ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 6, June 2020

Multilingual Automatic Extractive Text Summarizer

Sundarraj.R[1], Jacinth.G[2], Samuel Johnson.J[3], Elangamani.L[4], Anitha.A[5]

[1,2,3,4](Dept. of IT, UG Scholar, Francis Xavier Engineering College, Tirunelveli, India)

[5](Professor, Dept of IT, Francis Xavier Engineering College, Tirunelveli, India)

Abstract: This project "Multilingual Automatic Extractive Text Summarizer" deals with the Natural Language Processing (NLP) which summarizes the large document into a shorter version of the text. This automatic extractive text summarization is done by using Extraction based text summarization which involves selecting sentences of high relevance (rank) from the document based on word and sentence features and put them together to generate a summary. It summarize the text faster and effectively as this algorithm uses Graph-based sentences ranking. In this new era, where remarkable information is existing on the Internet, it is most important to provide an improved mechanism to extract the information quickly and most efficiently. There is plenty of text material available on the Internet. So there is a setback of searching for significant documents from the number of documents available and absorbing appropriate information from it. To solve the above two problems, the text summarization is very much necessary. This summarizer can reduce the paperwork and time, as the summarized text is converted into audio format and even share the summary as they wish by just one tap for almost every languages. The final performance of the application is validated using ROUGE-N value which gives precision value of 0.604 which is better than any other extraction based text summarization.

Keywords: Natural language processing, Automatic text summarizer, extraction based summarization, sentence ranking, Multilingual summarization.

I. INTRODUCTION

A Multilingual Automatic extractive text summarizer is the GUI based text summarization that uses the Text Rank algorithm to summarize the massive texts into a straight meaningful summary for multiple languages. The significance of Natural Language Processing (NLP) is ruling the AI industry for making humans to perform complex tasks with ease. Natural Language Processing is the process of building machines to understand, to read and to derive the meaning of the human language and making responses. As humans should not waste much time in reading the entire large text to recognize the subject of the larger document, just by one tap they could save their valuable time. In today's world, the Internet is the platform where all the study materials, news articles, even textbooks, etc., are searched. But the problem arises when grasping relevant information from the massive materials on the internet. So this application will provide the best-consolidated results will be given as a summary and extracts keywords for making humans understand the shortest version of the articles or text that describes the subject. Text summarization is the process of creating a short, precise and fluent summary of a longer text document. Automatic text summarization methods are very much needed to deal with the huge amount of text data existing online to get through relevant information quicker. This summarizer automatically summarizes the larger texts of any language and converted into audio for reducing the time for reading the subject.

II. SYSTEM STUDY

A. Existing System

The current summarizers automatically summarize the given documents based on any of the extractive methods. It uses the Spacy, Gensim, packages in python for summarizing the paragraphs. Text Summarization has two approaches namely Extractive Summarization and 2. Abstractive Summarization. The former extracts the words, phrases and sentences from the original text having high scorings based on algorithms. An abstractive based summarization produce a human-like summary, in which h the meaning will be extracted and sentences will be reframed according to the algorithms. The extractive method was used

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 6, June 2020

since it can perform faster than abstractive methods and also less complex when compared to abstractive methods as it uses statistical approaches. The current system works only for single language.

B. Proposed System

In this project we use an extractive method to summarize the huge texts automatically. Extractive summarization means extracting key sections of the passage and generating them a subset of the sentences from the original passage. When compared to automatic abstractive summaries, the Extractive method generates better results. This is because of the actuality that abstractive summarization methods deal with troubles such as semantic representation, inference and also natural language generation which is comparatively harder than data-driven approaches such as sentence extraction. To provide an effective summary of the large document automatically, to reduce reading time and paperwork, to select relevant informative subjects precisely in the research process, to generate summary automatically and faster than human understandings, to extract the keywords of the large passage for making summarization even more relevant to the subject, and also to share the summarized study-material of relevant topics in textbooks via QR code scanning of the summarized text to make it useful for others.

III. METHODOLOGY

The system is designed as a desktop application in which the user can get the required summaries by just interacting with the GUI .The full code is written in Python programming language as it reduces the LOC and even performs faster for AI related codes. And also python has numerous Library functions which would help programmer to code without effort paragraphs must be indented.

A. GUI

The GUI is built using Python Tkinter Package which helps to call in-built function to create Windows, Tabs, Scrolled Text, etc.,.In this project the entire GUI is build using python programming language. This desktop application has been designed of four tabs. The Window and tabs are designed using python as we can see in Fig.1.

Fig.1 GUI

B. Text Preprocessing

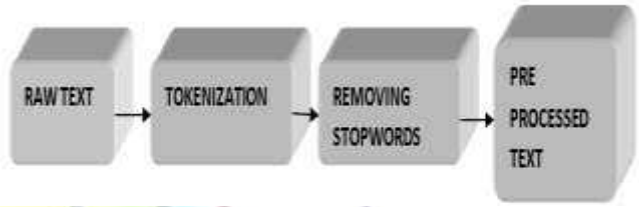

Fig.2 Text Preprocessing

From the Fig.2, the text should be tokenized that is to separate the words and even sentences and storing it in the dictionary. And then the stop words, that is the words which has no meaning when stand-alone, for example a, the, is, was, so these kind of words are called as stop words which should be removed as it will be huge in count in the input text, it will be the disturbance while scoring sentences. Using nltk.corpus.stopwords package, stop words can be neglected.

C. Summarization

This Application uses two summarization techniques namely, Text Rank algorithm which is similar to page rank algorithm and other one is Word frequency algorithm. Let us discuss these two methods of summarization in depth

1) WORD FREQUENCY ALGORITHM:

The Word frequency algorithm is the simplest way of finding the sentence score and ranking them to obtain a meaningful summary. The implementation is in Fig 3.

STEP 1: The input text should be preprocessed and ready for creating a frequency table. The Frequency

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 6, June 2020

table is the list of the count of each word in the pre processed texts.

STEP 2: From the frequency weight for individual tokens that is each word should be calculated by dividing its frequency by the frequency of the most occurring word.

$$W = \frac{\text{FrequencyOfTerm}}{\text{FrequencyOfMostOccuringWord}}$$

STEP 3: The sentence scores are calculated by the weights of words for each sentence.

STEP 4: Taking average on weights of all sentences and consider it as a threshold value.

STEP 5: The sentences which has weight value greater than or equal to the threshold value is taken as a summary.

Fig.3 Word frequency algorithm implementation

2) *TEXT RANK ALGORITHM:*

Text Rank is an unsupervised algorithm for summarizing the texts automatically that can be used to attain the most significant keywords in a document. This algorithm is a base of Page Rank over a graph constructed particularly for the summarization task. The only difference is what page rank uses ranking of pager while text rank used to rank sentences by using symmetrical graphs. This uses the ranking of the elements in the graph: the most essential elements are the ones that better describe the text. Text Rank used to build summaries without the need of a training corpus and allows summarizing different languages. The implementation is shown in Fig 6.

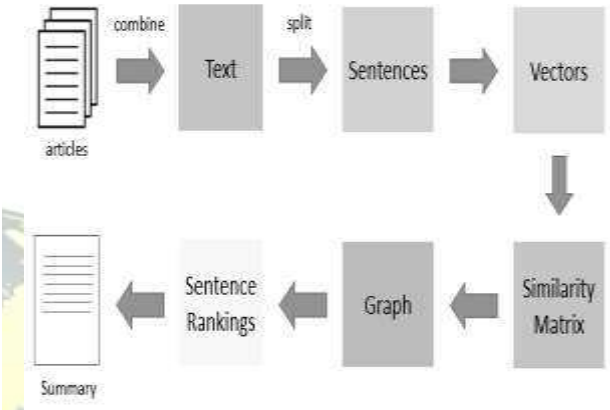

Fig.4 Process Flow of Text Rank Algorithm

1. Concatenating the text available in the article.
2. Then splitting the text into individual sentences.
3. Finding vector representation for every sentence.
4. Finding the similarities between sentence vectors are then calculated and stored as a matrix.
5. The similarity matrix is then converted into a graph, with vertices as sentences and edge as a similarity scores, for ranking sentence calculation.
6. The original Page Rank definition for graph-based ranking is assuming un-weighted graphs. However, in our model, the graphs are build from natural language texts, and may include multiple or partial links between the units (vertices) that are extracted from the text. It may be therefore useful to indicate and incorporate into the model the “strength” of the connection between two vertices and as a weight added to the corresponding edge that connects the two vertices.
7. A graph is then generated from this cosine similarity matrix. We will then apply the Page Rank ranking algorithm to the graph to calculate scores for each sentence.
8. In the case of Text Rank, we generate a cosine similarity matrix where we have the similarity of each sentence to each other. A graph is then generated from this cosine similarity matrix.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 6, June 2020

9. We will then apply the Page Rank ranking algorithm to the graph to calculate scores for each sentence.

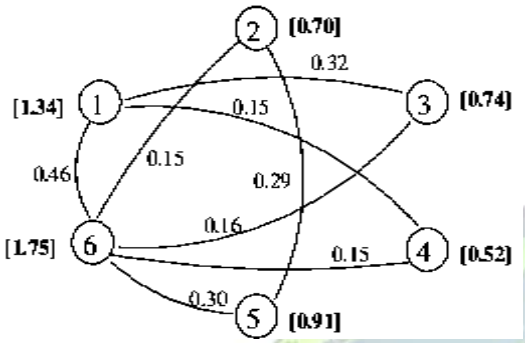

Fig.5 Sentence Vector Graph

10. With the help of graph the sentence scores will be obtained and join the top-ranked sentences which forms the final summary as shown in Fig 6.

Fig.6 Summary-Text Rank

3) FLOW DIAGRAM:

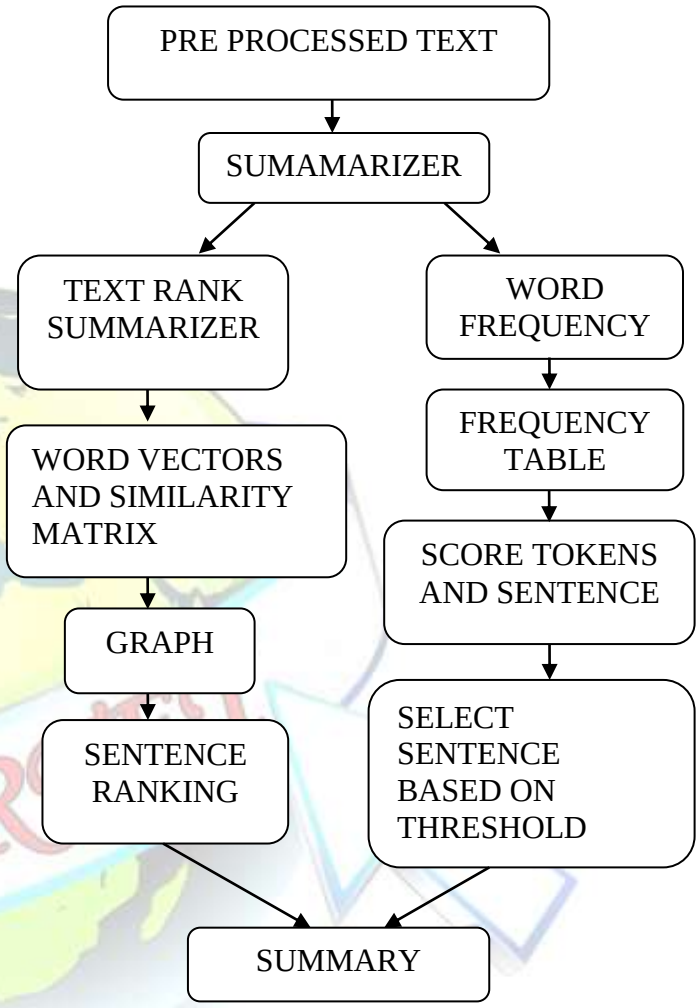

Fig.7 Summary-Text Rank

Fig 7.Represents the flow diagram of the entire application. It is a step by step process carried out in the system application. It illustrates that how the pre processed data is being processed and summarized based on two different algorithms.

IV. CONCLUSION

The GUI-based Automatic extractive Text summarization was designed and coded which emerges trends in education domains, biomedicine, product review,

emails and blogs. This is because there is information overload in these areas, especially on the World Wide Web. Automated summarization is an important area in NLP research. The purpose of extractive document summarization is to automatically select several indicative sentences, passages, or paragraphs from the original document. The objective of text summarization is to present the source text into an appropriate version with semantics. The primary plus of using summarization is to reduce reading time. Abstractive summarization requires complex code for language generation and it is difficult to make a model into the domain-specific areas. So using the extractive method of summarization, the Text rank algorithm made the summarizer application perform faster than any other extractive method and even it has the feature of converting text into audio. Since the Text rank algorithm is of statistical-based computations and graph-based sentence ranking it has a high ROUGE-1 score than any other extractive method of summarization. The final performance of the application is validated using ROUGE-N value which gives precision value of 0.604 which is better than any other extraction based text summarization. The sentences that are highly relevant to other sentences in the source text are likely to be more informative for the given text, and will be therefore given a higher score as being useful for the overall understanding of the text.

REFERENCES

[1]. H.Angel Castaneda, H.ReneArnulfo Garcia,Yulia Ledeneva and Christian Eduardo Millan-Hernandez,'Extractive Automatic Text Summarization Based on Lexical- Semantic Keywords', IEEE Access vol.8, pp.49896 – 49907, 2020

[2]. Alvee Rahman , Fahim Md Rafiq , Ramkrishna Saha , Ruhit Rafian and Hossain Arif,'Bengali Text Summarization using TextRank, Fuzzy C-Means and Aggregate Scoring methods'IEEE Region 10 Symposium (TENSYMP),pp. 331-336, 2019.

[3]. Devika, R. and Subramaniyaswamy. V, 'A semantic graph-based keyword extraction model using ranking method on big social data', Wireless Networks,Springer, pp.1-13, 2019.

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

[5]. K.Mona Teja, S.Mohan Sai , D.H S S S Ravitej , P.V.Sai Kushagra, 'Smart Summarizer for Blind People', 3rd International Conference on Inventive Computation Technologies (ICICT), pp.15-18, 2018.

[6]. Mudasir Mohd, Rafiya Jan and Muzaffar Shah, 'Text Document Summarization using Word Embedding',Expert Systems with Applications,Vol.143 Article 112958, 2019.

[7]. Reda Elbarougy, Gamal Behery and Akram El Khati, 'Extractive Arabic Text Summarization Using Modified Page Rank Algorithm', Egyptian Informatics Journal, 2019.

[8]. Suzan Verberne, Emiel Krahmer, Sander Wubben and Antal vanden Bosch,'Query-based summarization of discussion threads', Natural Language Engineering, Vol.26(1), pp.3-29, 2019.

[9]. K.S.Umadevi, Romansha Chopra, Nivedita Singh, Likitha Aruru and R.Jagadeesh Kannan,'Text Summarization of Spanish Documents', International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp.1793-1797,2018.

[10]. Akshi Kumar, Aditi Sharma, Sidhant Sharma and Shashwat Kashyap ,'Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization' International Conference on Computer, Communications and Electronics,pp.408 – 414,2017

[11]. Yogesh Kumar Meena, Dinesh Gopalani, 'Analysis of Sentence Scoring Methods for Extractive Automatic Text Summarization', International Conference on Information andCommunication Technology for Competitive Strategies(ICTCS),pp. 1-6,2014.