JES
Journal of Engineering Sciences

Vol 13, Issue 03, MARCH /2022

ISSN NO:0377-9254

AUTOMATIC TEXT SUMMARIZER APPLICATION USING EXTRACTIVE TEXT SUMMARIZATION

Tarun Sai Korukonda

Tarunrockzz308@gmail.com

Adarsh Vulli

adarshvulli@gmail.com

Machiraju Sai Satyam

saisatyam777@gmail.com

Gorrela Rikita Dharani

rikitadharani7532@gmail.com

Suresh Chittineni

schittin@gitam.edu

ABSTRACT: Text Summarization as a phenomenon has always been present and rather an evolving one with the advent of new technologies both in terms of data collection as well for the processing of this data. One reason of using text summarization is the huge amount of data floating over the internet in the form of text files, comments which is though potent enough to be used to extract useful information. but since the amount of text present in these sources is too huge, so the need of text summarization becomes justified by every argument. Some of the areas where text summarization is vastly used is applications involved in providing capsule information such as compact news applications, or websites providing academic notes for various examinations This paper presents an auto text summarizer application which takes the URL of a YouTube video as input, performs summarization on the selected elements and then presents this summarized text content on the front end of a web application. At the backend, the process of scraping of web page content (if an http URL is provided as input) using beautiful soup library or reading of text provided takes place. news in short forms, or micro blogging websites. The scraped content after being preprocessed properly is summarized using a suitable library which in our case is one among NLTK, Spacy, Genism and Sumy. The summarized content is presented at the frontend using flask framework of Python. The results produced using different libraries are compared in the end in terms of reading time of the summarized content. The application uses extractive text summarization technique in order to achieve its result which is a compact summary of the textual data prepared from the keywords already present in the document.

Keywords: ***Auto Text Summarizer, URL, Flask, Web Scraping, Nltk, Spacy, Sumy, Gensim, Extractive Text Summarization.***

1. INTRODUCTION

With the expansion of Internet, users nowadays are surrounded by a jargon of online information and documents.. This has led to a leap in the demand to have research in this domain . In [1] the summary of a given corpus of data is defined as “a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half ofthe original text(s) and usually, significantly less than that”.Automatic text summarization is the task of producing a shortyet a summary which is both meaningful as well as preserves the overall context of the document. The different approaches to text summarization can be broadly classified in 2 categories namely Extractive Text Summarization and Abstractive Text Summarization.

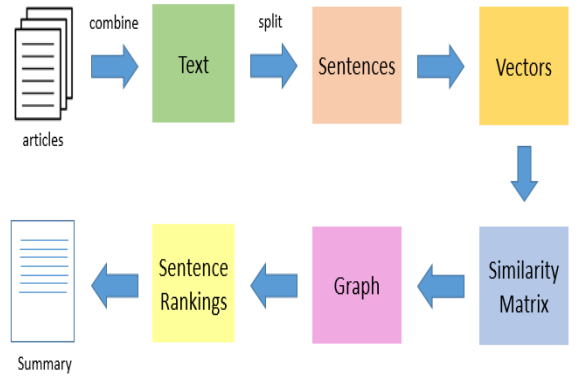

Fig.1: Steps in text summerization

Extractive Text Summarization works by extracting several pieces of information from the existing t text and group them together to create a summary. Abstractive Text Summarization techniques on the

other hand produce a summary which is both more coherent with the context of the given document as well has contains words and phrases other than those already present in the document .This type of summary is close to a summary produced by human understanding and hence is considered better than the one produced by extractive summarization. In order to obtain such a summary, it employs some of the most advanced techniques of Natural Language Processing. The application performs extractive summarization of a given webpage or a textual data using 4 different libraries as mentioned before and presents a relative comparison of results produced in terms of the estimated reading time of the resulting summarized content.

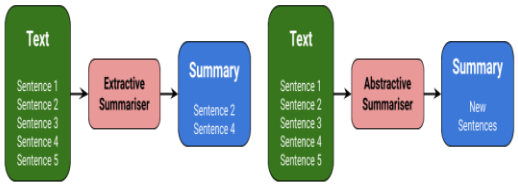

Fig.2: Extractive vs Abstractive summarisers

Extractive Summarisation

1. Identify the most important sentences or phrases from the original texts
2. Extract only sentences from the original text
3. Extracted sentences will be our summary
4. Mostly unsupervised learning

Abstractive Summarisation

1. Generate new sentences from the original text
2. Generated sentences may not be present in the original text
3. Much more difficult task, involving language understanding
4. Uses deep learning methods

2. PROPOSED SYSTEM

There is an enormous amount of textual information present in this world , and it is only growing every single day . Think of the internet which comprises news articles related to a wide range of topics webpages ,status updates, blogs and so much more. The data is unstructured and the best that we can do to navigate it is to use search and skim the results. There is a great need to reduce much of this text data to shorter text while preserving the important information contained in it. Summaries that capture the salient details, both so we can navigate it more effectively as well as check whether the larger documents contain the information that we are looking for. Textual information in the form of digital documents quickly accumulates large amounts of data. Most of this huge volume of documents is unstructured and has not been organized into traditional databases. Processing documents is therefore a difficult task.

As are no fixed guidelines for categorization on the techniques that we use for summary generation. Although for performing tasks in an organized way.

Approaches for automatic summarization:

Summarization algorithms are either extractive or abstractive in nature based on the summary generated. Extractive algorithms form summaries by identifying and pasting together relevant sections of the text. Depending only on extraction of sentences from the original text. For such a reason, extractive methods yield naturally grammatical summaries and require relatively little linguistic analysis. In contrast, abstractive algorithms are generally most human-like which mimic the process of paraphrasing a text.In this approach it may generate new text that is not present in the initial document. Texts summarized using this technique looks more human-like and produces condensed summaries which are easier to read . However, abstractive techniques are much harder to implement than extractive summarization techniques. Existing abstractive summarizers often rely on an extractive preprocessing component to produce the abstract of the text.

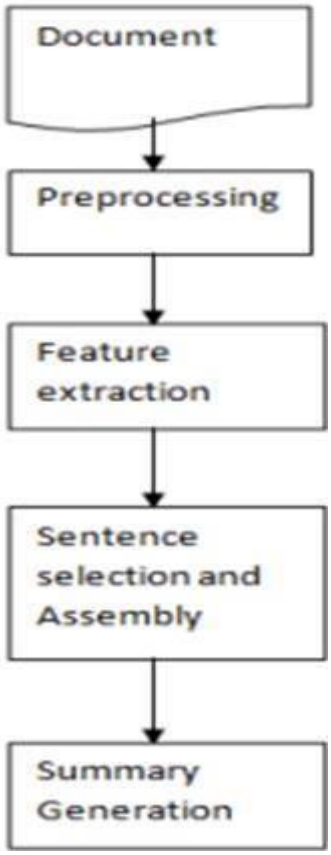

Fig.3: Extractive Text Summarization Process

MODULES:

1. Data Collection:

There can be multiple ways to fetch the input corpus of text data .It can be done either by importing a dataset ,or by scraping the textual data from the provided URL or it the text data can be simply fed directly to the application The proposed application the latter 2 options .The user can either provide a valid http URL of a webpage which allows scraping of textual data .Otherwise the data can be simply inserted in the other input field which accepts textual data .

2. Preprocessing Of Data:

The preprocessing of obtained raw data is performed so as to remove the anomalies such as redundancy, missing data in order to get the desired result in an appropriate manner.

3. Tokenization:

Using NLTK modules, the given data is first tokenized into sentences and then those sentences are further tokenized into words. The output of word tokenization is the complete list of words present inside the tokenized sentences. It is used to find the most relevant terms with respect to a document.

4. Weighted Frequency Of Words:

The weighted frequency of each word is calculated with respect to the maximum occurring term in the document.

5. Finding Weight of Sentences:

The words in each sentence are replaced with their weighted frequency. The sum of these frequencies is then calculated to obtain the final sum of each sentence.

6. Ranking sentences:

The sum of sentences is ranked in descending order of the sum of frequencies of words and the most relevant sentences are segregated in the final summary of the document.

7. Estimated reading time:

The reading time of a given text has been calculated by dividing the length of the document in terms of number of terms divided by 200.

3. RELATED WORK

3.1 Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization.

Automatic text summarization is the task of deriving a meaningful and concise brief from a given text while retaining the concept and key information conveyed by the original text. So far, numerous approaches and algorithms have been devised to achieve this goal with certain accuracy and effectiveness. One key aspect of text summarization is accurate identification of keywords from the given textual content. In this paper, the relative performance of three popular algorithms, namely TextRank, LexRank and Latent Semantic Analysis for keyword extraction were investigated by measuring their effectiveness in identifying keywords from set of articles. The performance of each of these

algorithms were contrasted with those of handwritten summaries of the same articles. The most effective algorithm was identified from the empirical results.

3.2 Sentiment Analysis and Text Summarization of Online Reviews: A Survey:

Sentiment analysis and text summarization has evoke the interest of many scientists and researchers in last few years, since the textual data has become useful for many real world applications and problems. Sentiment analysis is a machine learning approach in which machine learns and analyze the sentiments, emotions etc about some text data like reviews about movies or products. These reviews are increasing day by day, due to which summarization of reviews comes in role where summarized form of text in needed, which provides useful information from the large number of reviews. It is very difficult for a human being to extract useful data or summarize it from the very large document. In Text summarization, importance of sentences is decided based on linguistic features of sentences. This paper provides the comprehensive overview of recent and past research on sentiment analysis and text summarization and provides excellent research queries and approaches for future aspects.

3.3 Multiple Text Document Summarization System using Hybrid Summarization Technique:

Text Summarization plays an important role in the area of text mining and natural language processing. As the information resources are increasing tremendously, readers are overloaded with loads of information. Finding out the relevant data and manually summarizing it in short time is much more difficult, challenging and tedious task for a human being. Text Summarization aims to compress the source text into a shorter and concise form with preserving its information content and overall meaning. Summarization can be classified into two main categories i.e. extractive summarization and abstractive summarization. This paper presents a novel approach to generate abstractive summary from extractive summary using WordNet ontology. An experimental result shows the generated summary in well-compressed, grammatically correct and human readable format..

3.4 An Approach to automatic summarization for Chinese text based on the combination of spectral clustering and LexRank:

In the past half century, automatic summarization has been a hot topic in the field of natural language processing, and it will be paid more and more attention to with the rapid development of the mobile network technology. Most of the automatic summarization research today is on extractive summarization, which mainly ranks the sentences according to their simple heuristic features such as the frequency of words they contain, their position in the text or paragraph and so on. Inspired by the great performance of LexRank, we manage to introduce LexRank to Chinese texts. In order to make up the deficiency of LexRank, spectral clustering is adopted to process the component analysis. All in all, we propose an approach of extractive summarization for Chinese text based on the combination of spectral clustering and LexRank, which is of high coverage and low redundancy. It is demonstrated by experiments that our approach has been greatly improved compared to the original LexRank. In addition, our approach is easy to implement and robust to noise.

3.5 Semantic graph reduction approach for abstractive Text Summarization:

One of the important Natural Language Processing applications is Text Summarization, which helps users to manage the vast amount of information available, by condensing documents' content and extracting the most relevant facts or topics included. Text Summarization can be classified according to the type of summary: extractive, and abstractive. Extractive summary is the procedure of identifying important sections of the text and producing them verbatim while abstractive summary aims to produce important material in a new generalized form. In this paper, a novel approach is presented to create an abstractive summary for a single document using a rich semantic graph reducing technique. The approach summaries the input document by creating a rich semantic graph for the original document, reducing the generated graph, and then generating the abstractive summary from the reduced graph. Besides, a simulated case study is presented to show how the original text was minimized to fifty percent.

4. IMPLEMENTATION OF TOOLS USED

NLTK: NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc… In this article, we will go through how we can set up NLTK in our system and use them for performing various NLP tasks during the text processing step.

Fig.4: NLTK example figure

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more.

SPACY: spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython.The library is published under the MIT license and its main developers are Matthew Honnibal and Ines Montani, the founders of the software company Explosion. Unlike NLTK, which is widely used for teaching and research, spaCy focuses on providing software for production usage. spaCy also supports deep learning workflows that allow connecting statistical models trained by popular machine learning libraries like TensorFlow, PyTorch or MXNet through its own machine learning library Thinc. Using Thinc as its backend, spaCy features convolutional neural network models for part-of-speech tagging, dependency parsing, text categorization and named entity recognition (NER).

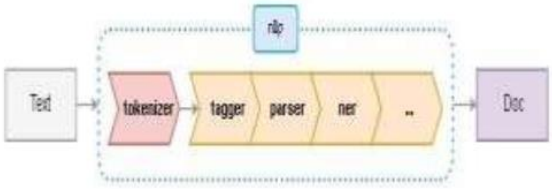

Fig.5: Spacy processing pipeline

SUMY: Sumy is able to create extractive summary. That means that it tries to find the most significant sentences in the document(s) and compose it into the shortened text. There is another approach called abstractive summary but to create it one needs to understand the topic and create new shortened text from it. This is out of the scope of Sumy's current capabilities.

GENSIM: Gensim is an open-source library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. Gensim is implemented in Python and Cython for performance. Gensim is designed to handle large text collections using data streaming and incremental online algorithms, which differentiates it from most other machine learning software packages that target only in-memory processing.

5. METHODOLOGY

A Web application was developed using flask framework in Python programming language .It consists of 2 input fields provided for getting input either in the form of textual data or as a Web URL .The summarized results are presented along with the reading time of provided text as well as that of the summarized content .The summary of a document as produced by 4 different libraries can be compared using the "Compare Summarizers" option on the home page.

Following are some of the snapshots of the application:

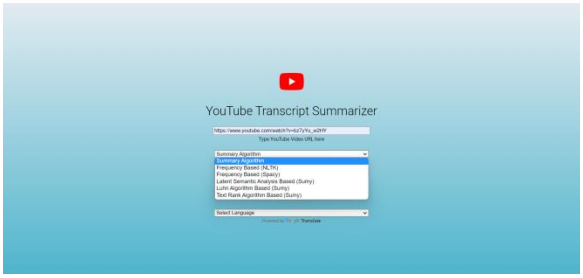

Fig.6: Input screen

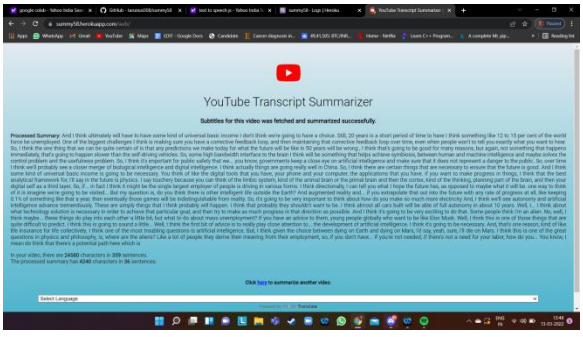

Fig.7: Spacy

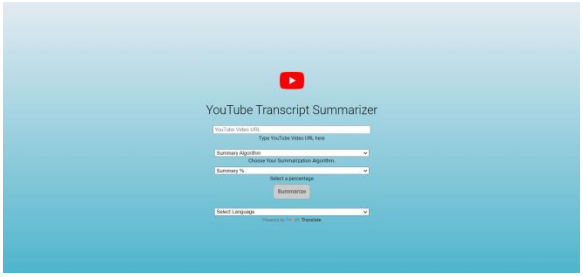

Fig.8: Result screen

ALGORITHMS	TOTAL CHARACTERS	CHARACTERS AFTER SUMMARIZATION	APPROACH
NLTK	24580	5132	TF-IDF
spaCY	24580	4240	TF-IDF
LSA	24580	4810	Algebric-Statistical(unsupervised learning)
Luhn's heuristic	24580	4340	TF-IDF
TextRank	24580	4774	graph-based(unsupervised)

Fig.9: Summarization Table

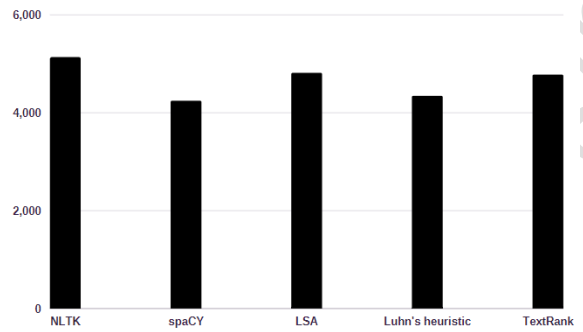

Fig.10: Analysis

6. CONCLUSION

In this paper an automatic text summarizer application using Flask framework was successfully built. The application summarized the contents of the textual data presented to it either as simple text or through a valid Web URL. Summarization was performed independently using NLTK and 3 other libraries in Python programming language namely Spacy, Sumy and Gensim. The results obtained could be compared using the comparing option on the application where the comparison was based on the estimated reading Time of the summarized content. This application can be further integrated with other applications as per their demand, due to the versatility of summarized results provided by the application.

7. FUTURE SCOPE

In this paper the application employed various algorithmic approaches of extractive text summarization such as Lex Rank. While it was able to produce a summary of the provided textual data successfully, one major scope in future is to implement it using Abstractive Summarization techniques. Abstractive Summarization produces summaries which are nearer to human understanding and are more coherent to the context of the text provided. In addition to this, in future other libraries can be used to implement the application in order to get better results.

REFERENCES

[1] Akshi Kumar, Aditi Sharma, Sidhant Sharma,Shashwat Kashyap, "Performance Analysis of Keyword Extraction Algorithms Assessing Extractive Text Summarization." International Conference on Computer, Communication,and Electronics (Comptelix), 2017)

[2] Pankaj Gupta, RituTiwariand Nirmal Robert, "Sentiment Analysis and Text Summarization of Online Reviews: A Survey." International Conference on Communication and Signal Processing, 2016.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Harsha Dave, Shree Jaswal, "Multiple Text Document Summarization System using Hybrid Summarization Technique." 1stInternational Conference on Next Generation Computing Technology (NGCT), 2015..

[4] Kang Wu, Ping Shi, Da Pan, "An Approach to automatic summarization for Chinese text based on the combination of spectral clustering and LexRank." IEEE Access 2016..

[5] Ibrahim F. Moawad, Mostafa Aref, "Semantic graph reduction approach for abstractive Text Summarization." Seventh International Conference on Computer Engineering & Systems(ICCES), 2012.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[6] K.Sparck Jones, "Automatic Summarising:The State ofthe Art" Information Processing & Management, vol. 43, pp. 1449-1481, Nov 2007.

[7] Radha Mihalcea, Paul Tarau, "TextRank: Bring Order into Texts." Association for Computational Linguistics, 2004.

[8] N. Moratanch, Dr. S. Chitrakala, "A Surveyon Abstractive Text Summarization."

[9] K.Sparck Jones, "Automatic Summarising:The State ofthe Art" Information Processing & Management, vol. 43, pp. 1449-1481, Nov 2007.

[10] Bhavana Lanjewar," Automatic text summarization withcontextbasedkeyword extraction ", International Journalof AdvanceResearch in Computer Science and Management Studies, Vol. 3, Issue 5, May 2015.

[11] Bhavana Lanjewar," Automatic text summarization withcontextbasedkeyword extraction ", International Journalof AdvanceResearch in Computer Science and Management Studies, Vol. 3, Issue 5, May 2015.

[12] GlorianYapinus, Alva Erwin, MaulahikmahGaliniu, WahyuMuliady, "Automatic Multi-Document Summarization for Indonesian Documents Using Hybrid Abstractive-Extractive Summarization Technique". 6thInternational Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 2014.

[13] S.ABabara, Pallavi D. Patilb, "Improving Performance of Text Summarization". International Conference on Information and Communication Technologies ICICT, 2014.

[14] Elena Lloret and Manuel Palomer, "Challenging issues of automatic summarization: relevance detection and qualitybased evaluation." Informatica 34, no. 1, 2010.

[15] T. Givón,T. "Isomorphism in the Grammatical Code: Cognitive and Biological Consideration." In R. Simone (ed.). 47-79, 1994.