Article and YouTube Transcript Summarizer Using Spacy and NLTK Module

Reshma Shaik
Department of Information Technology
G. H. Raisoni College of Engineering, Nagpur, Maharashtra
reshmashaik4567@gmail.com

Saloni Bargat
Department of Information Technology
G. H. Raisoni College of Engineering,Nagpur, Maharashtra
salonibargat726@gmail.com

Prof. Shilpa Ghode
Department of Information Technology
G. H. Raisoni College of Engineering,Nagpur,Maharashtra
shilpa.ghode@raisoni.net

***Abstract*—Summarization tools have become increasingly popular among students and professionals as they can save a considerable amount of time by generating summaries quickly and efficiently. With the help of these tools, individuals can shorten lengthy texts without having to go through the tedious process of reading and summarizing the information themselves.**

The usefulness of summarizers is not limited to merely reducing the length of the text. They can also help individuals generate brief and concise summaries of their work that are easier to read and understand. This is especially beneficial for professionals who need to communicate complex ideas and concepts to a wider audience.

Moreover, summarizers are versatile in that they can generate summaries of various lengths, depending on the needs of the individual. Some tools can provide a one-sentence summary, while others can generate a more detailed summary that covers all the important points of the text.

Overall, summarization tools have proven to be a valuable asset for students and professionals alike, enabling them to streamline their work and save time while still producing high-quality summaries.

Keywords— Text summarization, Extractive text summarization, Natural language processing

I. INTRODUCTION

Summarization is the process of creating a condensed version of a longer piece of text, such as an article, report, or document. The goal of summarization is to distill the most important information from the original text while removing any unnecessary or redundant details. This can be useful for a variety of purposes, such as quickly understanding the key points of a lengthy document or providing an overview of a complex topic. A summarizer is a tool or program that automates the process of summarization. There are various types of summarizers, ranging from simple rule-based systems to more complex machine learning models. Some summarizers use natural language processing (NLP) techniques to identify important sentences or phrases in the original text, while others may use statistical methods or other approaches. Summarizers can be trained on specific types of text, such as news articles or scientific papers, or they can be more general purpose. Overall, summarization and summarizers are important tools for quickly and efficiently extracting the most important information from a large amount of text Certainly! An application that can summarize long paragraphs is a type of software program or tool that can automatically generate a shorter version of a longer text. This is achieved through a process known as automatic summarization, which involves using algorithms and natural language processing (NLP) techniques to identify the most important information in the original text and condense it into a shorter form.

There are two main types of automatic summarization: extractive and abstractive. Extractive summarization involves selecting and combining the most relevant sentences or phrases from the original text to create a summary. This approach is typically simpler and faster, and can be more effective for summarizing factual texts, such as news articles or scientific papers.

Abstractive summarization, on the other hand, involves creating a summary by generating new sentences that capture the key information in the original text. This approach can be more challenging, as it requires the system to generate new language that is both grammatically correct and semantically meaningful. However, it can be more effective for summarizing more complex texts, such as opinion pieces or literary works.

Abstractive summarization, on the other hand, involves creating a summary by generating new sentences that capture the key information in the original text. This approach can be more challenging, as it requires the system to generate new language that is both grammatically correct and semantically meaningful. However, it can be more effective for summarizing more complex texts, such as opinion pieces or literary works.

Applications that can summarize long paragraphs can be useful in a variety of contexts, such as in business settings for summarizing reports or in educational settings for summarizing academic papers. They can also be helpful for individuals who need to quickly extract the most important information from a large amount of text, such as when conducting research or studying for exams.

As we mentioned earlier, abstractive summarization involves generating new sentences that capture the key information in the original text. This approach can be more challenging than extractive summarization, as it requires the system to not only identify the most important information but also to generate language that is both grammatically correct and semantically meaningful. Because of these challenges, errors can sometimes occur in the output of an abstractive summarization system. For example, the system may generate a summary that is grammatically incorrect, semantically unclear, or simply inaccurate in its representation of the original text.

To address this issue, researchers have developed methods for automatically detecting errors in abstractive summarization output. These methods typically involve comparing the output summary to the original text and identifying areas where there are discrepancies or inaccuracies. One common approach is to use metrics such as ROUGE (Recall-Oriented Understudy for Gisting Evaluation) or BLEU (Bilingual Evaluation Understudy) to measure the overlap between the output summary and the original text. These metrics can provide a quantitative measure of the quality of the summary and can be used to identify areas where the system may have made errors.

Another approach is to use human evaluation, in which human judges are asked to assess the quality of the output summary. This can be more time-consuming and expensive than automated methods, but it can provide a more nuanced understanding of the errors that the system is making and can help guide further improvements to the summarization algorithm. Overall, an application that can detect errors in abstractive summarization can be a valuable tool for improving the quality and accuracy of machine-generated summaries. By identifying areas where the system is making errors, developers can work to refine the algorithm and improve the overall performance of the system.

II. GENERAL DESCRIPTION

Our project comes under the domain of natural language processing, which is helpful for summarizing research papers and also YouTube videos. The domain of natural language processing (NLP) is a branch of artificial intelligence that focuses on enabling computers to understand, interpret, and generate human language. NLP has numerous applications in various fields, including summarization of research papers and YouTube videos.

In the case of research papers, NLP techniques can be utilized to summarize the content in a shorter, more manageable format. This is particularly useful for individuals who need to review large volumes of research papers, as it enables them to quickly extract the essential information from the papers without having to read through the entire text.

Similarly, NLP can be used to summarize YouTube videos, which are a popular source of information for many people. With the help of NLP techniques, the content of a YouTube video can be analyzed and summarized into a shorter, more digestible format. This can be particularly useful for individuals who are short on time or who are looking for a quick overview of the video's content. Overall, the application of NLP in summarizing research papers and YouTube videos is a promising development that can significantly improve efficiency and productivity for individuals working in various fields.

The main goal of the project is to help people save time when they are unable to dedicate significant amounts of time to reading lengthy texts, but still require essential information from them. This can be particularly useful for individuals who have busy schedules or who need to review large amounts of information quickly. The project provides an automated way of generating summaries of lengthy texts, such as articles, research papers, or reports. This is accomplished through the use of natural language processing (NLP) techniques, which enable the system to identify and extract the most important information from the text and present it in a condensed format.

By using this project, individuals can save time by obtaining the critical information they need without having to read through the entire text. This can be particularly useful for professionals, such as researchers, journalists, or academics, who need to review large volumes of information quickly and efficiently. The project's ability to generate concise summaries of lengthy texts in a short period can significantly improve productivity and enable individuals to manage their time more effectively. This project has the potential to revolutionize the way people consume and interact with information, making it easier and more efficient to obtain the necessary knowledge quickly and effectively.

GUI

GUI stands for Graphical User Interface, which refers to the visual interface that allows users to interact with a computer or electronic device. The GUI includes graphical elements such as icons, buttons, menus, and windows that users can manipulate using a mouse, touchpad, or other pointing device. A well-designed GUI provides a user-friendly experience, making it easy for users to navigate and operate a computer or device. The use of visual elements and intuitive design can help users quickly and easily find the information they need and complete tasks efficiently.

In software development, the GUI is an essential component of the user interface, and designers must consider usability, accessibility, and visual appeal when creating GUI elements. GUI frameworks and libraries such as Qt, JavaFX, and Tkinter provide developers with pre-built components and tools for creating GUI interfaces for their applications.

Overall, GUI plays a crucial role in providing an efficient and user-friendly interface for interacting with computers and electronic devices, making it easier for users to complete tasks, access information, and interact with software applications.

III. RELATED WORK

There had been much previous research work on automatic text summarization using natural language processing. Text Rank algorithm and text summarization using the k-means clustering is the automatic extractive text summarization techniques that use the inverse document frequency vectorizer technique(TFIDF)[9] or Count vectorizer to encode the text document for further scoring and ranking of text sentences in Single text Document. That was the basic approach which was followed by many NLP researcher for text data encoding in extractive summarizer approach. Using this vectorizer approach to encode the text data and then scoring and generating summary without knowing the word environment will lead to ambiguous summaries. For that reason, the contextual Elmo embedding has been used for text data encoding in extractive text summarizer.

There are several existing projects in the field of text summarization that utilize a variety of techniques and approaches. Some notable examples include:

- GPT (Generative Pretrained Transformer) - a language model developed by OpenAI that can be fine-tuned for text summarization.
- BERT (Bidirectional Encoder Representations from Transformers) - a language model developed by Google that can also be fine-tuned for text summarization.
- Sumy - an open-source library for Python that uses a variety of techniques for summarizing text, including LexRank and TextRank.
- TextTeaser - an open-source project that uses an unsupervised machine learning algorithm to generate summaries of news articles.
- SMMRY - a web-based summarization tool that uses a unique algorithm to generate summaries of any text input.
- Aylien - a text analysis API that includes a summarization feature that can generate summaries of news articles, blog posts, and other types of content.

Overall, these projects demonstrate the wide variety of approaches and techniques that can be used in text summarization, from unsupervised machine learning algorithms to pre-trained language models.

IV. EASE OF USE

For local files, users can simply upload the file to the summarizer tool, and it will generate a summary of the text within the file. This can be particularly useful for users who have large documents that they need to review quickly.

For web content, users can provide the URL of the webpage they want to summarize, and the summarizer tool will extract the relevant text and generate a summary. This can be particularly useful for researchers or professionals who need to review a large number of web pages quickly and efficiently. Once the summarizer generates the summary, users can save the output in a file format of their choice, such as TXT or PDF, making it easy to review and share the summarized content with others.

Overall, our summarizer project provides a versatile and efficient solution for generating summaries of various types of content, making it easier for users to manage and review information quickly and efficiently.

V. METHODOLOGY

A. Article Summarizer

The use of summarizers has become increasingly important in today's fast-paced world, where individuals and organizations need to process large amounts of information quickly and efficiently. Summarizers can help people save time, improve comprehension, and make better decisions by providing a concise and accurate summary of a text.

Suppose there is a document or a research paper you want to summarize then through the extractive technique we have the following steps:

1)Convert the paragraph into sentences: Let's divide the paragraph first into the sentences that go with it. Finding a sentence to extract whenever a period arrives is the best technique to convert text.

2)Pre-processing: In this we remove the stop words which are usually too common and do not carry much semantic meaning or context. Removing stop words helps to reduce noise and improve the efficiency of these techniques by focusing on the more meaningful words in the text. Examples of stop words include "the", "a", "an", "and", "in", "of", "to", "is", "that", "for", "with", "it", "on", "at", and "as".

Also, we eliminate any special characters and numerals from the research paper as well as any references to other studies.

3)Tokenization: Tokenize all the sentences to get all the words that exist in the sentences.

4)Find Weighted Frequency of Occurrence: Now, we find the weighted frequency of each word by dividing its frequency by the frequency of the most occurring word.

5)Replace each of the words found in the original sentences with their weighted frequencies. Then calculate the sum.

It is not essential to add the inconsequential words that were eliminated during the processing stage, such as stop words and special characters, because their weighted frequencies

are zero.

6)Finally Sort sentences in Descending order of Sum.

B. Text Summarization of a website article

Now, we want a summarization of information present in a website such as Wikipedia. This process is similar to text summarization but firstly we have to fetch the articles from the URL.

1)Fetch Articles from the url: For this we use library such as beautiful soup that helps in web scraping. Beautiful Soup converts the incoming text to Unicode characters and the outgoing text to UTF-8 characters.

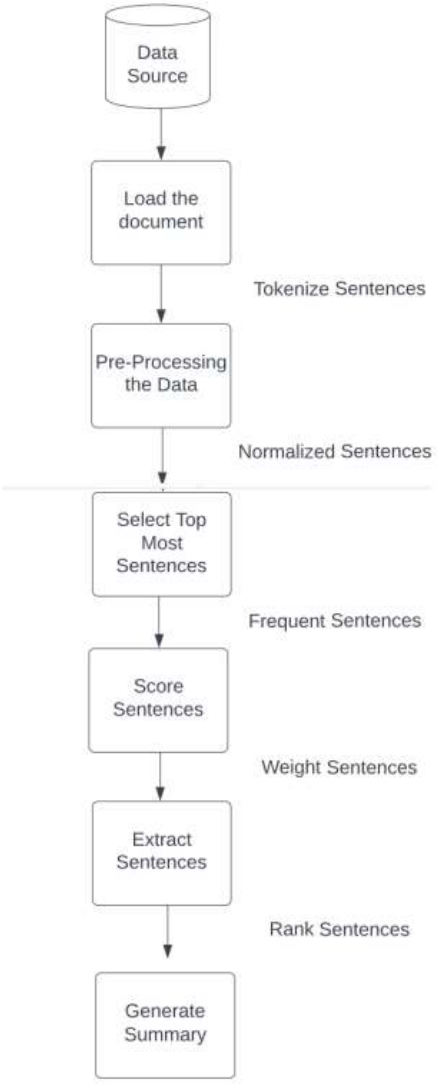
2)Processing the data: We will now remove the stop words and also special characters or numerals present. We'll compile a dictionary table with the frequency at which each term appears in the text. To get rid of any stop words, we'll iteratively loop through the text and the associated terms.

Following that, we'll see if the words are listed in the frequency table. The word's value is updated by 1 if it was previously listed in the dictionary. Otherwise, its value is set to 1 if it is heard for the first time.

3) Tokenization: Tokenize all the sentences to get all the words that exist in the sentences.

4) Finding the weighted frequencies of the sentences: Crucially, we divided each score of a sentence by the number of words in that sentence to ensure that long sentences did not receive excessively high scores compared to short sentences.

5) Determining the sentences' threshold: We'll calculate the average score for the sentences to further refine the categories of sentences that are appropriate for summary. By using this threshold, we may avoid choosing the sentences that scored below average.



Flowchart of preprocessing the data

Finally, since we have all the necessary inputs, we can now produce an article summary.

C)File Summarizer

In this if we have a file saved in the computer, it can be an article or research paper that is not available on internet or we just want the file to be uploaded then it can also be done by opening the file we have saved in the computer and get the text present in it. Then similar to text summarizer we can get summary for this file. We can also save the summary in a file if we want.

D)YouTube Transcript Summarizer

As we all know some lectures on YouTube or lengthy

videos which we want to summarize so that it can be read easily. For this we first get the YouTube video URL and then through an API we get the transcript or subtitles present in the YouTube video.

Then similar to text processing we convert the text that is paragraphs into sentences then we remove the stop words and tokenize the sentences and also, we perform stemming through which we get the root words. Then we find the weighted frequency of occurrence that is the count of each word which are repeated through the text.

To find probability of each word occurring we divide its frequency by the frequency of the most occurring word.

Now, replace each word found in the original sentences with their weighted frequencies and calculate the sum. And then sort sentences in Descending order of sum.

E) YouTube Video Summarizer

There are YouTube videos present which do not have transcript or subtitles then how do we summarize it? For this we use hugging face library.

The following steps are carried out for this process:

1) Download the audio of the YouTube video: For this, we use the PyTube library. We then import the video by providing its URL, keeping in mind that it must be of the mp4 format.

2) Convert the speech or audio into text: Automatic speech recognition or speech to text. We use a library called HuggingSound. And from this library we import speech recognition model. Later we also do audio chunking to avoid out of memory error.

3)Summarize the text: After getting the text we simply summarizer by applying the same steps as we did in text or article summarizer

VI. RESULTS AND DISCUSSION

The anaconda environment and the Spyder console are used to write this program, which was written in the Python programming language. For text processing, the NLTK package is utilized.

Spacy module performs various NLP tasks such as tokenization, part-of-speech tagging, named entity recognition, and other linguistic features. Using Spacy built in functionalities such as part-of-speech tagging and named entity recognition we can calculate the frequencies of each word.

Similarly, The Natural Language Toolkit (NLTK) is a popular open-source library for Natural Language Processing (NLP) in Python. NLTK also includes a variety of corpora, or large collections of text, that can be used for training and testing NLP models. Additionally, NLTK provides an easy-to-use interface for accessing and processing these corpora. The key difference between spacy and nltk is that the spacy module supports multiple language. SpaCy is known for its high performance and efficiency whereas NLTK can be slower and less memory-efficient in some cases.

A quick and efficient summarizing method is required because there is such a huge growth in the volume of content that is available online. This approach is applicable to many other industries, including education, question generating, and many other application-focused disciplines.

VI. CONCLUSION AND FUTURE SCOPE

In our project we have provided one platform for all summarizers such as article, file, URL and YouTube video summarizer. With help of tkinter and Python we have made a graphical user interface (GUI) which provides one platform for all these summarizers and also we have used genism module which is a powerful tool for working with large collections of text data and performing unsupervised machine learning tasks such as topic modeling and document similarity analysis. It is widely used in both research and industry applications. Also, with the help of Spacy and nltk we have made our summarizer more accurate and easier to use. We have also used transformers which is also an open-source library for Natural Language Processing (NLP) developed by Hugging Face.

Future plans for this extractive summarizer include making one that is considerably more accurate and has a faster method of processing data. Additionally, this summarizer can be improved by processing numerous papers at once and providing a generalized summary. The title of the summary can also be used to describe the text, which will boost the accuracy of the summary.

The title of the summary describes the text, such as what the material is about. Extractive summarization involves selecting sentences from the source document to create a summary. However, abstractive summarization involves generating new sentences that capture the essence of the original document. Future work could explore techniques for combining extractive and abstractive summarization to create more informative and readable summaries. extractive summarization techniques are typically developed for a specific language. However, in an increasingly globalized world, there is a growing need for summarization techniques that can work across multiple languages. Future work could explore techniques for cross-lingual summarization.

REFERENCES

[1] P. P. Balage Filho, T. A. Salgueiro Pardo and M. das Gracas Volpe Nunes, "Summarizing Scientific Texts: Experiments with Extractive Summarizers," Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), Rio de Janeiro, Brazil, 2007, pp. 520-524, doi: 10.1109/ISDA.2007.92.

[2] A. Lukyamuzi, J. Ngubiri and W. Okori, "Topic Based Machine Learning Summarizer," 2019 IEEE International Smart Cities Conference (ISC2), Casablanca, Morocco, 2019, pp. 288-291, doi: 10.1109/ISC246665.2019.9071737.L.

[3] S. S. Naik and M. N. Gaonkar, "Extractive text summarization by feature-based sentence extraction using rule-based concept," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 2017, pp. 1364-1368, doi: 10.1109/RTEICT.2017.8256821.

[4] P. P. Balage Filho, T. A. Salgueiro Pardo and M. das Gracas Volpe Nunes, "Summarizing Scientific Texts: Experiments with Extractive Summarizers," Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), Rio de Janeiro, Brazil, 2007, pp. 520-524, doi: 10.1109/ISDA.2007.92.

[5] A. P. Patil, S. Dalmia, S. Abu Ayub Ansari, T. Aul and V. Bhatnagar, "Automatic text summarizer," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Delhi, India, 2014, pp. 1530-1534, doi: 10.1109/ICACCI.2014.6968629.

[6] Vishal gupta, " A survey of text summarization extractive techniques",Journel of emerging technologies in web intellegence**, vol 2, No**, 3 august **2010**

[7] A. N. S. S. Vybhavi, L. V. Saroja, J. Duvvuru and J. Bayana, "Video Transcript Summarizer," 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 2022, pp. 461-465, doi: 10.1109/MECON53876.2022.9751991

[8] R Sanjana et al. "Video Summarization using NLP" International Research Journal of Engineering and Technology (IRJET) 2021.

[9] K. Kulkarni and R. Padaki, "Video Based Transcript Summarizer for Online Courses using Natural Language Processing," 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2021, pp. 1-5, doi: 10.1109/CSITSS54238.2021.9683609.

[10] S. JUGRAN A. KUMAR B. S. TYAGI and V. ANAND "Extractive Automatic Text Summarization using SpaCy in Python NLP" 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) pp. 582-585 2021.

[11] P. Batra S. Chaudhary K. Bhatt S. Varshney and S. Verma "A Review: Abstractive Text Summarization Techniques using NLP" 2020 International Conference on Advances in Computing Communication Materials (ICACCM) pp. 23-28 2020.

[12] D Supreetha S B Rajeshwari and Jagadish S Kallimani "Abstractive Text Summarization Techniques" International Journal of Engineering Science and Computing vol. 10 no. 7 2020.

[13] Vishal Gupta, G.Sl Lehal, " A Survey of Text Mining Techniques and Applications", Journal of Emerging Technologies in Web Intelligence,
VOL. 1, NO. 1, pp- **60-76**, **August 2009**

[14] Chin-yew Lin, "A package for automatic evaluation of summaries",in Proc. ACL workshop on text summarization branches **O UT**,**2004**.

[15] Mayank Patel, Neelam Badi, Amit Sinhal (2019). The Role of Fuzzy Logic in Improving Accuracy of Phishing Detection System. International Journal of Innovative Technology and Exploring Engineering, **Volume-8 Issue-8**, ISSN: 2278-3075, pp. **3162-3164.**

[16] J. N. Madhuri and R. Ganesh Kumar "Extractive Text Summarization Using Sentence Ranking" 2019 International Conference on Data Science and Communication (IconDSC) pp. 1-3 2019.

[17] 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) pp. 1803-1807 2018.

[18] Rahim Khan Yurong Qian and Sajid Naeem "Extractive based Text Summarization Using KMeans and TF-IDF" International Journal of Information Engineering and Electronic Business(IJIEEB) vol. 11 no. 3 pp. 33-44 2019.

[19] https://www.mdpi.com/2076-3417/12/9/4479

[20] Divya Santwani Akash Bedi and Mohit Bahrani "Textizer" International Journal of Engineering Research and Technology vol. 9 no. 07 July 2020.

[21] Verma Pradeepika and Verma Anshul "Accountability of NLP Tools in Text Summarization for Indian Languages" Journal of scientific research vol. 64 pp. 258-263 2020.