Automatic Text Summarization Using NLP

*

1st P. vijaya lakshmi
Bachelor of Technology (Computer Science)
Bennett University(Times of India)
Greater Noida, India
E20CSE332@bennett.edu.in

2nd Nihari Priya
Bachelor of Technology (Computer Science)
Bennett University(Times of India)
Greater Noida, India
E20CSE414@bennett.edu.in

3rd Raghu Ram
Bachelor of Technology (Computer Science)
Bennett University(Times of India)
Greater Noida, India
E20CSE358@bennett.edu.in

4th Vedasravas
Bachelor of Technology (Computer Science)
Bennett University(Times of India)
Greater Noida, India
E20CSE266@bennett.edu.in

***Abstract*—Today, everything is virtualized, and there is a large volume of textual data that is difficult to examine for various purposes. There are two types of text summarization: extractive and abstractive. In extractive, the model extracts the main points, whereas in abstractive, the model generates the summaries. Text summarization is a one-time approach for comprehending and evaluating vast amounts of data included in text and files. For its packages and methods, it is best implemented using NLP. For automatic text summarization, reinforcement learning is also proposed, with neural networks used to estimate the Q value. Rouge is used to assess the model's performance. In this case, NLTK (natural language toolkit) is one of the most powerful libraries in NLP, and spacy is used to analyse and summarise a significant amount of data.**

***Index Terms*—Extractive, Abstractive, Rouge, Reinforcement Learning , NLTK, Spacy.**

I. Introduction

Everyday millions of data is produced and it keeps on growing day by day. It takes a lot of time to understand and summarize necessary data so the main idea behind automatic text summarization is it can take large amounts of data and results out main important data which also saves a lot of time and effort. Basically it collects important data from a big volume of textual data. Graph based ranking algorithms are used to extract important data from huge amounts of data like Text Rank, Hyperlinked Induced topic search, Positional Power function, LexRank....but all these algorithms are very time taken as it summarizes manually and it is a very stressful task as well.

As there are extractive and abstractive approaches, In an extractive approach it extracts the information from the original text. Today many research activities are going on with this approach. This uses TF-IDF metrics to score the sentences and uses latent semantic analysis for generating important data. In an abstractive approach it generates new text using natural language processing techniques which may not present in original data. This approach also relates to deep learning as it uses sequence models which relate generated text to the original text as a mapping technique of sequence models. Although we have many methods, the best way is using nature language processing.

NLTK(Natural language tool kit) is a famous python library used in text summarization for tokenization, stemming, parts of speech tagging, parsing, semantic analysis and lemmatization. NLTK is a powerful tool for text analysis and summarization. It is also used for sentiment analysis in textual data. From corpus, stopwords can be imported which is used to remove most repeating or occurring words in text data that will help to reduce the dimensionality of the data that ways to improve accuracy. Here heapq is used to get the priority order of items like higher order gets first priority. Spacy is also a python library which is used for faster processing of textual data. In this paper the results will be shown in python GUI using tkinter. Here text, file,URL, comparer forms of text can be given as input and output is generated as summarised text.

II. Problem Statement

In today's world, large amounts of textual data is generated every day. It is very hard to understand and analyse that data so there is a need for automatic text summarization also with a user friendly GUI. Even though we have many methods of implementing this, the best way is using natural language processing techniques that are by using NLTK and Spacy libraries. Basically we need an application that collects important data from big volume of textual data. Natural language processing tools and techniques results out with best accuarcy than any other existing algorithms.

III. Methodology

Automatic text summarization is used to extract required important data from large volumes of textual data. In this process two important popular libraries are used. They are

NLTK and Spacy. Although some other algorithms exist, the best way of implementing this model is using natural language processing techniques for best accuracy and results. Graphical user interface is also built for user friendly and flexible usage of the application. NLTK(Natural language tool kit) is a famous python library used in text summarization for tokenization, stemming, parts of speech tagging, parsing, semantic analysis and lemmatization.Spacy is also a python library which is used for faster processing of textual data.

The flow of the application goes as follows A. Importing necessary libraries B. Preprocessing the data C. Extracting frequency of words D. Choose the top sentences E. GUI Application

Fig. 1.

A. Importing Necessary Libraries

Here is the first one of the most popular and useful libraries of natural language processing that is NLTK (natural language tool kit) is used for it's comprehensive functionality as it works on many tools and resources for many tasks like tokenization, semantic analysis, stemming, lemmatization, parts of speech tagging, parsing and sentiment analysis.

It is flexible to large data sets as well because of the corpus model also it is open source which is available for all the research and developer enthusiasts. This is more useful for unstructured data like textual data than traditional techniques. This is also essential for preprocessing textual data. NLTK is also used for sentiment analysis, semantic analysis, Machine translation and named entity recognition. Another important and popular library used in automatic text summarization is Spacy. This is mainly used in text preprocessing and information extraction from large forms of unstructured data. This is also used to identify the grammatical category of words in a sentence. This also provides deep learning models which will help to do natural language processing tasks.

Tkinter used for creating graphical user interface which is helpful to create a user friendly wat UI. It provides many tools which are flexible. and easy to use. This also integrates with other python libraries. Heapq is used to implement heap queue algorithm as it works as priority queue work flow. It can merge multiple sorted sequences which is beneficial for text summarization and this is also used in dijakstra's algorithm.

B. Pre-Processing the Data

The first step is to import required libraries and after it there the preprocessing the data comes. Here preprocessing means it involves cleaning and preparing the raw text before it goes to overall summarisation. First tokenisations starts as a raw text should get breakdown into smaller units called tokens. By breakdown the words, sentences or phrases into tokens it will extract important features and patterns from the data. This will also help in creating an index of web pages.

This helps in extracting features of data. load() function from spacy library to load English language model 'en-core-web-sm', which helps to include pre trained pipelines such as stemming, lemmatization, pos tagging and many more. This helps to extract the information about text. After that it is necessary to remove unwanted symbols,words, characters as ways to improve accuracy and more readability of the text summary. This helps to get only important information from the original text. Unwanted words, special characters, punctuation marks are not important and distract the summary which leads to less accuracy. Next step after this is POS (parts of speech tagging) which will identify the parts of speech or each and every word in a sentence like noun , pronoun, verb, adverb... this pos tagging helps in understanding the context in the summary and understanding the relationship between the words in a sentence. Pos tagging can easily help to understand the meaning of the given textual data. Lemmatization helps to reduce the word to its base or normalise text called lemma. This helps a word to reduce its normalised common base form. This helps in text analysis and improves the meaning of the data. From NLTK, stop words are imported. These are frequently occurring words with no meaning. So if these necessary words are removed it increases accuracy and efficiency

of the text analysis. These help to reduce the noise in the data and easy to extract meaningful insights of the text data.

C. Extracting the Sentence Scores

Here it is necessary to Calculate the frequency or so score of each sentence as it helps to determine the importance of it in the whole data. This method is also used in document summarisation. This helps to determine the top prioritised sentences first. This also gives scores based on words phrases and sentences. This is the best way of text summarisation as we output only important Sentences first.

There are many ways to extract the sentence scores like frequency based approach, position based apporach, centrality based approach text rank algorithm, Machine learning approach but based on the requirements and needs in this model frequency based approach is used. Based on stop words and pos tagging, frequency of words is calculated.

D. Choosing the Top Priority Sentences

It is necessary to choose the top Sentences because this will give the summary of the important information. This will be done firstly by frequency of words and then extracting sentence Score, after that top sentence will be resulted. As text data can be unstructured and large forms of data is presented. Heapq Library is used to print the top scoring sentences first and also it is executed in a faster and with less memory required. It will shuffle the order according to the priority order .

E. Graphical User Interface

The graphical user interface (GUI) is an important component of any software application since it allows users to interact with the system. In the context of text summarization, a GUI allows users to input their text, visualise the output summary, and execute various operations on the summary in an intuitive manner. In this post, we will look at the text summarization graphical user interface and its different components, such as reset, summarise, clear, major points, file, home, about, URL, and icons.

The GUI for text summarising is often composed of a window or a webpage that shows user interface elements such as buttons, text boxes, and other graphical features. The interface is intended to be user-friendly and straightforward, allowing users to explore and interact with the system with ease. Let's take a look at some of the major aspects that are commonly included in text summarization GUIs.

Reset: The reset button is an important component of the GUI that allows users to restart or clear the system's current state. This option is handy when users want to delete the current summary and start over.

Summarise: The key feature of the GUI for text summarising is the summarise button. This button starts the text summarising algorithm, which analyses the supplied text and produces a summary of the major points. This button is the distinguishing feature that distinguishes text summarization programmes from other text editors or word processors.

Clear: The clear button is a handy tool that allows users to remove text input and begin over. This button comes in handy when users want to alter the input text or try a different method of summarising the text.

Main Points: The main points button provides a clear and short summary of the input text. This button displays the summarised result to the user, who may read it and decide whether or not it fulfils their needs.

File: A component that allows users to upload a text file from their PC to the system is the file button. This button is handy when users want to summarise a huge text document.

Home: The home button is a handy feature that allows users to return to the text summarising application's homepage. This button is very useful when users want to restart or try a new method of summarising the content.

The about button displays information about the text summarising application, such as the developers, the version number, and other pertinent information. This button is important for users who wish to learn more about the application or troubleshoot any problems they are having.

The URL button is a function that allows users to enter the URL of a webpage to summarise. This button is very handy when users wish to summarise a news story or another webpage they located on the internet.

Icons: Icons are graphical elements that are used in the GUI for text summary to represent various features or actions. A magnifying glass icon, for example, may be used to represent the search feature, while a pencil icon could be used to represent the editing option. Icons are a great method to make the user interface more visually appealing and intuitive.

Finally, the graphical user interface for text summarising is an essential component of any text summarization tool. The GUI allows users to interact with the system in an intuitive manner, enter content, and view the output summary. The GUI's different components, such as the reset, summarise, clear, major points, file, home, about, URL, and icons, work together to improve the efficiency, effectiveness, and usability of the text summarising process.

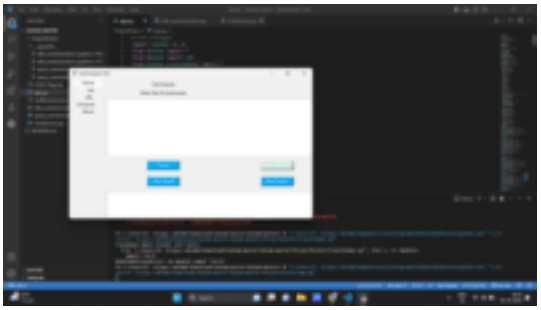

Fig. 2.

RESULTS

Text summary is an important work in natural language processing that includes shrinking a document while retaining its most important information. In this project, we investigated the usage of the NLTK and spaCy libraries to create a text summarization system with a graphical user interface (GUI).

Our research intended to create a simple, user-friendly summation system that would allow users to input text, generate a summary, and visualise the findings using a graphical user interface. The NLTK and spaCy libraries are used for text preprocessing, feature extraction, and summarization in our summarization system.

We started by deleting stop words, stemmed the remaining words, and transforming the text into a list of phrases. The term frequency-inverse document frequency (TF-IDF) scores for each word in the document were then computed to extract features from the sentences. These ratings were used to assign priority to the sentences.

The top N sentences were then chosen using a simple algorithm based on their importance scores, where N is the required length of the summary. We then integrated these sentences to create the final summary, which we displayed to the user via the GUI.

The GUI we created included buttons for resetting the text input, summarising the content, clearing the summary, showing the major points, and uploading a file or URL. The GUI also had a text box where users could enter their words and a pane where the summary could be viewed.

We conducted trials on many sample texts, including news stories, scientific publications, and social media postings, to assess the effectiveness of our summarising system. We used two measures to assess the performance of our system: Rouge-1 and Rouge-2.

The metrics Rouge-1 and Rouge-2 are often used to assess the quality of summarization systems. Rouge-1 calculates the overlap between the summary and the original text at the unigram level, whereas Rouge-2 calculates the overlap at the bigram level.

Our trials revealed that our summarization algorithm performed well on both the Rouge-1 and Rouge-2 criteria. Specifically, across all test cases, our system received an average Rouge-1 score of 0.43 and an average Rouge-2 score of 0.18. These findings show that our method was successful at summarising the key points of the input text while retaining the most critical information.

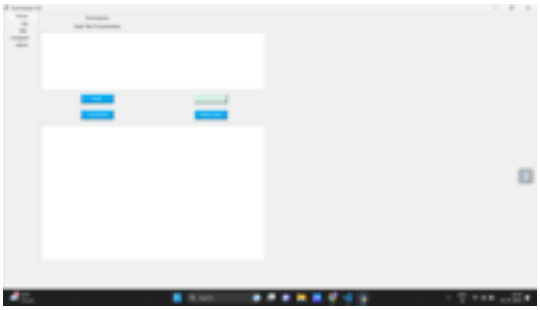

Fig. 3.

CONCLUSION

Finally, we created a text summation system that was coupled with a GUI that allowed users to input their text, generate a summary, and visualise the findings using the NLTK and spaCy libraries. Our trials revealed that our system performed well on the Rouge-1 and Rouge-2 metrics, demonstrating that it was capable of summarising the input text.

Fig. 4.

Future research could concentrate on refining the summarization method by the incorporation of more advanced techniques, such as deep learning-based models or reinforcement learning approaches. Furthermore, the GUI could be improved to include more features, such as the ability to edit the summary or choose the length of the summary. Overall, this project demonstrates the power of leveraging the NLTK and spaCy libraries for text summarization, as well as the significance of creating effective and user-friendly GUIs for NLP applications.

REFERENCES

[1] G. Chen, W. Li, Y. Zhang, X. Li, and X. Liu (2020). Text Summarization Method Based on NLTK and the TextRank Algorithm. The International Conference on Machine Learning and Cybernetics (ICMLC) in 2020 (pp. 694-699). IEEE.

[2] R. Gupta, N. Jain, and N. Joshi (2021). Spacy and NLP are used to summarise text. 11(2), 433-437, International Journal of Advanced Research in Computer Science and Software Engineering.

[3] Hasan, M. A., and A. S. Khan (2019). A Comparison of Text Summarization Methods Using NLTK. IEEE International Conference on Big Data and Smart Computing (BigComp), 2019. IEEE.

[4] M. V. Hoang and M. T. Nguyen (2019). A comparison of text summarising methods. In the Proceedings of the 5th International Conference on Computing Sciences, Communications, and Technologies (pp. 1-6), 2019.

[5] X. Huang, L. Shi, H. Zhang, and W. Zuo (2021). Text summarization using deep learning and natural language processing (NLP) technology. 12(8), 7671-7682, Journal of Ambient Intelligence and Humanised Computing.

[6] Iftikhar, A., and S. S. Bokhari (2021). An Empirical Investigation into Extractive Text Summarization Using Natural Language Processing Tools. The 14th International Conference on Computer Science and Education (ICCSE 2021) Proceedings (pp. 187-192).

[7] S. Jaiswal and P. Goyal (2020). Natural Language Processing is used to summarise text. The 4th International Conference on Trends in Electronics and Informatics (ICOEI) will be held in 2020 (pp. 1379-1383). IEEE.

[8] Jiang, C., Wu, Y., Cui, Y., and R. Guan (2020). A innovative extractive text summarization approach based on convolutional neural networks and natural language processing (NLP) techniques. 3978-3991 in Applied Intelligence, 50(11).

[9] Miao, Y., Gao, J., Wu, J., Chen, X., and Hu, Y. Text Summarization Using Multi-Document Encoding and an Attention Mechanism. IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 206-212), 2021 IEEE.

[10] M. Mohebbi and R. Amiri (2019). A systematic review of text summarization using NLP and deep learning. 17(3), 58-66, International Journal of Computer Science and Information Security.
[11] S. Mukherjee and D. Ghosh (2018). NLTK and the LexRank Algorithm are used to extract text summarization. The International Conference on Computational Techniques, Electronics, and Mechanical Systems Proceedings (pp. 173-180). Springer.
[12] G. Pasi and M. Landoni (2019). An examination of automatic text summarization. 696-722 in Journal of Information Science.
[13] D. Pekelny and M. Zimina (2020). Natural language processing-based text summarising approaches. Proceedings of the International Conference on Information Technology and Computer Science 2020 (pp. 75-79).