Grenze International Journal of Engineering and Technology, June Issue

Text Summerization using Natural Language Processing

Manasaveerashyva Y N[1] and Prathibha B S[2]
[1-2]The National Institute of Engineering, Department of Information Science & Engineering, Mysuru, India
Email: manasa19962604@gmail.com, prathibha@nie.ac.in

***Abstract*—Although there are several chunks of information available on the internet, it is critical to give a method for obtaining information in the most effective and correct manner possible. The most common challenge in the present period is text summarization.**
The primary purpose of Text Summarization is to extract summaries from vast amounts of text in an efficient and reliable manner. Text summary can help you save time by reducing the amount of time you spend reading. Text summarising approaches include extractive and abstractive summarization. The proposed system for text summarization and keyword extraction goes through a series of steps, starting with data extraction from a website link, removing outliers and irrelevant information, emphasising the importance of particular data extracted from the website, and creating a summary of the extracted data. The extraction of essential information from the retrieved data necessitates the use of natural language processing. By giving a summary derived from multiple internet connections and papers, as well as keywords from a specific website or document, the proposed solution supports users in decreasing their surfing time.

***Index Terms*— NLP, Text Summarization, News Summarization.**

I. INTRODUCTION

The process of compressing vast volumes of text into a smaller quantity of text without changing the meaning is known as text summarising. Text summarization is in high demand in to day's world. The primary advantage of Text Summarization is that it reduces the amount of time a user spends reading.
There are two types of text summarising: extractive and abstractive text summarization. People have become overwhelmed by the vast amount of information and documents available on the internet as the internet and big data have grown in popularity.
As a result, several academics are being urged to develop technologies that can automatically summarise texts. Summarization is the process of condensing a long piece of text into a shorter version that retains significant informative features and content meaning while lowering the size of the original text. Because manual text summarization is a time consuming and exhausting activity, automating it is becoming more popular, and so serves as a powerful stimulus for academic study.
The goal of automatic text summarising is to keep important information while reducing the length of documents or reports. [5] To acquire a brief overview of the results, text summarization can be utilised in a variety of contexts, including news stories, emails, research papers, and online search engines. Condensing a large text into a smaller, more accurate, and fluid group of phrases that is easy to understand and gives the reader with the information they need in a minimal number of words is known as text summary. [1] Extractive and abstractive summarising are the two primary approaches to text summarization. The summary is created using extractive

Grenze ID: 01.GIJET.8.2.15

based summarising, which extracts significant phrases and sentences from the original text. When identifying the most essential sentences from the text, the statistical features of the phrases are taken into account. Abstractive summarization extracts the main points from the original text and summarises them in plain English. Here, the grammatical properties of the sentences are considered.
Only a few of the currently available text summary apps generate abstractive summaries, with the bulk being extractive.
For constructing multi-document summaries, graph based algorithms for abstractive text summarising have proven to be superior than previous methods. Knowledge graphs provide a summary that is more consistent with human reading patterns and logic. As a result, we suggest a possible solution to this problem that entails extracting only the substance from news websites in the form of an automatically created summary, allowing newsreaders to make better selections in less time. To construct summaries, machine learning approaches rely on machine learning algorithms. Summarization is treated as a classification problem in machine learning systems, with sentences being classed as summary or non-summary depending on their attributes.

A. Motivation

More and more information is available these days thanks to the internet and other sources. We need a tool to extract the right collection of sentences from the given documents in order to manage this data more efficiently. When working with a big number of papers, summarising the text is necessary to gather the most significant information. Information has been an integral part of our lives since the introduction of the World Wide Web. The human mind is incapable of remembering every detail of every piece of knowledge .As a result, text document summarising is essential for data collecting. The NLP Algorithm is used to do the summarization task in this work, and the system also performs language translation in addition to text summarization.

II. LITERATURE SURVEY

Massimo Mauro et al. created EXT summaries using a sentence extraction algorithm. The sentences are examined for relevancy and rated accordingly using this procedure. After that, similar sentences were clustered together to get the most informative sentences, which were chosen based on sentence ratings.[2]
To develop EXT summaries of articles, Sarda A.T. et al. recommended using NNs and Rhetorical Structure Theory (RST).
To create an appropriate statement for the summary, features are compared and blended. The initial stage is to teach the NN how to choose the types of sentences that should be included in the summary[3].
WEKA's selection techniques, CFS Subset Evaluator, Information Gain Evaluator, and SVM Attribute, were used to minimise the dimensionality of feature vectors. Among five classifier s tested on WEKA's platform, Naive Bayes was shown to be the best[4].
Taeho Jo has presented a method for summarising text that takes into account attribute similarity and use the KNN algorithm . As input, a paragraph is given, which is then broken down in to sentences. Vectors are used to represent words. Each senten ce is then classed as summary or remaining, and it is included in the summary based on the similarity score between it and a human-generated summary[5].
A query based strategy for EXT text summerization was presented by Mahsa Afsharizadeh et al. It extracts the most important sentences from the document and adds them to the summary. To locate the key sentences, eleven query dependent appropriate features were identified and employed i n the paper[6].
The DUC 2007 corpus was used for training and testing.
This strategy outperformed earlier methods in terms of average precision, average recall, and average Fscore in the paper[7].
EXT text summaries were generated using Nave Bayes, Random Forest, and SVD. On Twitter, Kamalanathan Kandasamy and colleagues employed machine learning to classify spam.
SVM was also used to test the data sets, however Nave Bayes was determined to be more accurate. A total of 100 people were tested in this trial. 60 of them were genuine, while the other 40 were spam. A total of 98 users were accurately classified [9].
Many studies have also used deep learning approaches for EX T and ABS text summarization. Deep learning is a branch of machine learning. NN approaches have been applied in a variety of ways. Text summaries have also been generated using reinforcement learning, Convolutional NN(CNN), and RNN[10].
To produce key phrases, Nicholas Giamblanco et al presented a Newtonian technique. For keyword and key phrase extraction, the author employed four steps. The stop words are deleted first, which is known as noise

filtering, after which each word is assigned a mass, the relationships between words are calculated, also known as word attraction, and lastly a key phrase is generated[11].
The document is broken down into sentences and examined to see if it's related or not to the main theme. The sentence is the n evaluated with a NN to see if it should be included or not. For this model, 100 CNN news articles were employed as data sets, along with their related ABS summaries. Sentences from the news story and the summary are examined, and the ones with the highest similarity score are picked for the EXT summary[13].

III. METHODOLOGY

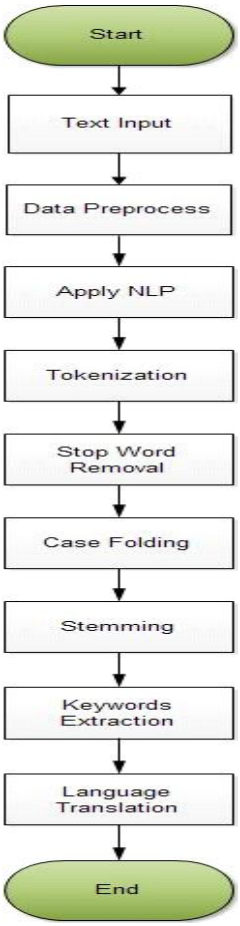

Fig 3.1: Proposed System Flowchart

A. Preprocessing

To make the content lighter, several linguist-developed approaches are used to preprocess the text document for structuring. There are numerous techniques for reducing the density of a written document. The following techniques are being used in this research.

Stemming

Stemming is the process of reverting a word to its root or base form, such as using the singular form of a word rather than the plural (boys as boy) or deleting the *ing* from a verb (changing doing to do)[8]. Stemming can be accomplished using a variety of algorithms, which are referred to as stemmers.

Part of Speech Tagging

The process of tagging or classifying text words according to the part of speech category they belong to (nouns, verbs, adverbs, adjectives) is known as part of speech tagging.

Stop Word Filtering

Words that are taken out before or after the preprocessing activity are known as stop words. There is no universally accepted standard for what classifies a word as a stop word; it is purely subjective and situation dependent[2]. Stop words in our circumstance include a, an, and , in, byand this term is eliminated from the original copy.

B. Keyword Extraction

After the density has been lowered, the document is sorted into a matrix. A phrase matrix S of order n*v contains the features for each sentence in a matrix[8]. For very informative summarization, we extract four qualities of a sentence from a text document: similarity with title, relative position of phrase, term weight of keywords generating phrases, and concept extraction of sentence.

C. Summary Generation

During the summary generation phase, the obtained optimal keyword set is used to generate the document's extractive summary. Collect a phrase score for each phrase in the document as the first stage in constructing a summary. The sentence score is calculated by intersecting the user question with the sentence[1]. The sentence is then ranked, and a final set of phrases for text summary synthesis is obtained, defining the summary.

D. Language Translation

The system's final phase is translating the summary text into several languages. The user will have the choice to select their preferred language. The technology can convert data from English to 8 different languages and vice versa.

IV. Results and Discussion

The framework has built up a programmed text summarization framework utilizing characteristic language handling system for example semantic anchoring in blend with highlight extraction utilizing fluffy rationale and treatment of relationship of sentences. Summarization utilizes the instinct behind the PageRank calculation to rank sentences Summarization is an unaided strategy for figuring the extractive rundown of a text. No preparation is essential. Given that it's language autonomous, can be utilized on any language, not just English. Right now, utilized as the comparability measure, the cosine likeness. We convert sentences to vectors to register this similarity.

Following snapshots shows the result of proposed work:

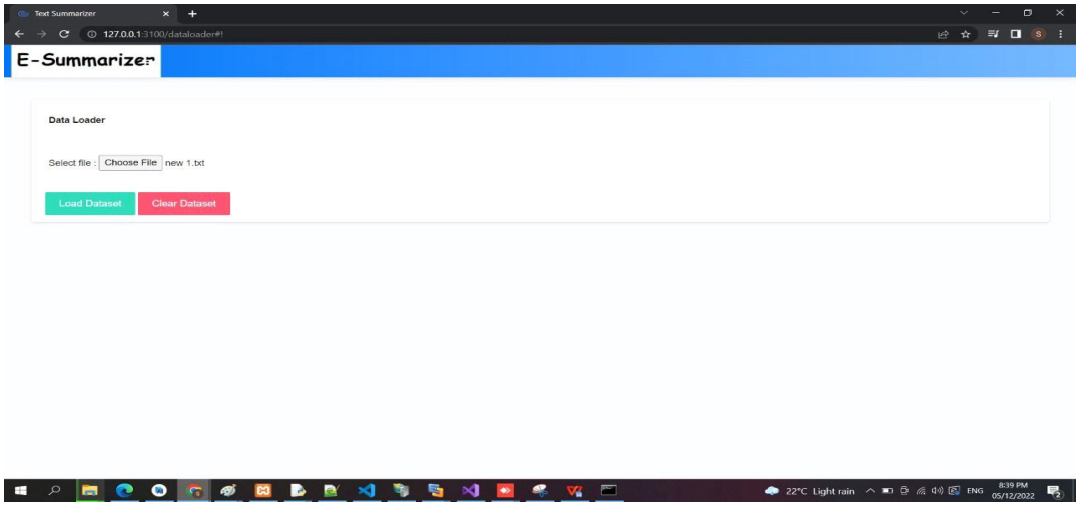

Fig: 4.1: E-Summerisation Preprocessing

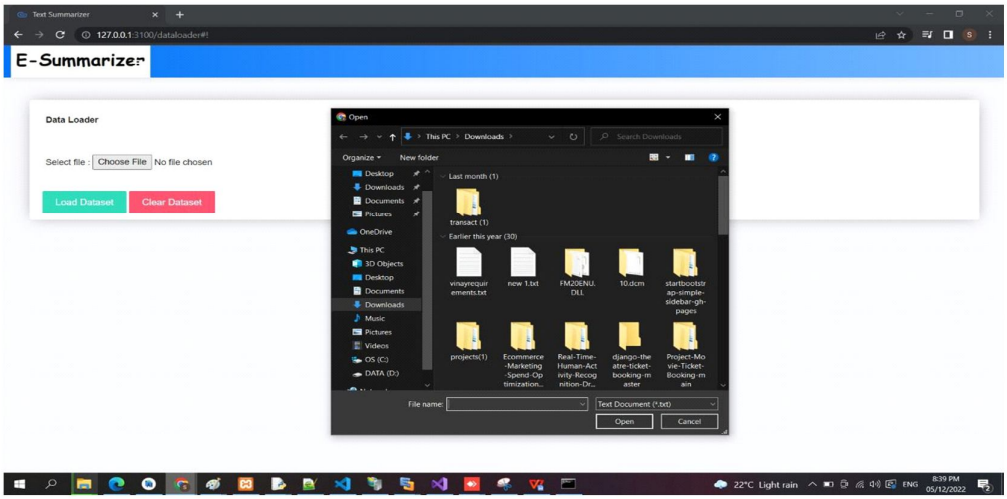

Fig: 4.2: Keyword extraction - Data extraction from a website link

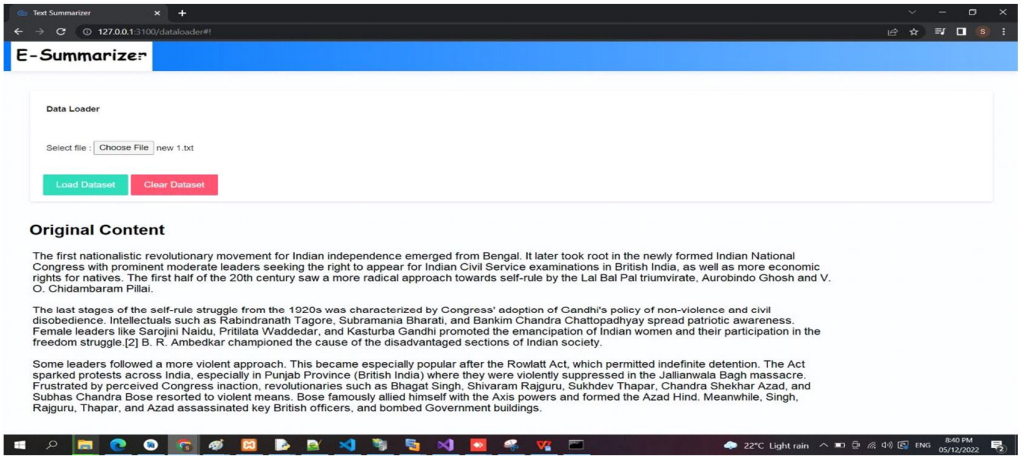

Fig: 4.3 Original content - Input in a paragraph

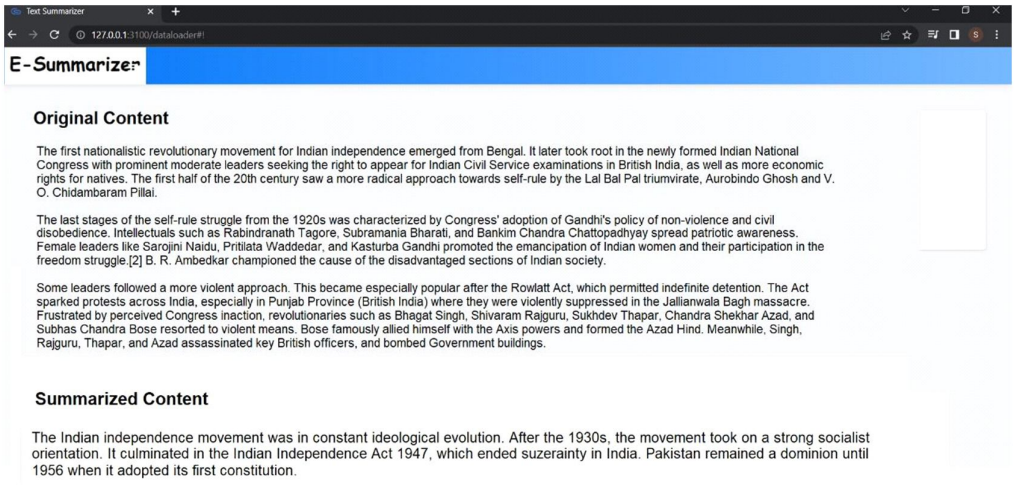

Fig: 4.4 Summary generation - output in a summarised content

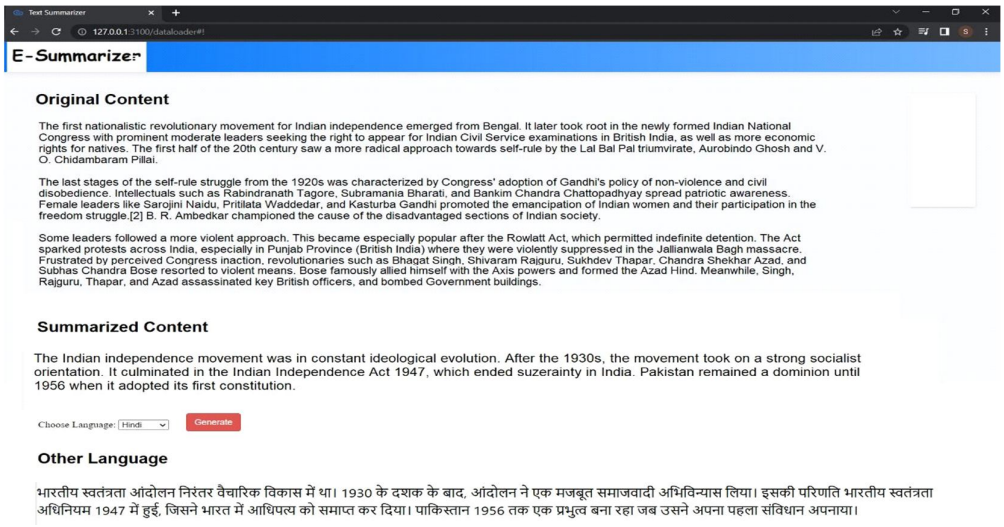

Fig: 4.5 Language translation - Summarised content can be translated into 8 different languages

V. CONCLUSION

This paper mainly focused on creating a system that gets concise summaries of articles or large chunks of text or any blog past. These implications would mean that knowledge gathering would be easier and time saving. This project work will reduce reading time and provides concise summary. On implementing Unsupervised Text Rank Algorithm Efficient results will be takes place. Access time for Information searching will be improved. Converting the Summarized text at various Real Time Scenarios. In this paper implementation of GTTS API has been shown to perform summarization into different languages.

REFERENCES

[1] M. Mauro, L. Canini, S. Benini, N. Adami, A. Signoroni, and R. Leonardi, “A freeWeb API for single and multidocument summarization,” ACM Int. Conf. Proceeding Ser., vol. Part F1301, 2017, doi: 10.1145/3095713.3095738.
[2] A. T. Sarda and M. Kulkarni, “Text Summarization using Neural Networks and Rhetorical Structure Theory” Int. J. Adv. Res. Comput. Commun. Eng., vol. 4, no. 6, pp. 49–52, 2015, doi: 10.17148/IJARCCE.2015.4612.
[3] G. Silva, R. Ferreira, S. J. Simske, L. Rafael Lins, M. Riss, and H. O. Cabral, “Automatic text document summarization based on machine learning,” DocEng 2015 - Proc. 2015 ACM Symp. Doc. Eng.,pp.191–194, 2015, doi: 10.1145/2682571.2797099.
[4] T. Jo, “K nearest neighbor for text summarization using feature similarity,” Proc. - 2017 Int. Conf. Commun. Control. Comput. Electron. Eng. ICCCCEE 2017, pp. 1–5, 2017, doi: 10.1109/ICCCCEE.2017.7866705.
[5] B. Mutlu, E. A. Sezer, and M. A. Akcayol, “Multidocument extractive text summarization: A comparative assessment on features,” Knowledge-Based Syst., vol. 183, p. 104848, 2019, doi: 10.1016/j.knosys.2019.07.019.
[6] M. Afsharizadeh, H. Ebrahimpour-Komleh, and A. Bagheri, “Query-oriented text summarization using sentence extraction technique,” 2018 4th Int. Conf. Web Res. ICWR 2018, pp. 128–132, 2018, doi: 10.1109/ICWR.2018.8387248.
[7] J. N. Madhuri and R. Ganesh Kumar, “Extractive Text Summarization Using Sentence Ranking” 2019 Int. Conf. Data Sci. Commun. IconDSC 2019, pp. 1–3, 2019, doi: 10.1109/IconDSC.2019.8817040.
[8] K. Kandasamy and P. Koroth, “An integrated approach to spam classification on Twitter using URL analysis, natural language processing and machine learning techniques,” 2014 IEEE Students’ Conf. Electr. Electron. Comput. Sci. SCEECS 2014, pp. 1–5, 2014, doi: 10.1109/SCEECS.2014.6804508.
[9] M. A. Fattah, “A hybrid machine learning model for multidocument summarization,” Appl. Intell., vol. 40, no. 4, pp. 592–600, 2014, doi: 10.1007/s10489-013-0490-0.
[10] N. Giamblanco and P. Siddavaatam, “Keyword and Keyphrase Extraction using Newton’s Law of Universal Gravitation,” Can. Conf. Electr. Comput. Eng., pp. 1–4, 2017, doi: 10.1109/CCECE.2017.7946724.
[11] R. Alguliyev, R. Aliguliyev, and N. Isazade, “A sentence selection model and HLO algorithm for extractive text summarization,” Appl. Inf. Commun. Technol. AICT 2016 - Conf. Proc., pp. 1–4, 2017,doi: 10.1109/ICAICT.2016.7991686.

[12] M. Moradi, G. Dorffner, and M. Samwald, “Deep contextualized embeddings for quantifying the informative content in biomedical text summarization,” Comput. Methods Programs Biomed., vol. 184, p. 105117, 2020, doi: 10.1016/j.cmpb.2019.105117.
[13] A. Jain, D. Bhatia, and M. K. Thakur, “Extractive Text Summarization Using Word Vector Embedding,” Proc. - 2017 Int. Conf. Mach. Learn. Data Sci. MLDS 2017, vol. 2018Janua, pp. 51– 55, 2018, doi: 10.1109/MLDS.2017.12.