Automatic Text Summarizer Application Using Extractive Text Summarization

Ritesh Kumar Bharadwaj

School of Computing Science and Engineering

Galgotias University Greater Noida, India

Riteshkumar.Bharadwaj@galgotiasuniversity.edu.in

***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 web page 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

I. 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 of the original text(s) and usually, significantly less than that".Automatic text summarization is the task of producing a short yet 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.

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.

II. Literature review

In [1] a comparative analysis of performance of 3 different algorithms, based on keyword extraction namely Latent Semantic Analysis (LSA), Text Rank and Lex Rank. is presented. Extraction based techniques rely upon extracting keywords from the original text and attempt to present a summary out of them. The paper used Python programming language in order to implement the algorithms. ROGUE 1 is used to evaluate the significance of extracted keywords. A comparison of different techniques of sentiment analysis and text summarization has been presented in [2] In [3] a system to generate an abstractive summary from the extractive summary of a document has been proposed. The paper describes WordNet Ontology based method for achieving this. WordNet is a lexical database which consists of relationship between words of more than 200 languages on the basis of their semantic properties It is used to relate words present in a document with their synonyms and antonyms on the basis of their semantic relationships. [7] introduces a model for processing of the text using graph-based ranking, namely Text Rank Text Rank is an unsupervised method for keyword and sentence extraction. A review of various abstractive text summarization techniques has been provided in [8]. A survey of various effects of preprocessing on extractive summarization has been reviewed in [11]. In [22] automatic text summarization of Wikipedia articles has been attempted.[23] proposes text summarization of documents using K-Nearest Neighbors (KNN) based on feature

similarity. Using sentiment analysis and text summarization [

III. TEXT SUMMARIZATION PROCESS

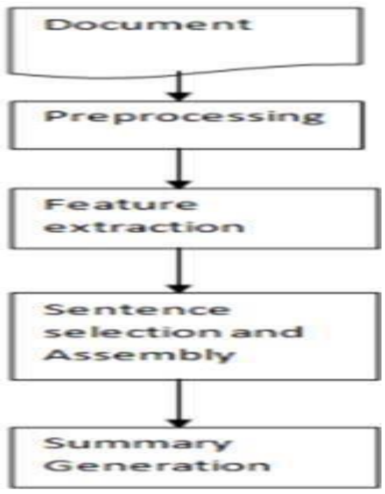


Fig. 1 Extractive Text Summarization Process

A. 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 .

B. 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.
Some of the steps of preprocessing are aa follows:

1. Converting paragraphs to sentences

Instead of analyzing paragraphs as a whole is not suitable for preparing a summary, so a paragraph can be split into multiple sentences using some splitting function.

2.Removal of stop words

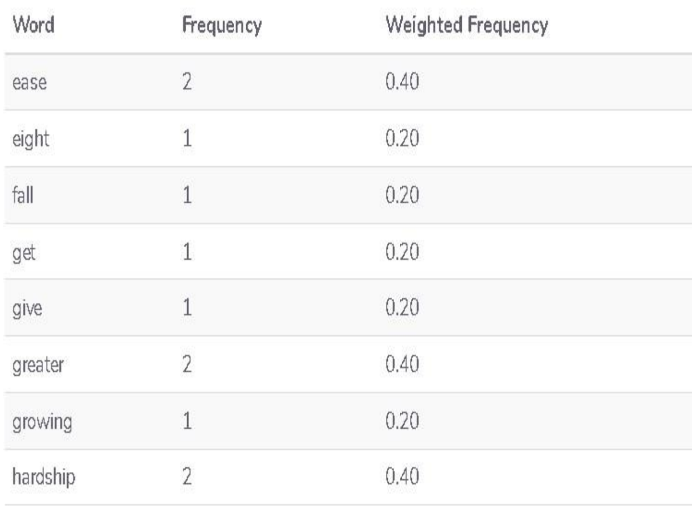
Stop words are those words which are the most common words in a document and hence should be refrained from being considered while preparing the summary of the document. The application uses NLTK corpus in order to remove the matching words from the given input text data.

C. 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.

D. Weighted Frequency Of Words

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



Word	Frequency	Weighted Frequency
ease	2	0.40
eight	1	0.20
fall	1	0.20
get	1	0.20
give	1	0.20
greater	2	0.40
growing	1	0.20
hardship	2	0.40

Fig. 2 Finding Weighted Frequency of Words (Example)

E. 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.

F. 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.

G. 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.

IV. OVERVIEW OF USED TOOLS OF SUMMARIZATION

A. Nltk

It is a large collection of libraries and programs intended for supporting natural language processing tasks in Python programming language. The various tasks which are possible with this package includes preprocessing of data such as removing stop words from the data, lexical analysis of data which includes word tokenizing, sentence tokenizing etc. The application uses NLTK corpus in order to remove stop words from the input data and uses appropriate modules to tokenize the text into sentences and then further into words.

B. Spacy

Spacy is a free open-source library which is used for performing some advanced natural language processing tasks in Python programming language.

Some of the tasks which can be done using. spacy are:
1. Dependency Parsing

2.Named Entity Recognition

3.Entity Linking

It also consists of a wide variety of statistical models which are basically of 2 types:

1.Core models: These are pretrained models which are used for generic purposes such as tagging parts-of-speech etc.

2.Starter models: These are more advanced models and are used to assist as support models while training the existing models.

Spacy consists of a processing pipeline of various components which is used to transform a text document into a doc file which is then used for other purposes by the statistical models.

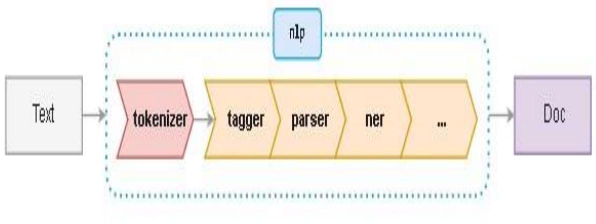


Fig. 3 Spacy processing pipeline

The application uses this pipeline which starts from nlp which when called on a text input is used to tokenize it and convert into a doc. Some of the components of this processing pipeline are Tokenizer, tagger, parser and ner which is an entity recognizer and is used for detecting and labelling named entities.

C. Sumy

Sumy is a python library which is used specifically for the purpose of extracting summary out of HTML content from webpages. It consists of various summarization methods as well as an evaluation framework for produced summaries.

Some of the summarization methods that are available inside this library are:

1.Luhn-It is a heuristic based summarization method based on TF-IDF score.

2.Latent Semantic Analysis

3.TextRank

4.SumBasic

5.KL-Sum

In this paper the application uses Lex Rank algorithm which is an unsupervised approach based on PageRank algorithm used by Google to sort the result webpages based on the keywords present in the search query.

D. Gensim

It is a Python library which is employed in various tasks related to natural language processing such as topic modelling, document indexing, similarity retrieval Some of the key features of genism include: 1. It is independent of the corpus size of input text.
2. Provides multicore implementation of algorithms like Latent Semantic Analysis, Latent Dirichlet Association, Random Projections (RP), Hierarchical Dirichlet Process (HDP) or word2vec deep learning.

3.It also supports distributed computing of various algorithms like Latent Semantic Analysis (LSA).

V.Results

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:

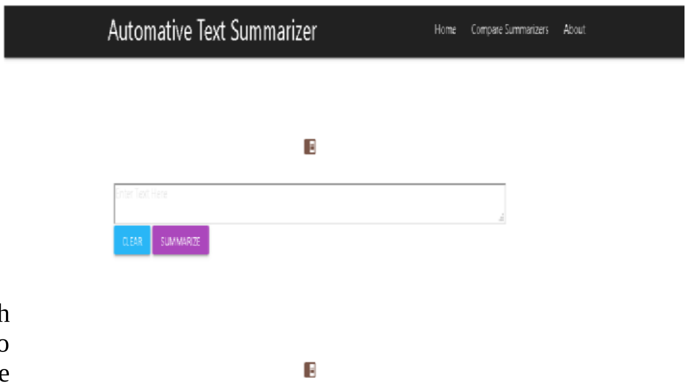


Fig 4 Home Page of Application

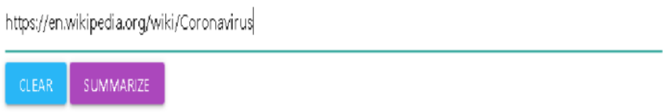
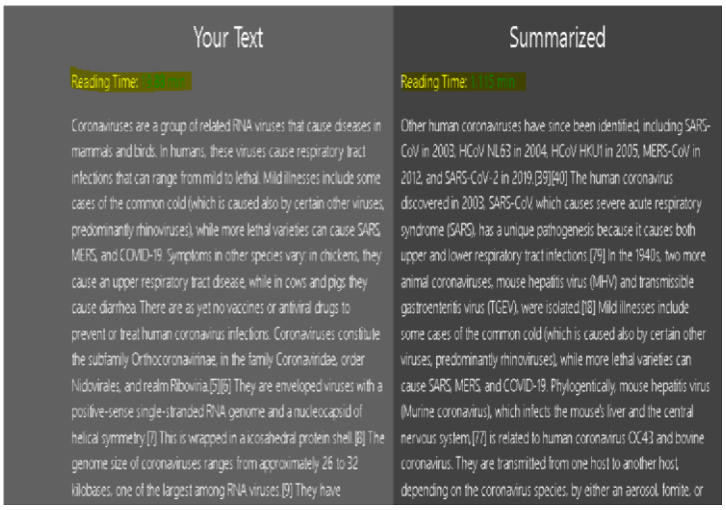


Fig 5 Input as a Web URL


Your Text

Reading Time: 19.88 min

Coronaviruses are a group of related RNA viruses that cause diseases in mammals and birds. In humans, these viruses cause respiratory tract infections that can range from mild to lethal. Mild illnesses include some cases of the common cold (which is caused also by certain other viruses, predominantly rhinoviruses), while more lethal varieties can cause SARS, MERS, and COVID-19. Symptoms in other species vary: in chickens, they cause an upper respiratory tract disease, while in cows and pigs they cause diarrhea. There are as yet no vaccines or antiviral drugs to prevent or treat human coronavirus infections. Coronaviruses constitute the subfamily Orthocoronavirinae, in the family Coronaviridae, order Nidovirales, and realm Riboviria.[5][6] They are enveloped viruses with a positive-sense single-stranded RNA genome and a nucleocapsid of helical symmetry.[7] This is wrapped in a icosahedral protein shell.[8] The genome size of coronaviruses ranges from approximately 26 to 32 kilobases, one of the largest among RNA viruses.[9] They have

Summarized

Reading Time: 1.115 min

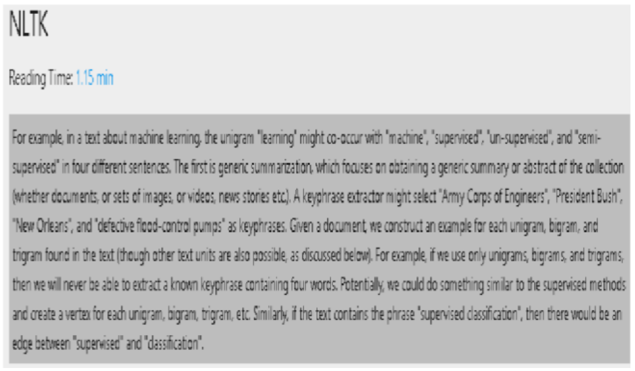
Other human coronaviruses have since been identified, including SARS-CoV in 2003, HCoV NL63 in 2004, HCoV HKU1 in 2005, MERS-CoV in 2012, and SARS-CoV-2 in 2019.[39][40] The human coronavirus discovered in 2003, SARS-CoV, which causes severe acute respiratory syndrome (SARS), has a unique pathogenesis because it causes both upper and lower respiratory tract infections.[79] In the 1940s, two more animal coronaviruses, mouse hepatitis virus (MHV) and transmissible gastroenteritis virus (TGEV), were isolated.[18] Mild illnesses include some cases of the common cold (which is caused also by certain other viruses, predominantly rhinoviruses), while more lethal varieties can cause SARS, MERS, and COVID-19. Phylogentically, mouse hepatitis virus (Murine coronavirus), which infects the mouse's liver and the central nervous system,[77] is related to human coronavirus OC43 and bovine coronavirus. They are transmitted from one host to another host, depending on the coronavirus species, by either an aerosol, fomite, or

Fig.6 Resultant Summary Of webpage

In figures 4, and 5 (shown above), there are 2 input fields meant for taking text input in one of the two ways (as explained above). The user can choose to enter the text using either of the choices and press on "Summarize" button provided below.

Fig. 6 shows the summarized output of the text provided, with reading time of both the texts highlighted

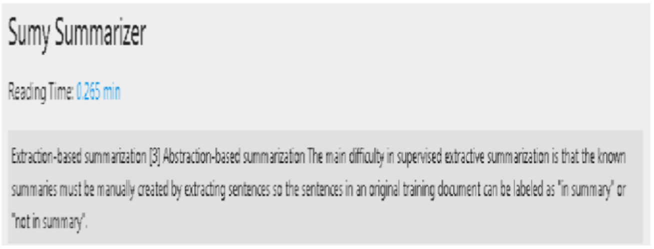
The summary using different libraries are as follows:


NLTK

Reading Time: 1.15 min

For example, in a text about machine learning, the unigram "learning" might co-occur with "machine", "supervised", "un-supervised", and "semi-supervised" in four different sentences. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). A keyphrase extractor might select "Army Corps of Engineers", "President Bush", "New Orleans", and "defective flood-control pumps" as keyphrases. Given a document, we construct an example for each unigram, bigram, and trigram found in the text (though other text units are also possible, as discussed below). For example, if we use only unigrams, bigrams, and trigrams, then we will never be able to extract a known keyphrase containing four words. Potentially, we could do something similar to the supervised methods and create a vertex for each unigram, bigram, trigram, etc. Similarly, if the text contains the phrase "supervised classification", then there would be an edge between "supervised" and "classification".

Fig.7 Summary Using NLTK (**Reading Time:1.15 minutes**)


Sumy Summarizer

Reading Time: 0.265 min

Extraction-based summarization [3] Abstraction-based summarization The main difficulty in supervised extractive summarization is that the known summaries must be manually created by extracting sentences so the sentences in an original training document can be labeled as "in summary" or "not in summary".

Fig.8 Sumy Summary **(Reading Time:1.225 minutes)**

Similarly, summaries using other libraries were obtained along with their respective estimated reading time.

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.

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 quality-based 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.

[16] H.P.Edumundson, “New Methods in automaticExtracting.” In: Inderjeet Mani and Mark Maybury, editors, Advances in Automatic Text Summarization, MIT Press pp. 23-42, 1969.

[17] H.P. Luhn, “The Automatic Creation of literature abstracts.” In: Inderjeet Mani and Mark Maybury, editors, Advances in Automatic Text Summarization, MIT Press pp. 15-22, 1958.

[18] E. Lloret, O. Ferrández, R. Muñoz, M. Palomar, “A Text summarization Approach Under the Influence of Text Entailment.” In: Proceeding of the 5thInternational Workshop on Natural Language Processing and Cognitive Science (NLPCS 2008) 1216 June, Barcelona, Spain. 22-31, 2008.

[19] D.R.Radev, S. Blair-Goldensohn,Z. Zhang, Similarity.” International Conference on Communication, Control, “Experiment in Single and Multi-Document

Summarization Computing and Electronics Engineering (ICCCCEE), 2017. using MEAD." In: First DocumentOrleans, LA. 1-7, 2001.

[20] M. Fuentes Fort, "A Flexible Multitask Summarizer for Document from Different Media, Domain and Language." Ph.D.thesis (2008) Adviser-Horacio Rodríguez.

[21] . Mani, "Summarization Evaluation: An Overview." In: Proceeding of the North American Chapter of the Association for Computational Linguistics(NAACL). Workshop on Automatic Summarization, 2001.

[22] DharmendraHingu, Deep Shah, Sandeep S.Udmale, "Automatic Text Summarization of Wikipedia Articles." InternationalConference on Communication, Information & Computing Technology (ICCICT), 2015.

[23] Taeho Jo, "K Nearest Neighbor for TextSummarization using Feature