S.Chitrakala, "A Survey on Extractive Text Summarization", IEEE International Conference on Computer, Communication, and Signal Processing, 2018

[6] Yuanfeng Song, Di Jiang, Xuefang Zhao, Xiaoling Huang, Qian Xu, Raymond Chi-Wing Wong, Qiang Yang, "SmartMeeting: Automatic Meeting Transcription and Summarization for In-Person Conversations", In Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 2021.

[7] Siddhartha Banerjee, Prasenjit Mitra, Kazunari Sugiyama, "Generating Extractive Summaries from Meeting Transcripts", DocEng '15: Proceedings of the 2015 ACM Symposium on Document Engineering, September 2015.

[8] Pankaj Gupta, Ritu Tiwari and Nirmal Robert, "Sentiment Analysis and Text Summarization of Online Reviews: A Survey." International Conference on Communication and Signal Processing, 2016.

[9] Deepali K. Gaikwad, C. Namrata Mahender,"A Review Paper on Text Summarization",International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016

[10] Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, "Review of automatic text summarization techniques &amp; methods." J. King Saud Univ. Comput. Inf. Sci. 34, 4, Apr 2022, 1029–1046. https://doi.org/10.1016/j.jksuci.2020.05.006

[11] Emad, A., Bassel, F., Refaat, M., Abdelhamed, M., Shorim, N., & AbdelRaouf, A. (2021). Automatic Video summarization with Timestamps using natural language processing text fusion. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC).

[12] Nouf Ibrahim Altmami and Mohamed El Bachir, "Automatic summarization of scientific articles: A survey." J. King Saud Univ. Comput. Inf. Sci. 34, 4, Apr 2022, 1011–1028. https://doi.org/10.1016/j.jksuci.2020.04.020

[13] Liu, Y.; Lapata, M. Text Summarization with Pretrained Encoders. arXiv 2019, arXiv:1908.08345

[14] El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. EdgeSumm: Graph-based framework for automatic text summarization. Inf. Process. Manag. 2020, 57, 102264.

[15] Joshi, A.; Fidalgo, E.; Alegre, E.; Fernández-Robles, L. SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. Expert Syst. Appl. 2019, 129, 200–215.