ff

by Surekha Kumari

Submission date: 30-Apr-2023 11:56PM (UTC-0500)
Submission ID: 1851114041
File name: big_data.pdf (3.07M)
Word count: 1654
Character count: 8957

Document Summarization Using NLP

Sai Abhishek Thota
Computer Engineering
University of Texas at Dallas
Dallas, Texas
saiabhishek.thota@utdallas.edu

Saicharan Reddy Ragannagari
Computer Engineering
University of Texas at Dallas
Dallas, Texas
saicharanreddy.ragannagari@utdallas.edu

Sai Shantan Nagelli
Computer Engineering
University of Texas at Dallas
Dallas, Texas
saishantan.nagelli@utdallas.edu

Surekha Kumari
Computer Engineering
University of Texas at Dallas
Dallas, Texas
surekha.kumari@utdallas.edu

***Abstract*—In this paper we discuss the implementation of a method to summarize documents irrespective of its size. With abundance of data and information, it is essential that there be a method to automatically summarize documents in order to save people the effort and time in selecting relevant information. Automating summarizing is crucial to the amount of information that can be processed when compared with manually summarizing and this method is also wary of human biases. This paper further discusses the algorithm used and implementation steps.**

***Index Terms*—implementation, summarize, algorithm [1]**

I. Introduction

We have worked on the Python implementation of the text summarization algorithm, which seeks to extract the most important sentence from a given paragraph and generate summaries. The algorithm uses (NLP) techniques such as word frequency, sentence relevance, and graph-based algorithms to identify the most salient information in the text. That extraction approach is the most widely used method of summarising text, given that it provides greater clarity and interpretation compared with abstract methods, which generate summaries based on paraphrasing and rewording. However, the whole meaning of the original text cannot be taken into account by extractive methods. we achieve this by using libraries such as spaCy and NLP library NLTK, we built a code around it to paraphrase long paragraphs into meaningful sentences to form a summary. This paper elaborates on the libraries used and algorithms implemented to achieve the desired result.

***Index Terms*—NLP, NLTK, spaCy**

II. Background work

A. NLP

NLP stands for Natural Language Processing, which is a subfield of Artificial Intelligence used to train machines to understand Natural languages used by humans. It helps machines understand human language to perform redundant tasks post automation.

B. NLTK

NLTK stands for Natural Language ToolKit which is a python library used to aid NLP.

C. SpaCy

spaCy is a highly advanced, open source library used to perform NLP. It uses the best algorithms and is faster than NLTK when it comes to word tokenizing.

D. Stopwords

Stopwords are commonly used words which are irrelevant to the machine when constituting meaning from the text. It generally consists of articles, prepositions, conjunctions, pronouns, etc which are present in a text to aid human understanding. As machines do not need typical grammar, stopwords are filtered out for NLP.

III. Theoretical and Conceptual Study

Text summarization, which can be used for a variety of applications in different areas, such as news summaries, documents, and chatbot responses, is an extensively discussed problem within the field of natural language processing NLP. The summarization of texts is made up of two main approaches: extractive and abstract. The best sentences or phrases in the original texts are selected by an extract summarization algorithm that uses them to produce a summary. These algorithms typically use statistical methods such as word frequency, sentence relevance, and graph-based algorithms to identify the most salient information. Generally, extraction methods are quicker to interpret than abstract methods, but they can't describe the whole meaning of a text. On the other hand, paraphrasing and rewording an existing text is also commonly done in abstract summarization algorithms that use deep learning models like neural networks to produce summaries. Abstractive methods require more computational resources and may introduce errors or biases but are able to provide better clarity and consistency of summaries. Before,

improvements in the accuracy and efficiency of both extractive and abstract algorithms have been a major focus of work on text summarization. Hybrid approaches with overlapping strengths between these methods were also developed. To enhance the quality of text summarization, techniques such as compressed sentences, naming entity recognition, and scenography analysis have been investigated.

IV. Project Flow

A. Code Description

Our Code implements text summarization algorithm which uses frequency-based approach to identify and extract meaningful sentences from the provided information. This algorithm uses several libraries such as pySpark, NLTK, spaCy and spark-nlp to achieve desired goal. The code is designed in order to first process the provided input with removing stop words(commonly used words that are irrelevant to the machine when constituting meaning data from the text. It generally consists of articles, prepositions, conjunctions, pronouns, etc which are present in a text to aid human understanding. As machines do not need typical grammar, stopwords are filtered out for NLP). We then use urlLib specifically to store the above text so that we have the output in String format rather than table for ease of further steps. We further convert our string to spaCy readable format with Doctext to provide as input to spaCy operations. We use spaCy for dependency parsing and named entity recognition. We then calculate frequency of each word by assigning score to each sentence. The individual scores are then used to find max score to find top 30% words used to find most relevant sentences. The summary is finally generated using these sentences to capture important information.

We selected this algorithm under the assumption that most commonly words used are important and carry higher weight and should be present in the summary. When calculating individual frequency of the important words, the algorithm can identify words crucial to the document and can create condensed summary with relevant information.

Our output contains a summary of input document with all crucial information identified with NLP. This is achieved by selecting most relevant words and combining them together to form meaningful conclusions. This way of processing document is helpful when there is a need for quick extraction of information from huge documents or several documents that need to be precessed in smaller time frames.

V. Limitations

our one potential limitation is that we rely solely on frequency of the words and may not provide precise summary in case of complex or nuanced language documents. Another limitation could be because the algorithm only extracts sentences and does not generate new sentences or paraphrase existing sentences, which may limit its ability to produce summaries that are highly readable or stylistically similar to the original text.

VI. Results

For this project, we are trying to sum up a paragraph of any size in its smaller form with no loss of integrity. Consequently, we used a summarizing algorithm from the extracted text. It is in its work that it finds the most important and relevant sentences of a paragraph, converting them into their shorter form.

Based on the project flow described, we have been able to find a frequency for each word in paragraphs 1, 2, and 3:

Fig. 1. word frequency

Then normalized frequency score is calculated for every word by dividing each frequency of a word by the maximum frequency. Following is the output for the line scores:

Fig. 2. Summary

Line score is the sum of normalized frequency scores in a sentence We are now going to get sentences with the top-line score. We wanted our algorithm to get 30% as a summary of the original document, so we selected the 30We have used the nlargest function in the heapq library. nlargest will return the list of sentences with the amount of selected percentage we want.

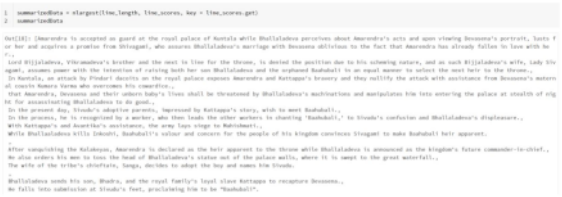

Fig. 3. line frequency calculation

The summary of amount of words in the actual paragraph and the amount of words in the summarized paragraph is shown below

VII. Conclusion and Future Work

Our code is a result of frequency based approach to extract text summaries. Algorithm used identifies the most relevant sentences in a text and then extract those sentences to combine and form a summary of the whole input provided. In conclusion, though we do have limitations, by using extractive text summarization, this approach is quick in extracting most important information from a large body of text, we now know the significance/effectiveness of the NLP in producing exact summaries of huge paragraphs. This algorithm selects the most important sentence in the paragraphs and adds all those into one summary. Our current implementation is limited to summarization only has room for improvement in areas where more precise information is to be extracted. The project could be extended to create presentation from extracted summaries, post automated online reviews.

Acknowledgment

It is my sincere thanks to Professor Anurag Nagar for his invaluable assistance, valuable observations, and constant support in the course of this project. He was instrumental in shaping our understanding of the subject and helping us achieve our objectives through his expertise in Natural Language Processing. Our heartfelt thanks go out to our teacher's assistant, Truong Quang Pham, who has been doing everything he can to let us know our questions and help us keep track of the project.

We would like to thank our team for their contribution, SaiCharan Reddy, Sai Abhishek Thota, Sai Shantan Nagelli, and Surekha Kumari, who have played a major role in this project. This project's successful achievement was largely made possible through their dedication, hard work, and creativity. Lastly, we'd like to say our deepest thank you to each and everyone who has supported us in this endeavor directly or

Fig. 4. final output

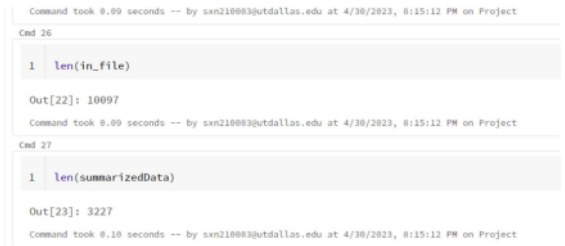

Fig. 5. length of summary

indirectly. Encouragement and feedback have been the source of motivation for us, and we appreciate your support.

REFERENCES

[1] Dan W. Patterson, Introduction to Artificial Intelligence and Expert System , PHI, 2001, Chapter 12
[2] https://en.m.wikipedia.org/wiki/India
[3] https://en.m.wikipedia.org/wiki/Dallas
[5] https://en.m.wikipedia.org/wiki/Spider-Man:_Far_From_Home
[5] https://en.m.wikipedia.org/wiki/Baahubali:_The_Beginning
[6] https://en.m.wikipedia.org/wiki/Titanic_(1997_film)

ff

ORIGINALITY REPORT

7%	3%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Larice Toko Lumanda. "Reflexivity in English, French and Kinshasa Lingala: Similarities and Differences", Communication and Linguistics Studies, 2019 Publication	1%
2	Submitted to UT, Dallas Student Paper	1%
3	krchowdhary.com Internet Source	1%
4	Delva Culp. "Developmental apraxia and augmentative or alternative communication–a case example", Augmentative and Alternative Communication, 2009 Publication	1%
5	Submitted to Mentone Grammar Student Paper	1%
6	Submitted to Kingston University Student Paper	1%
7	www.analyticsvidhya.com Internet Source	1%

Exclude quotes	Off	Exclude matches	< 3 words
Exclude bibliography	Off		