Comparative Analysis of Extractive Summarization Using NLP

Anjana Mishra
Dept. Of Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar
anjanamishra@gmail.com

Ashutosh Ray
Dept. Of Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar
manoj.ray.puri@gmail.com

Bisal Sahoo
Dept. Of Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar
bisalkumar2001@gmail.com

Kirti Tung
Dept. Of Computer Science and Engineering
Raman Global University
Bhubaneswar
kirtitung@gmail.com

Pritam Kar
Dept. Of Computer Science and Engineering
C.V. Raman Global University
Bhubaneswar
pritamkar2000@gmail.com

***Abstract*— The process of constructing short, easy to read, understands, and to the context summary of an extensive text document is called text summarization. The principal role is to acquire the correct amount of information within a period of time. Text report is finished either by humans, which requires experience there in space, also terribly tedious and time overwhelming. Along with the advancement of technology, the amount of text data is difficult to understand and is not available in a structured manner due to the larger number of characters. Hence, this is a necessary tool for today. It has been divided into two subparts, text summarization with an abstractive approach or else known as Abstractive Text Summarization (ATS) and the second one summarization with an extractive approach or else known as the Extractive text summarization (ETS). The second approach is very much manageable and efficient in comparison to ATS. ETS mainly focuses on working and hence, it takes out crucial tokens and may be sometimes sentences which are considered as the tokens from the text document which is considered as the input. Hence, the other technique creates a summary by itself.**

Keywords — Summarization, ETS, ATS, Tokens, Document

I. INTRODUCTION

In the era of technology, the proportion of data, information and sources is getting larger and larger inch by inch. These things are incrementing in the form of pictures, text and videos, etc[1]. For people who want very accurate information and precise information within less time, it is a bigger trauma due to this overflowing data[2]. If a person wants information about a specific thing, on a web page he will face very irrelevant and unnecessary documents which are not needed[3]. These all drawbacks lead to time waste, resource waste and effort loss[4]. To bring peace and enhancement in the equilibrium of life, content summarization using NLP is the game changer. This enhancement or a small requirement is aimed to bring a generic upliftment in the quality of human life. The struggling people thirst for fruitful information on this challenging life can boost their life skill with the help of a proper content summarization algorithm. The push here is to enhance the existing text summarization methods in order to bring a more sophisticated, fruitful and commercially useful content summarization technique with more flexibility of changing output sizes as per desire and help a busy person to listen to the summary also.

To make a comparative analysis, it has been decided to go with an extractive based approach which will be more reliable and efficient. In this approach we look to follow a frequency based method which will generate a resource oriented summary. Following this approach will be used to address answers to issues of many universities. NLP[5], NLTK[6], Spacy[7], Sumy[8] and Cosine Similarity are the technology used here.

II. LITERATURE SURVEY

Text summarization is an essential task in natural language processing (NLP) that aims to condense large volumes of text into shorter summaries while retaining the key information and the main advantage was the Recall Oriented Understudy for Gisting Evaluation - ROGUE metric still it is complex as it focused more on directed and undirected graphs[9].

Text and audio summarization are crucial tasks in natural language processing (NLP) that aim to condense information from text and audio files into concise summaries. This literature survey focuses on an efficient model capable of producing summaries of varying lengths for both text and audio files. Gensim library can be incorporated into this learning as it enables topic modeling and word embeddings . Summaries generated by both human experts and the system is carried out using the ROUGE toolkit, which measures recall, precision, and F-measure[10]. However, the use of a predefined K value is a potential limitation that needs to be addressed for improved flexibility and wider applicability. The reliance on a predefined K value restricts its adaptability and generalizability. Models such as Hybrid Text Document Summarization based models depending upon WordNet ontology can bridge the gap between extractive and abstractive summarization[11]. The hybrid approach that combines machine learning and clustering techniques for extractive document summarization shows promise in generating informative summaries. While the approach's ability to learn independently of feature space dimensionality is advantageous, addressing the limitations related to measuring feature complexity is crucial for further improvement in summarization performance[12]. By employing two different methodologies, it provides insights into the effectiveness of various techniques for opinion summarization. The evaluation using ROUGE-1 and ROUGE-2 scores enables a quantitative assessment of the summarization quality, allowing for a fair comparison between the methodologies. While the evaluation using ROUGE scores provides quantitative assessments, the potential drawback of missing important parts emphasizes the need for further research and advancements in this area[13]. The surveyed approach of automatic text summarization using NLP and reinforcement learning-based algorithms demonstrates advantages in improving summarization performance and enabling abstractive summarization. However, challenges related to balancing relevant information, redundancy, coherence, query relevance, and representation need to be addressed for further advancements in the field[14]. The use of maximum relevance extractive text summarization methods with multi-document datasets shows promise in generating relevant summaries. While the approach offers advantages in improving summarization relevance, careful attention should be given to balancing reinforcement to avoid potential drawbacks[17]. the implementation of automatic text summarization using Spacy NLP demonstrates advantages in utilizing advanced parser/tagger models. While the approach offers advantages, the potential limitations in tokenization performance for non-canonical language and unconventional text formats should be considered.

Also some survey papers were considered which depicted the comparisonbetween the traditional methods of summarization and cosine similarity between the input text and output summary. The other summarization methods uses all the information from the input text and uses it for summarization algorithm in the extractive based approach[17].

III. METHODOLOGY

The methodology for the project implementation involved the utilization of various technologies and frameworks, including HTML, CSS, JavaScript, and Python languages, along with Flask and NLP technology/framework. To begin with, a virtual environment was created to ensure effective working with the Flask framework. The main file was developed to handle routing and imported essential libraries For text processing, the NLTK tokenizer downloaded to break the text into sentences. Additionally, an unsupervised algorithm was used to build a model for handling abbreviations, collocations, and sentence starters. An "index.html" file was established under the "templates" folder, allowing users to input text through a text area. A slider was implemented to determine the desired length of the summary, and submit and reset buttons were provided. To extract data from webpage, the Beautiful Soup library was utilized. The process involved sending a GET request to the webpage, retrieving the HTML content, creating a Beautiful Soup object, and extracting the text data using the get_text() method. This approach enables further processing and extraction of specific information from webpage.

A. Lemmatization

As shown in Fig. 1. The input text is passed in the form of a sentence as a collection of strings , then all the punctuations like !,",#,$,%,',*,+ etc. is removed and then tokenization is performed on the collection of strings . In tokenization text of string is spitted into a list of tokens i.e. key words are pointedout. To determine the best sentences for the summary, the heapq library was employed. By using a priority queue, the sentences were ordered based on their scores, with the highest-scoring sentences being selected. Finally, the selected sentences were combined to generate the summary, capturing the key information and main points of the original text.

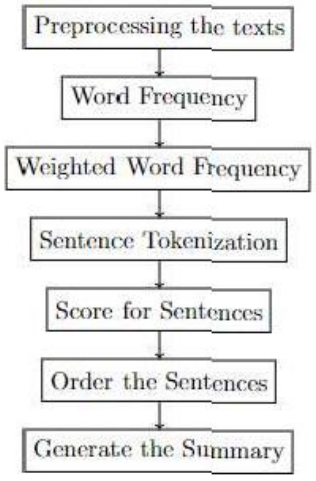

Fig. 1. Lemmatization Process

B. NLTK Approach And Working

One of the main advantages of using NLTK as a text summarizer is its flexibility and customizability. NLTK allows you to choose between different summarization techniques, such as extraction-based or abstraction-based summarization, and to adjust various parameters to achieve the desired level of summarization. The steps which can be followed to implement the approach is under the following ways :

1. Import the necessary modules and load the text you want to summarize.
2. Tokenize the text into individual sentences.
3. Remove stop words from the sentences to focus on important words.
4. Calculate the word frequency of each word in the filtered sentences.
5. Calculate the weighted frequency of each sentence based on the word frequencies.
6. Select the top N sentences with the highest score to include in the summary.
7. Combine the selected sentences into a summary string and return it.

C. spaCy Approach And Working

spaCy is like a superhero for natural language processing (NLP)! It's a powerful tool that helps computers understand human language better. Just like how superheroes have a unique set of skills, spaCy has a range of powerful features that can help you process text, extract useful information, and gain insights from data. Think of it this way: if you wanted to teach a computer to read and understand human language, you'd have to start by teaching it the building blocks of language, like grammar rules, sentence structure, and vocabulary. But that's a lot of work! Instead, you can use spaCy to do the heavy lifting for you. spaCy comes with pre-trained models that can automatically analyze text and extract information like named entities, parts of speech, and dependencies between words. It 19 can also be customized to work with specific languages, domains, or tasks. With spaCy, you can build applications that can understand and process text in a humanlike way, making it easier to automate tasks like chatbots, sentiment analysis, and information extraction. The steps to follow to implement spaCy starts with pre-training the model.

1. Import spaCy and load a pre-trained model.
2. Load the text you want to summarize into a Doc object.
3. Calculate the sentence scores based on important features such as word frequency, sentence length, entity recognition and position using a scoring function.
4. Calculate the score of each sentence in the Doc object
5. Sort the sentences in descending order of score and select the top N sentences to include in the summary.
6. Combine the selected sentences into a summary string and return it.

D. Sumy Approach And Working

Sumy is a Python package that can be installed through pip and provides a range of algorithms and techniques for text summarization. These techniques include Luhn's heuristic method, Latent Semantic Analysis, Edmundson's heuristic method that uses previous statistical research, unsupervised approaches such as LexRank and TextRank, Sum Basic which is a common baseline method, and KL-Sum which 20 selects sentences that decrease the KL Divergence when added to the summary.

1. Install Sumy library using pip command.
2. Import the Sumy module and the summarization method you want to use.
3. Read the input text you want to summarize and store it as a string.
4. Initialize the tokenizer and parser using the Tokenizer and PlaintextParser classes from Sumy.
5. Use the parser to create a Document object from the input text.
6. Initialize the summarizer using the desired summarization algorithm, such as LexRankSummarizer or LsaSummarizer.
7. Set the parameters for the summarizer, such as the number of sentences or the threshold for sentence similarity.
8. Generate the summary by calling the summarizer's summary() method on the Document object.
9. Convert the summary to a string and print it to the console.

IV. Result Analysis

A. Model Analysis

The frontend of the project underwent significant modifications to meet the specific requirements. As part of these modifications, two important features, Reading Time and Time Elapsed, were incorporated into the interface. These additions aimed to enhance user experience and provide valuable information. In terms of methodology, we employed distinct methods for comparison and implemented robust processes. Our successful implementation focused on the extractive approach for text summarization. To achieve this, we leveraged various techniques and algorithms. Firstly, we employed the NLTK-based approach, which involved utilizing the spaCy summarization algorithm and the sumy LexRank method. These approaches yielded promising results. Additionally, we integrated the OpenAI GPT-3 Summarizer to compare its performance with other algorithms. Through extensive research and analysis, we observed the significance and effectiveness of this cumulative extractive summarization process. The current interface allows users to input text and generate a summary accordingly. Furthermore, we introduced a URL to text extractor feature, which enables summary generation from webpage. This functionality involves sending a GET request to the provided URL to retrieve the HTML content.

In order to evaluate the quality of the generated summaries, we employed similarity metrics such as Cosine Similarity or Jaccard Similarity, which compare sets of words or vectors. Upon comparison, we found that the cosine similarity score for GPT-3 Summary was approximately 0.653, indicating a lower level of similarity compared to the sumy libraries' LexRank method, which scored 0.774. However, it is important to note that the NLTK and SpaCy approaches demonstrated superior performance, with the NLTK approach achieving a cosine similarity of 0.888 and SpaCy outperforming with a cosine similarity of 0.894. These results highlight the effectiveness and superiority of the NLTK and SpaCy approaches in generating high-quality summaries. The cumulative process employed in this extractive-based summarization implementation not only met the requirements but also showcased significant enhancements in the field.

A. User-Interface of the Summarizer

Homepage interface is shown in Fig. 2 and the compare page interface is shown in Fig. 3 and the output comparison in Fig. 4. User can ask query both in the form of text or website link.

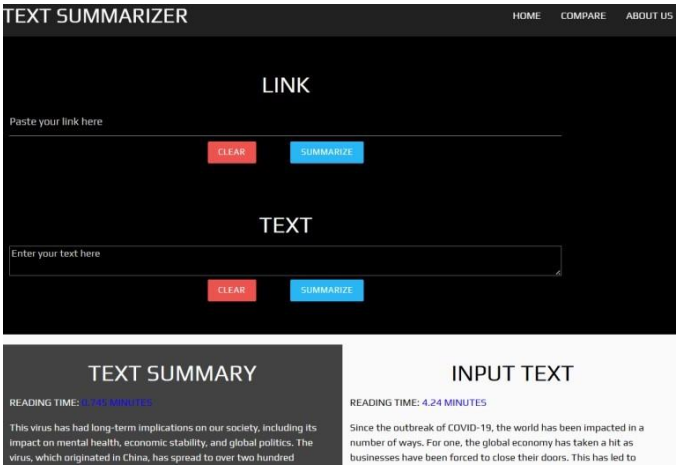

Fig. 2. Home Page Interface

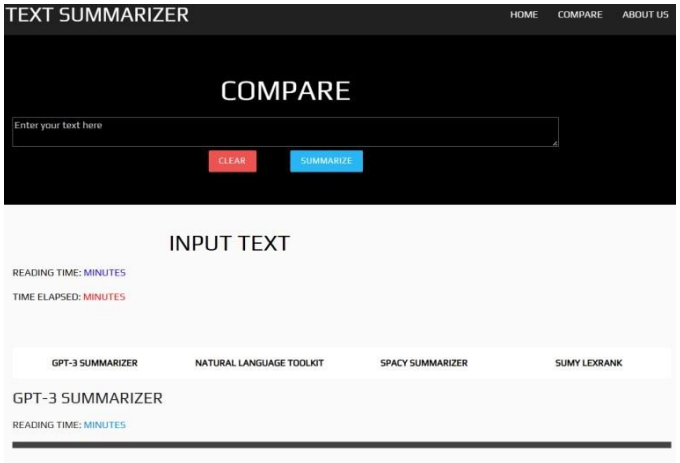

Fig. 3. Compare Page Interface

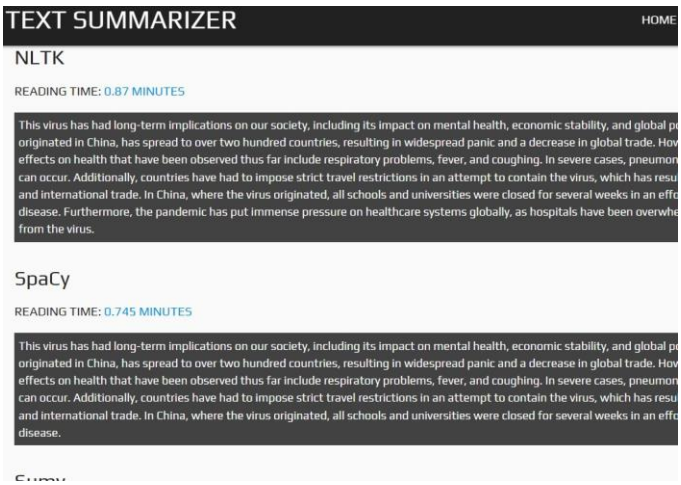

Fig. 6. NLTK, SpaCy and Sumy Summary

V. CONCLUSION AND FUTURE WORKS

The main objective of the summarizer is to create an application which will generate meaningful and compact summary for students and anybody else to get answers more swiftly. Also help in reducing the human work effort to resolve the read an entire document[16-19]. Future research should explore alternative approaches beyond graph-based methods to overcome limitations and improve the accuracy of summarization in this domain. Addressing the limitations related to accuracy and time, particularly with larger documents, is crucial for maximizing the effectiveness of the hybrid summarization technique. Achieving a balance between various aspects of summary quality remains a challenge that requires further research and optimization. The result demonstrates that it is capable of generating the required information in response to the user's question concerning Students, faculty, and anybody who want to generate the summary.

REFERENCES

[1] M. Rajesh, K. Vengatesan, and M. Gnanasekar, Recent Trends in intensive computing. Amsterdam, NY: IOS Press, 2021.

[2] T. Priyadharshan and S. Sumathipala, "Text summarization for Tamil online sports news using NLP," in 2018 3rd International Conference on Information Technology Research (ICITR), 2018, pp. 1–5.

[3] H. Li, J. Zhu, C. Ma, J. Zhang, and C. Zong, "Read, watch, listen, and summarize: Multi-modal summarization for asynchronous text, image, audio and video," IEEE Trans. Knowl. Data Eng., vol. 31, no. 5, pp. 996–1009, 2019.

[4] H. Dave and S. Jaswal, "Multiple Text Document Summarization System using hybrid Summarization

technique," in 2015 1st International Conference on Next Generation Computing Technologies (NGCT), 2015, pp. 804–808.

[5] R. Khan, School of Software, Xinjiang University, Urumqi 830008, China, Y. Qian, and S. Naeem, "Extractive based Text Summarization Using KMeans and TF-IDF," Int. J. Inf. Eng. Electron. Bus., vol. 11, no. 3, pp. 33–44, 2019.

[6] M. S. Patil #, M. S. Bewoor, and S. H. Patil#, "A hybrid approach for extractive document summarization using machine learning and clustering technique," Ijcsit.com. [Online].Available:https://www.ijcsit.com/docs/Volume%205/vol5issue02/ijcsit20140502140.pdf. [Accessed: 24-Mar-2023].

[7] S. Bhatia, "A Comparative Study of Opinion Summarization Techniques," IEEE Trans. Comput. Soc. Syst., vol. 8, no. 1, pp. 110–117, 2021.

[8] P. N. Varalakshmi K and J. S. Kallimani, "Survey on Extractive Text Summarization Methods with Multi-Document Datasets," in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 2113–2119.

[9] N. C. P. Prakash, A. P. Narasimhaiah, J. B. Nagaraj, P. K. Pareek, N. B. Maruthikumar, and R. I. Manjunath, "Implementation of NLP based automatic text summarization using spacy," Int. J. Health Sci. (IJHS), pp. 7508–7521, 2022.

[10] D. Gupta, "Autism Detection using r-fMRI: Subspace Approximation and CNN Based Approach," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 2, pp. 1029–1036, 2020.

[11] M. Aswani, "Automatic text summarization from unstructured text using natural language processing," Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 2, pp. 2265–2269, 2020.

[12] K. Kadriu and M. Obradovic, "Extractive approach for text summarization using graphs."

[13] S. Kadagadkai, M. Patil, A. Nagathan, A. Harish, and A. Mv, "Summarization tool for multimedia data," Global Transitions Proceedings, vol. 3, no. 1, pp. 2–7, 2022.

[14] K. Prudhvi, A. Bharath Chowdary, P. Subba Rami Reddy, and P. Lakshmi Prasanna, "Text summarization using natural language processing," in Advances in Intelligent Systems and Computing, Singapore: Springer Singapore, 2021, pp. 535–547.

[15] Y. Gao, C. M. Meyer, and I. Gurevych, "Preference-based interactive multi-document summarisation," Inf. Retr. Boston., vol. 23, no. 6, pp. 555–585, 2020.

[16] A. Bagalkotkar, A. Kandelwal, S. Pandey, and S. S. Kamath, "A Novel Technique for Efficient Text Document Summarization as a Service," in 2013 Third International Conference on Advances in Computing and Communications, 2013, pp. 50–53.

[17] N. Ibrahim Altmami and M. El Bachir Menai, "Automatic summarization of scientific articles: A survey," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 4, pp. 1011–1028, 2022.

[18] O. Ahuja, J. Xu, A. Gupta, K. Horecka, and G. Durrett, "ASPECTNEWS: Aspect-oriented summarization of news documents," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 6494–6506.

[19] M. Hirohata, Y. Shinnaka, K. Iwano, and S. Furui, "Sentence extraction-based presentation summarization techniques and evaluation metrics," in Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, 2006, vol. 1, p. I/1065-I/1068 Vol. 1.