Summary Generation using NLP Techniques

Sweta Gupta
Dept. of Information Technology
Vidyavardhini's College of
Engineering & Technology
Mumbai, India
shwetagupta22031607@vcet.edu.in

Yash Jobalia
Dept. of Information Technology
Vidyavardhini's College of
Engineering & Technology
Mumbai, India
jobaliayash@gmail.com

Isheet Shetty
Dept. of Information Technology
Vidyavardhini's College of
Engineering & Technology
Mumbai, India
isheetshetty@gmail.com

Anagha Patil
Dept. of Information Technology
Vidyavardhini's College of
Engineering & Technology
Mumbai, India
anagha.patil@vcet.edu.in

Abstract—**Due of the large amount of information and electronic documents available on the internet, it is impossible for a human to study, examine, and analyze this material. The main concept because the core idea of summarization allows humans to swiftly read the summary of a large amount of content and makes a decision whether or not to go deeper into the intricacies. Detailed summaries create a summary by extracting a certain collection of sentences from a document in which NLP approaches like Genism, Text rank, Rule based and Pegasus can be used on a single document or several documents. Thereupon, synopsis generation is the charge of generating incisive summaries unescorted by human assistance while conserving the genuine sense of the overlong document.**

Keywords—Speech to Text, Text Processing, Encoding, Summarization, Pegasus, Genism, Rule Based, Text rank, Decoding

I. Introduction

The human attention span is less than 8 seconds so, if you need to capture someone's attention or highlight an important topic, you need to have a strong headline or summary. A prime example of this phenomenon is that whenever you open a newspaper you glance through the headlines and read through only the ones which have captivating headlines or an interest which aligns with yours. Hence, in this ever growing and vastly expanding world with abundance of data, you need to make sure that the important topics gain your priority attention and you get precise information and knowledge from the vast abyss of data. Information is knowledge and knowledge is power. With the huge increase in the amount of data available the fine distinction between the knowledge information and data is thinning. The motive is to use the huge amount of data provided, abstract the required information and try to highlight emphasize on the necessary knowledge that it contains. Moreover no one has the luxury of spending their precious time on reading reports which are more than a few pages long just of the regular day to day meeting or a general conversation. Hence in order to save your valuable time the meeting summarizer can summarize the entire meeting into a few paragraphs and also highlight the entire gist of the content in a few lines. If the user receives an excellent summary, he or she will be able to grasp the text at a glance without having to read it completely, saving time and effort. Analysis, transformation, and synthesis [1][2] are the three processes in the text summarization process. The project's purpose is to learn about natural language processing concepts and construct a text summary machine learning tool that only incorporates the most important information from the material. The process takes place as follows. We first record the conversation or the meeting from a voice recorder on a device like phone or a mic then we use highly sophisticated STT (speech To Text) algorithms which recognise speech and convert them to text as our first input. These algorithms have the capacity to convert huge audio files with large number of recorded minutes into smaller chunks of audio files slashed at regular intervals of time in order to reduce discrepancy and for better time complexity and management with better results. This converted text file will be then encoded and decoded by algorithms by either extractive summarisation or abstractive summarisation based on the user's requirements in order to convert to convert the lengthy conversation or meetings into small summaries of the entire topic containing of the most important topics and covering the gist of the text.

II. Motivation

The amount of data these days is exponentially increasing through the internet and various other sources. To avoid browsing through these overutilized and long-drawn-out documents and converting them to succinct summary, we stand in need of a tool that can help withdraw summary by clipping of the data in these documents and giving the foremost sentences with pivotal meaning from the prolix document or from a cluster of documents. It is difficult for the human mind to reminisce all this data, so a synopsis generator plays a pivotal role to save the human effort and time. Our work aims to build a synopsis generator that provides the user the liberty to make a selection from n number of summarization methods in accordance with their needs. Furthermore humans are highly intellectual species who can communicate through sophisticated measures and in order to make a futuristic tool which can utilize these highly efficient communicative methods the tool must have additional features like converting human dictated or vocalized speech to a written readable and legible texts and hence giving them a boon of summarizing verbose and elongated speeches and conversations to minimalistic, concise and to the point summaries containing the heart of the entire passage all the while keeping them limited up to a certain precise point.

III. Related Work

This segment elucidates about the techniques that have been used for text summarization. It is one of the branches in natural processing language. Text summarization is bifurcated into categories as Extractive Summarization and Abstractive Summarization.

Extractive text summarization is to handpick must-have and imperative sentence from the text. This necessitous and important paragraph can be picked out by using linguistic and statistical features of paragraphs. Abstractive summarization learns the main concept of the long-drawn-out document and meaning of the same document or text. It discovers the new concept from the document by using linguistic method by interpreting the text. In preliminary researches, text summarization which is a part of natural processing language was carried out on scientific documents focused on the proposed features like sentence ranking. J.N. Madhuri and Ganesh Kumar R [3] have done analysis on extractive summarization by extracting upmost weighted frequency sentences. In this paper we see that after finishing the pre-processing step they calculate the frequency of each keyword like how habitually that keyword has arisen, from that greatest frequency of the keyword is taken. Then weighted frequency of the word is calculated by dividing frequency of the keywords by maximum frequency of the keywords. In this step, they calculate the sum of weighted frequencies. Finally, the summarizer extracted the high weighted frequency sentences. Similarly, Aakanksha Sharaff, Amit Siddharth Khaire and Dimple Sharma [4] analyzed rule-based logic for extractive text summarization in which first the dataset is pre-processed which is used to calculate frequency and position. This is done by using hash map and formulae after which the weight of sentences is calculated which is done by using frequency and position of words formula. The last step is analysis where the calculated values are used to find the mean for that sentence and are fed to the triangular membership function which gives values between 0 and 1 to each sentence and the fuzzy rule is applied. These values are arranged in descending order and they are picked according to the percentage of the original text needed. Finally, the ROUGE score is given to each and every summary to be compared efficiently.

In the next paper, Siya Sadashiv Naik, Manisha Naik Gaonkar [5] research about extractive text summarization by feature-based sentence extraction using rule based. The main focus of this paper is to summarize a single document and create its extractive summary. After pre-processing step, each sentence of document is represented as attribute vector of features. Seven features are calculated for each sentence and each feature is given a value from 0 to 1 after normalization. Features considered are Sentence Position, Title Feature, Numerical Value, Keyword Weight, Proper Noun, Sentence-To-Sentence Similarity and Sentence Length. Based on their ratings, all of the sentences are arranged in ascending order. Finally, the extractive summary of the document will be generated and displayed. In the following paper, Kaiz Merchant and Yash Pande [6] use latent semantic analysis approach for creating short summaries on basis of similar words. They use 2 approaches depending on the type of case if it is a criminal case, they used single document untrained approach and for civil case they used multi-document trained approach. They first preprocessed the data which is cleaning it, lemmatization and removing stop words then they pass it through the model and depending on the type of case (civil case or a criminal case) it is decided which process is used. Then using the appropriate model, they generate a summary and finally add sentence selection, in which the final line is always added because it is the judgement passed and hence the final summary output is generated.

Followed by Parth Rajesh Dedhia, Hardik Pradeep Pachgade, Aditya Pradip Malani, Nataasha Raul, Meghana Naik [7] the goal of this paper is to highlight and examine current models for abstractive text summarization, as well as to identify topics for future research. This study explains the foundations of RNN models used to construct attention models, as well as a quick overview of feature selection, attention models, pointer mechanisms, and how they work together to produce abstractive text summaries. When numerous documents are supplied to the model, the existing model fails. A system could be invented in the future to preserve the context of the previous document before moving on to the next.

Arunlfo and Ledeneva[8] using tf-idf, they suggested a method for term selection and weighting. They created a non-redundant summary using an unsupervised learning system. Mofifiz Mojib Haider and Md. Arman Hossin[9] in it for a single text, this study presents a sentence-based clustering approach (K-Means). They utilised Gensim word2vec for feature extraction, which is designed to extract semantic concepts from documents in the most efficient way possible.

Das, D. and Martins, A. F. [10] they demonstrated extractive and abstractive text summarizing techniques for single and multi-document summarization. Various strategies, such as the Naïve Bayes approach, Rich features and Decision trees, Hidden Markov methods, and Long Linear models, were used to manifest the performance depending on the data set.

Partha Mukherjee, Soumen Santra [11] they created a basic application that turns inputted text into synthetic speech and reads it out to the user, which can then be saved as an mp3 file.

Daksha Singhal [12] they used a transformer model to reduce n-gram blocking to reduce repetition and successfully presented supervised abstractive summarization on the Switchboard Dataset. The dialogues were summed up Model will be used in future projects. with a pointer generator and training on a transformer for a state-of-the-art hyper-parameter tweaking summarizer. Putting the model through its paces and assessing it on several platforms. Google Dialogue Dataset is an example of a dialogue.

Narendra Andhale [13] they gave an overview of various text summarization techniques which includes abstractive as well as extractive methods. A few of the methods which they inspect are text summarization with fuzzy logic, text summarization with neural network, which are renowned extractive summarization techniques, as well as rule based method, template based method and tree based method, which are abstractive summarization techniques.

Meena S M, Ram Kumar M P [14] they used text frequency ranking sentence prediction which is a combination of both, abstractive as well as extractive text summarization. They used rouge score, a standardized text summarization scoring system and gained good precision F-measure and recall with higher precision in abstractive text summarization.

M Indu, Kavitha K V [15] they reviewed various text summarization evaluation methods in order to evaluate the informativeness of the automatic summaries by comparing them to human made models. The two evaluation methods could be either intrinsic or extrinsic, each containing of various methods like utility method, BLEU scores, content similarity, the question game, Shannon game and last but not the least, keyword association, which were competitive candidates to rob scoring method.

Mhasa Afsharizadeh, Hossein Ebrahimpour-Komleh [16] the pre-prcoessing used by their query oriented text summarization uising sentence extraction technique extracts sentences which contain useful information and display them in the summary. They prepared data using various pre-processing methods such as tokenization, stop word removal, stemming, etc. Then they used feature extraction to extract various features such as normalized sentence length, numerical data, proper noun, topic frequency, headline feature and sentence scoring. They also used rogue scores as a summarization evaluation technique.

IV. Proposed Methodology

In this section, we represent our methodology for making an effective text summarizer for English-to-English text document. The project uses a top-down approach in which each segment is divided into smaller component and each component performs a particular role. The objective of the project is to understand the concept of natural processing language and creating a tool for text summarization containing only main points described in the document. For example, the entire project is divided into three phases: Speech to text phase and Summarization phase and Text to speech phase. The first phase aims at converting the audio files into texts. The output generated from the first phase is further used as an input for the succeeding phase which consists of summarizing these verbose texts into short excerpts along with exterminating discrepancies which may have been introduced in the previous phase due to disturbances in the audio file. Finally, the third phase is an optional feature provided to the users especially with special needs which helps them convert the short summary into an audio output. The users can use the model in various ways depending on their specific requirements and the numerous options provided in each and every phase. Furthermore, each phase also has various options provided to the users which have been discussed in detail below.

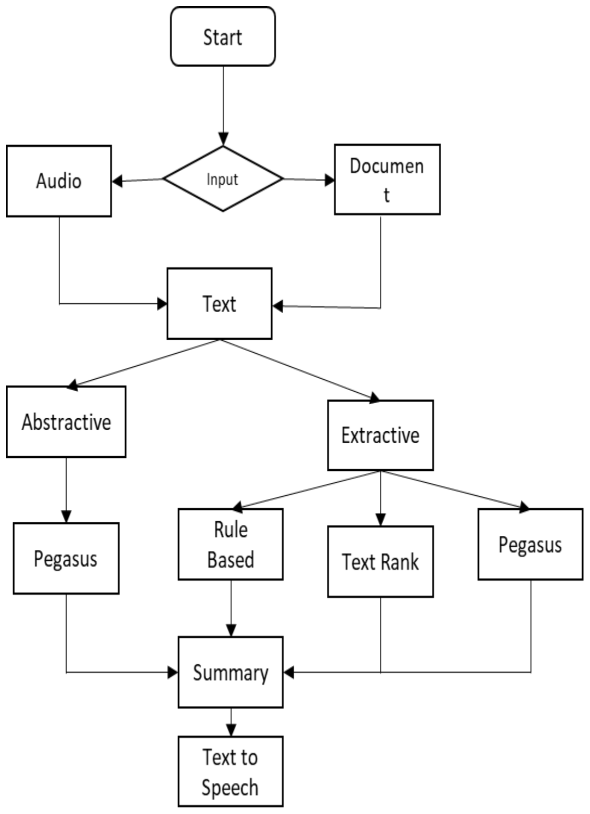


Fig. 1. Flow Diagram

A. *Speech to Text Phase*

The first phase is the speech to text phase. In this phase, the given audio input file is converted into text. This process is done through passing the audio file through an algorithm which uses google speech recognition in order to identify the audio speech and convert it into text [10].

B. *Text Summarization Phase*

The second phase is the core of the project. It deals with the summarization of the text which were the output from the previous phase. This segment gives the user the ability to select several choices depending on his/her needs of summarization. The first set of options from which the user can select the type of summarization is either abstractive or extractive. In abstractive method, the user gets further options to either summarize the gist of the text in limited lines or generate an abstractive headline. These methods try to understand the context of the input and generate the desirable output in completely new words either using Pegasus Xsum or Pegasus reddit tifu model [13].

On the counter part, the extractive method gives the user another set of choices consisting of different algorithms based on extractive summarization in which key pivotal lines are extracted from the input text and used as it is for the output. There are several extractive methods listed below which are explained in great detail in the working chapter. The extractive methods used are as follows: Gensim, Rule based, Text rank and Pegasus.

1) *Gensim*

Gensim is one of the most used free opensource python libraries for converting documents efficiently. It is a useful module for conducting natural language processing tasks. It is a mild variation of the Text Rank algorithm. The Text Rank Algorithm is used to summarize text using the gensim package [9]. Text Rank is a technique for extracting information from documents. It is based on the idea that words that appear more frequently are more important. As a result, sentences with a high frequency of words are crucial. The system then assigns ratings to each sentence in the text based on this. The top-scoring sentences are included in the summary. The first step after importing the gensim package is to import summarize from gensim summarization. Text Rank is implemented using an in-built function. Then, as an input, provide the text corpus to the summary function. The default parameters of the summary function can be changed to suit your needs. The parameters are as follows:

- It has a range of values from 0 to 1.
- It reflects the percentage of the summary that is different from the original text.
- The number of words in the summary is determined by word count.

2) Rule based

The Rule-Based Summarizer's suggested design is detailed below.

Pre-processing: The most important stage in any summarizing approach is pre-processing. Pre-processing is used to sanitize the document, eliminate noisy data, and correct grammatical problems. Tokenization, stop word elimination, and stemming are some of the preprocessing techniques used.

Tokenization: Tokenization is the process of breaking down paragraphs into sentences and then breaking down each phrase into individual words or tokens.

Stop Word Removal: After tokenization, the data is evaluated, and frequent terms or stop words such as a, an, and they are deleted from the document. All words are reduced to their root format in stemming.

Keyword Extraction: The frequency count of each word or phrase in a text is determined in the keyword extraction phase to determine its relevance. This is accomplished by computing the document's (tf-idf) scores.

Threshold: A threshold is defined during this stage. This number is derived by adding the terms with the lowest and highest frequencies and taking the mean of the two. All phrases with tf less than the threshold value are trimmed from the document after the threshold is computed.

Feature Extraction: Each phrase of the document is represented as an attribute vector of features after the pre-processing stage. Each phrase has seven characteristics, each of which is assigned a value between 0 and 1 after normalization.

Rule Generation: For each of the seven characteristics, low and high values are determined. Following that, a single rule is constructed with all feature values set to high, with the exception of sentence-to-sentence similarity, which is set to low because the summary requires fewer comparable sentences. All sentences are run via this rule once it has been written. This rule's feature values are mapped to the sentence's feature values. If there is a match, the result is 0; otherwise, the output is 1. Finally, with reference to the rule, all 1's is totaled, giving us the number of mismatching characteristics for that statement. A single score value has been assigned to each sentence.

Sentence Completion: Based on their ratings, all sentences are arranged in ascending order. To retrieve sentences, the extent of summarization is determined. Sentences are extracted to a 20 percent extent; it has been proven that extracting sentences to this percentage yields a more informative summary than extracting sentences from the entire manuscript.

Extractive Summary: The document's final extractive summary will be shown. The technique of constructing a quick and succinct summary of a source text that captures the major ideas is known as abstractive text summarization. Additional phrases and sentences not present in the original text may be included in the summaries.

3) Text Rank

Text Rank is a text summation approach used to construct Document summarizes in Natural Language Processing. Text Rank is an unsupervised graph-based text summarizing tool that uses an extractive method. PageRank is a search engine ranking algorithm that Google and other search engines use to determine the importance of webpages. Text Rank is built on top of the PageRank Algorithm. This algorithm was inspired by Larry Page, one of Google's co-founders. A page's rank and worth are determined by the number and quality of links it has. It does numerous iterations on the pages to arrive at a final value. The basic steps involved in Text Rank algorithm are as follows -

Step 1: Use any approach you like to extract all of the sentences from the text document, such as separating at whitespaces or full stops.

Step 2: The phrases retrieved in Step 1 are used to generate a graph. The weight on the edges between two nodes is computed using a similarity function such as Cosine or Jaccard Similarity, and the nodes represent texts. For this purpose, you can create your own similarity metric.

Step 3: This stage entails iterating the algorithm until convergence until consistent scores are acquired, in order to determine the significance (scores) of each node.

Step 4: The sentences are sorted in descending order based on their scores. To be included in the summary, the first k sentences of the text are picked.

4) Pegasus

Abstractive text summarization is one of the most difficult jobs in natural language processing since it requires reading long passages, compression of information, and language development. In 2020, Google AI released PEGASUS, which stands for Pre-training with Extracted Gap-Sentences for abstractive summarization. They suggest that massive Transformer-based encoder-decoder models be pre-trained using a single self-supervised objective on big text corpora. PEGASUS, an encoder-decoder-based language model, is used by the abstractive summarizer to create a semantically sound summary. PEGASUS is a gap sentence masking and summary generating pre-training approach. The PEGASUS model's architecture typically includes 15 layers of encoders and 15 layers of decoders, all of which consider text documents as input after masking. The closer the pre-training self-supervised target is to the final down-stream task, the better the fine-tuning performance, according to their premise. Several sentences are deleted from documents and the model is charged with recovering them in the proposed technique, pre-training. A document with missing sentences can be used as an example input for pre-training, with the output consisting of the missing phrases concatenated together. PEGASUS masks many complete sentences from a manuscript. These sentences have been taught to PEGASUS to anticipate. PEGASUS will recover missing sentences from an input document, and the outcome will be a concatenated list of missing sentences. The task at hand is Gap Sentence Generation. Although it features an encoder and a decoder in its fundamental design, PEGASUS main addition is Gap Sentence Generation. PEGASUS uses a pre-trained encoder as a masked language model. In PEGASUS, complete sentences are removed from a text and the model is trained to predict these sentences, as seen in the image. The authors concede that this endeavor appears to be challenging even for humans. However, such training enhances comprehension for the development of words with a relationship to the original content, so confirming their theory. Gap Sentence Generation is the name given to this endeavor (GSG). Furthermore, the authors recommend that hiding the document's most significant phrases is the most effective technique. This is accomplished by utilizing a statistic called ROUGE to identify phrases that are the most similar to the entire text (which is usually used to evaluate the quality of a summary in summarization tasks). Masked Language Model (MLM). Although, the GSG (explained in the preceding section) is PEGASUS's most important contribution, its basic architecture consists of an encoder and a decoder; hence, pretraining the encoder as a masked language model is a good idea. We mask words from sequences at random and guess the masked words using other words from the sequence in this challenge. The GSG job, which is based on this paradigm, might be considered a document-level MLM [16]. The model has been fine-tuned using 12 accessible abstractive summarization datasets. It beats the prior state of the art on six of these datasets and it did so with a little number of samples.

C. Text To Speech

This is the final phase of the project. We sometimes prefer to listen to information rather than read it. While listening to the crucial file data, we can multitask. Python has a number of APIs for converting text to voice. The Google Text to Speech API, or gTTS API, [11] is a well-known and widely used API. The program is simple to use and comes with a number of built-in features, including the ability to save a text file as an mp3 file. We don't need to employ a neural network or train a model to convert the file to voice because that is likewise difficult. Instead, we'll finish a task using these APIs. We will use another offline package called pyttsx3 to convert text files into. The gTTS function is a three-argument function. The first parameter is a text value to be converted to a speech. The second parameter is a language to use. It supports a wide range of languages. We can make an audio file out of the text. The pace of the discourse is represented by the third argument. We set the slow value to false, which implies that the speech will be delivered at usual pace. As a result, we obtain speech as a result.

TABLE I. STANDARD INPUT TEXT (EXAMPLE - FOR NUMBER OF WORDS): 2000

Models	Method	Output (word count)
Pegasus (Head)	Abstractive	96
Pegasus	Extractive	711
Pegasus(reddit-tifu Summary)	Abstractive	174
Gensim	Extractive	Ratio: 0.2=1545 0.3=2298
Rule Based	Extractive	Depends on sentences
Text Rank	Extractive	Input Top-3 = 490

V. RESULTS AND DISCUSSION

The first and the foremost step in generating a concise summary is taking the input from the user and giving the user the liberty to choose the type of input which needs processing. The Figure 2 displays the extraction of text from the audio file provided as an input by the user.

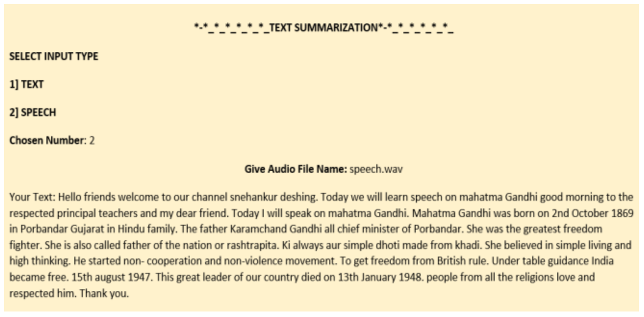

```
*-*_*_*_*_*_*_TEXT SUMMARIZATION*-*_*_*_*_*_*_

SELECT INPUT TYPE

1] TEXT

2] SPEECH

Chosen Number: 2

                    Give Audio File Name: speech.wav

Your Text: Hello friends welcome to our channel snehankur deshing. Today we will learn speech on mahatma Gandhi good morning to the respected principal teachers and my dear friend. Today I will speak on mahatma Gandhi. Mahatma Gandhi was born on 2nd October 1869 in Porbandar Gujarat in Hindu family. The father Karamchand Gandhi all chief minister of Porbandar. She was the greatest freedom fighter. She is also called father of the nation or rashtrapita. Ki always aur simple dhoti made from khadi. She believed in simple living and high thinking. He started non- cooperation and non-violence movement. To get freedom from British rule. Under table guidance India became free. 15th august 1947. This great leader of our country died on 13th January 1948. people from all the religions love and respected him. Thank you.
```

Fig. 2. Extraction of text from speech

The next step after extracting the required text is applying various algorithms according to the users needs and wishes. From the wide array of extractive and abstractive summarization algorithms the user can select which method the summarization must be performed (Extractive or Abstractive) and which specific algorithm of the selected method must be employed.

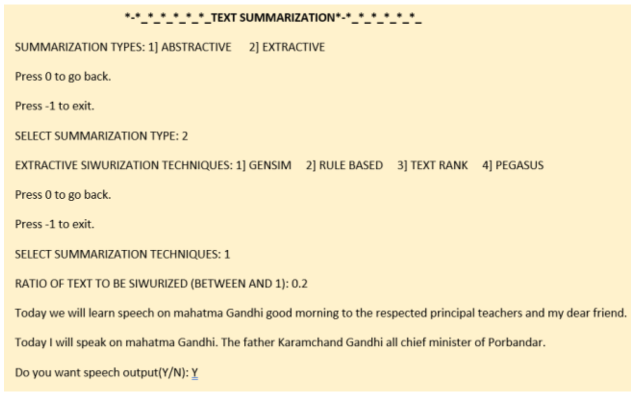
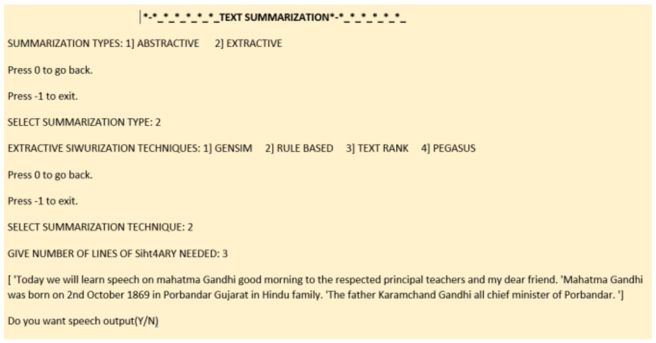

```
*-*_*_*_*_*_*_TEXT SUMMARIZATION*-*_*_*_*_*_*_

SUMMARIZATION TYPES: 1] ABSTRACTIVE     2] EXTRACTIVE

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TYPE: 2

EXTRACTIVE SIWURIZATION TECHNIQUES: 1] GENSIM    2] RULE BASED    3] TEXT RANK    4] PEGASUS

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TECHNIQUES: 1

RATIO OF TEXT TO BE SIWURIZED (BETWEEN AND 1): 0.2

Today we will learn speech on mahatma Gandhi good morning to the respected principal teachers and my dear friend.

Today I will speak on mahatma Gandhi. The father Karamchand Gandhi all chief minister of Porbandar.

Do you want speech output(Y/N): Y
```

Fig. 3. Gensim approach output

The Figure 3 shows the various extractive algoritms which can be utilized to generate a precise summary. The user has selected the gensim approach in which the model has selected and applied gensim text summarization algorithm to the input text. This algorithm gives user the ability to select the ratio of text which is required for the summary.

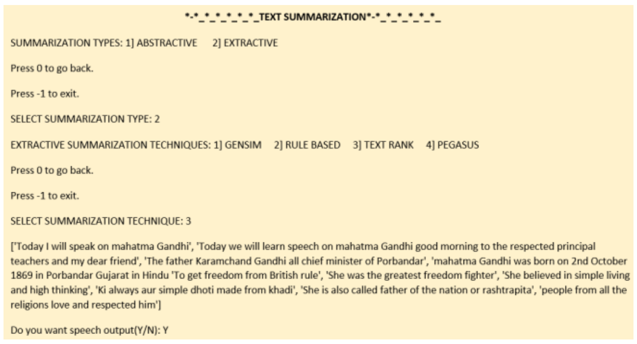
Alternatively, the user can also select the rule based algorithm of the extractive method to summarize the given input. This option as demonstrated in the Figure 4 gives the user the ability to select the "N" number of most important lines.



```
                    |*-*_*_*_*_*_*_TEXT SUMMARIZATION*-*_*_*_*_*_*_

SUMMARIZATION TYPES: 1] ABSTRACTIVE    2] EXTRACTIVE

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TYPE: 2

EXTRACTIVE SIWURIZATION TECHNIQUES: 1] GENSIM    2] RULE BASED    3] TEXT RANK    4] PEGASUS

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TECHNIQUE: 2

GIVE NUMBER OF LINES OF Siht4ARY NEEDED: 3

[ 'Today we will learn speech on mahatma Gandhi good morning to the respected principal teachers and my dear friend. 'Mahatma Gandhi
was born on 2nd October 1869 in Porbandar Gujarat in Hindu family. 'The father Karamchand Gandhi all chief minister of Porbandar. ']

Do you want speech output(Y/N)
```

Fig. 4. Rule based approach output

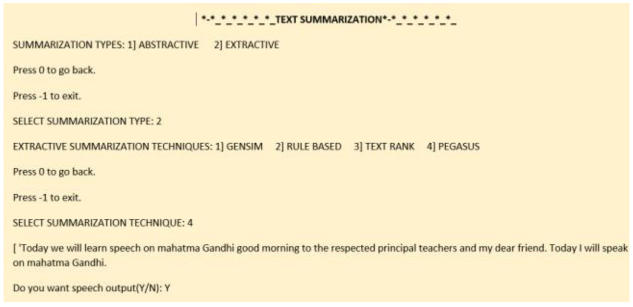
The third approach which can be selected by the user for extractive text summarization is the Text Rank algorithm as shown in Figure 5. In this algorithm, the amount of summarization is auto selected and the user lacks the ability to control the amount of summarization which may take place.



```
                         *-*_*_*_*_*_*_TEXT SUMMARIZATION*-*_*_*_*_*_*_

SUMMARIZATION TYPES: 1] ABSTRACTIVE    2] EXTRACTIVE

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TYPE: 2

EXTRACTIVE SUMMARIZATION TECHNIQUES: 1] GENSIM    2] RULE BASED    3] TEXT RANK    4] PEGASUS

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TECHNIQUE: 3

['Today I will speak on mahatma Gandhi', 'Today we will learn speech on mahatma Gandhi good morning to the respected principal
teachers and my dear friend', 'The father Karamchand Gandhi all chief minister of Porbandar', 'mahatma Gandhi was born on 2nd October
1869 in Porbandar Gujarat in Hindu 'To get freedom from British rule', 'She was the greatest freedom fighter', 'She believed in simple living
and high thinking', 'Ki always aur simple dhoti made from khadi', 'She is also called father of the nation or rashtrapita', 'people from all the
religions love and respected him']

Do you want speech output(Y/N): Y
```

Fig. 5. Text Rank approach output

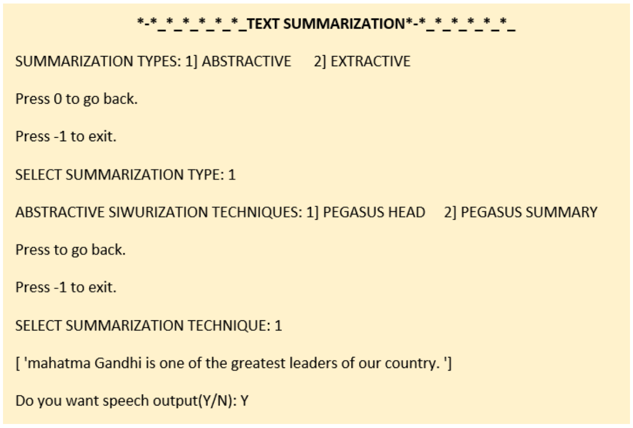
The last and the final approach extractive text summarization is Pegasus extractive text summarization method. This method uses the pre-built pegasus model to create a to-the-point summary as shown in the Figure 6.



```
                         | *-*_*_*_*_*_*_TEXT SUMMARIZATION*-*_*_*_*_*_*_

SUMMARIZATION TYPES: 1] ABSTRACTIVE    2] EXTRACTIVE

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TYPE: 2

EXTRACTIVE SUMMARIZATION TECHNIQUES: 1] GENSIM    2] RULE BASED    3] TEXT RANK    4] PEGASUS

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TECHNIQUE: 4

[ 'Today we will learn speech on mahatma Gandhi good morning to the respected principal teachers and my dear friend. Today I will speak
on mahatma Gandhi.

Do you want speech output(Y/N): Y
```

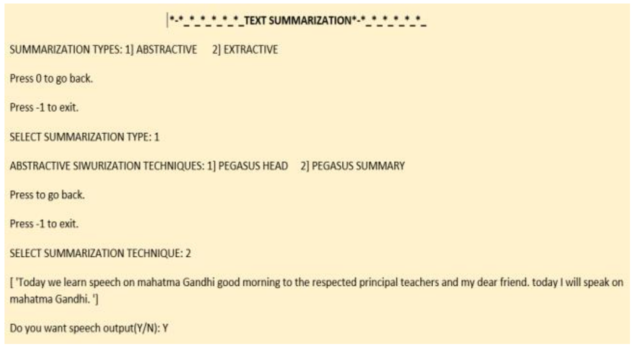
Fig. 6. Extractive Pegasus output

After exhausting through the extractive text summarization methods, we move on to abstractive text summarization methods. The abstractive text summarization uses different pegasus abstractive models to generate differnet types of summary based on the users needs which can be either the heading or the whole abstractive summary. The Figure 7 demonstrates the use of pegasus abstractive text summarization to generate a meaningful heading for the input text. On the contrary, Figure 8 displays the use of pegasus abstractive text summarization to generate the summary of the entire input text.



```
                    *-*_*_*_*_*_*_TEXT SUMMARIZATION*-*_*_*_*_*_*_

SUMMARIZATION TYPES: 1] ABSTRACTIVE    2] EXTRACTIVE

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TYPE: 1

ABSTRACTIVE SIWURIZATION TECHNIQUES: 1] PEGASUS HEAD    2] PEGASUS SUMMARY

Press to go back.

Press -1 to exit.

SELECT SUMMARIZATION TECHNIQUE: 1

[ 'mahatma Gandhi is one of the greatest leaders of our country. ']

Do you want speech output(Y/N): Y
```

Fig. 7. Abstractive Pegasus Head output



```
                    |*-*_*_*_*_*_*_TEXT SUMMARIZATION*-*_*_*_*_*_*_

SUMMARIZATION TYPES: 1] ABSTRACTIVE    2] EXTRACTIVE

Press 0 to go back.

Press -1 to exit.

SELECT SUMMARIZATION TYPE: 1

ABSTRACTIVE SIWURIZATION TECHNIQUES: 1] PEGASUS HEAD    2] PEGASUS SUMMARY

Press to go back.

Press -1 to exit.

SELECT SUMMARIZATION TECHNIQUE: 2

[ 'Today we learn speech on mahatma Gandhi good morning to the respected principal teachers and my dear friend. today I will speak on
mahatma Gandhi. ']

Do you want speech output(Y/N): Y
```

Fig. 8. Abstractive Pegasus Summary output

VI. CONCLUSION

As a result of implementing various extractive and abstractive methods of text summarization, each having its own perks and swindles we can conclude the efficiencies and the conclusions of enacting various techniques. In recent days, several studies on the creation of summaries from numerous documents have been done. Text summarizing generates a summary comprising essential phrases and all pertinent important information from the source material automatically. According to the summary data, extractive and abstractive techniques are among the most common. Text summary has a number of key advantages, including the following: They make reading easier. It saves you time. It makes it easier to remember information. It improves work rate efficiency. The basic objectives of text summarization are as follows: -Optimal topic coverage Maximum readability. There are a few evaluation standards to assure these two aspects. One of these is salience, or the retention of the most significant component. To capture the most significant information from the original document, a summarizer must be programmed. The final summary must be exactly the right length. It should be neither too long nor too short. The structure must be user-friendly. The sentences must be logical and understandable. It should not include any unusual pronouns. The summary as a whole must be balanced. This implies that it must include all of the main components of the paper and, of course, be grammatically perfect throughout. Finally, the phrases should not be repetitive. If a summarizer meets all of these characteristics, it will be able to generate reader-friendly summaries that can benefit us in a variety of ways. Text summary has become an important component of the daily lives of academics, students, and anyone who work with large amounts of text. With advancement of technology and AIso, it is likely that one day, automatic text summarizing may be as excellent and clear as manual text summarization.

VII. Future Work

For future works, the current project works only and only for English language but in future we can expand this in such a way that it can support not only various languages but also inter language summarization. Apart from the single document summarization, we can enhance this project to summarize multiple documents at the same time. The time complexity of converting speech to text can be modified to achieve a faster and more accurate level. The input type for speech to text currently only accepts wav files. This drawback needs to be amended such that it can support various file types. A wide array of input types such as pdf, docx, jpg, etc. can also be used as a supplement to the current input types.

References

[1] U. Hahn and I. Mani, "of Automatic Researchers are investigating summarization tools and methods that," in IEEE Computer 33.11, no. November, pp. 29–36, IEEE, 2000.

[2] K. Sparck Jones, "Automatic summarising: The state of the art," Information Processing & Management, vol. 43, pp. 1449–1481, Nov 2007.

[3] J.N.Madhuri, Ganesh Kumar "Extractive Text Summarization Using Sentence Ranking", Institute of Electrical and Electronics Engineers (IEEE), 2019.

[4] Aakanksha Sharaff, Amit Siddharth Khaire, Dimple Sharma," Analyzing fuzzy based approach for extractive text summarization", International Conference on Intelligent Computing and Control Systems (ICICCS 2019),

[5] Siya Sadashiv Naik, Manisha Naik Gaonkar," Extractive Text Summarization by Feature based sentence extraction using rule based", IEEE International Conference On Recent Trends in Electronics Information Communication Technology (RTEICT),2017,

[6] Kaiz Merchant, Yash Pande," NLP Based Latent Semantic Analysis for Legal Text Summarization", IEEE,2018.

[7] Parth Rajesh Dedhia, Hardik Pradeep Pachgade, Aditya Pradip Malani, Nataasha Raul, Meghana Naik, "Study on Abstractive Text Summarization Techniques", International Conference on Emerging) Trends in Information Technology and Engineering, 2020.

[8] R. A. Garcıa-Hernandez and Y. Ledeneva, "Word sequence models for single text summarization," in Proceedings of the 2nd International Conferences on Advances in Computer-Human Interactions, ACHI 2009, pp. 44–48, IEEE, 2009.

[9] Mofiz Mojib Haider, Md. Arman Hossin," Automatic Text Summarization Using Gensim Word2Vec and K-Means Clustering Algorithm", IEEE, 2020.

[10] D. Das and A. Martins, "A survey on automatic text summarization. literature survey for language and statistics," II Course at CMU, 2007.

[11] Partha Mukherjee, Soumen Santra, Subhajit Bhowmick,"Development of GUI for Text-to-Speech Recognition using Natural Language Processing", IEEE, 2018.

[12] Daksha Singhal, Kavya Khatter, Tejaswini A , Jayashree R," Abstractive Summarization of Meeting Conversations", IEEE, 2020.

[13] Narendra Andhale, L.A. Bewoor, "An Overview of Text Summarization Techniques", IEEE, 2016.

[14] Meena S M, Ramkumar M P, "Text Summarization Using Text Frequency Ranking Sentence Prediction", ICCCSP, 2000.

[15] M Indu, Kavitha K V, "Review on text summarization evaluation methods", IEEE, 2020.

[16] Mahsa Afsharizadeh, Hossein Ebrahimpour-Komleh, Ayoub Bagheri "Query-oriented Text Summarization using Sentence Extraction Technique", ICWR, 2018.