Effective Text Processing utilizing NLP

Dharmaiah Devarapalli, Srija Padmini Guduri, Pericharla Jaya Madhuri, Sathi Navya Vahini Reddy, Pavani Pasupuleti

Abstract: Summarizing is the practice of condensing a body of material into a more manageable size while retaining all of the key data elements and the intended meaning. Automatic text summarization systems can now quickly retrieve summary phrases from input documents. However, it is a number of shortcomings, such as duplication, insufficient coverage, incorrect extraction of key lines, and poor sentence coherence. In this study, a new concept of summarizer technique is proposed using the Python spacy package. It extracts the most significant information from the text. The scoring system is also used to compute the score for the words in order to determine the word frequency. The findings show that the proposed method completes the summary process faster than the current algorithm. An online tool called the text to summary converter aids in material summarizing. This programmer will give us a summary of the data that we upload. The primary goal is to accurately summaries the data entered. The most crucial sentences will be removed before the unnecessary ones.

Keywords: Spacy, NLP, stops words, word frequencies, Text to summary converter

I. INTRODUCTION

In recent years, there has been an increase in the amount of textual material available. The production of text-based data is constantly increasing [2]. The user loses interest when the textual material becomes increasingly challenging for them to read. Text Summarization was developed to solve this issue. Data mining has rapidly expanded in recent years as a result of significant advancements in hardware and software technologies. More types of data are becoming accessible because of technological advancements, which is particularly advantageous for text data. [7]. Platforms for social networks and the web's software and hardware have accelerated the development of massive collections of data of all kinds. Text data is often managed by a search engine since it lacks structures, whereas structured data is frequently maintained by a database system. [9]. The search engine enables the internet user to apply a keyword query to find the pertinent information from the collected works. [5]. Text summarizing is a technique for condensing a lengthy original text into a more condensed version, producing a summary of the original topic[14]. The major points and significant passages from the original book serve as the foundation for the summary. As a result, the reader has both an understanding of the original material and a narrowed perspective of it. Automated text summarization utilizes computer systems to construct a text summary of papers while preserving their main phrases, which helps minimise the length of text documents [4]. Automated text summarization is the technique of employing computer algorithms to extract and describe significant information from a given material [6]. A computer software emulated human reading patterns for choosing “subject sentences” and phrases made up of nouns and modifiers [18]. The automatic production of a concise and useful summary of a lengthy text is known as text summarization, and it is a crucial problem in the field of natural language processing (NLP) [19].

II. DEFINITION OF A PROBLEM

The amount of text data accessible from various sources has recently increased. This body of literature is a fantastic resource for knowledge and information, but it needs to be effectively summarized in order to be effective. The main goal of the issue is to automatically sum up the text [5]. People are becoming overwhelmed by the abundance of online information and articles as a result of the Internet's rapid development [3]. Further investigation on automated text summarizing is required due to the rise in paper production. The number of words before and after the summary will also be stated.
Effective Text Processing utilizing NLP

III. METHODOLOGY

A. Natural Language Processing (NLP)

The simplest definition of NLP is “training an algorithm to read and analyse human (natural) languages in the same way that a human does,” but more quickly, more accurately, and on considerably bigger datasets [5]. It used to take a lot of physical labour to create a textual content summary.

B. SpaCy:

A new Python package called SpaCy was developed for "Industrial-strength Natural Language processing.” In comparison to NLTK, SpaCy is a significantly more recent NLP library [13]. It can help us create apps that effectively process vast amounts of text because it is designed for use in production settings [15].

C. Heapq Library:

Binary trees called heaps have parent nodes that are equal to or less valuable than any of their offspring. [12]. Using arrays, this approach uses heap[k]=heap[2*k+1] for all k, counting from...
0. In order to compare, nonexistent things are represented as infinite. A heap's root is always its tiniest component.

D. **String:**

Constants for manipulating strings as well as practical functions and classes are available in the Python String module [17].

IV. **DATA SET DESCRIPTION**

A. **Data Gathering:**

Information must be gathered for data gathering from a variety of sources. We use a lot of text from various sources, like newspapers, Wikipedia, and other sources, in our project.

B. **A Pre−Processing Step:**

Text is a remarkably rich source of information. Every minute, hundreds of millions of fresh emails and texts are sent [16]. There is a mountain of text data that is simply begging to be mined for knowledge. Stop words, punctuation, and capital words were deleted, along with other stages like entity detection, tokenization, and parts of speech (POS) tagging [1].

C. **Tokenization:**

Tokenization is breaking up text into tokens and removing characters like spaces and punctuation marks (., ”). The tokenizer in spaCy generates a series of token objects using unicode text as input [13]. Word tokenization is the process of separating the text into its component words. This is a crucial step because many language processing algorithms need input in the form of single words rather than long text strings [11].

D. **Designing the model:**

The design of the model comes next.

1) Text cleaning: Stop words, punctuation, and uppercase and lowercase word substitutions were made[3].

2) Tokenization of Words: Tokenize each Word in Sentences [7].
3) Word Frequency Table: Count the frequency of each word and divide the maximum frequency by each frequency to obtain the normalised word frequency count. [8]

![Fig. 6. Word Tokenization](image)

4) Sentence Tokenization: Based on sentence frequency.

5) Recapitulation
The five steps that go into making our model are as follows.

**E. Testing the model:**
These are the several evaluation phases that the model uses to vouch for its ability to predict outcomes correctly. Preparing for the testing set is the first stage. The results we obtained when testing the model are practically exactly what we predicted [10], as shown in Figures 8, 9, and 10.
F. Model implementation:

ALGORITHM: Text INPUT; Summary OUTPUT
Step 1: Import packages
Step 2: Text preprocessing
Step 3: Word tokenization, word separation
Step 4: Word frequency table, Count the frequency of each word and divide the maximum frequency by each frequency to get the normalized word frequency count. [8].
Step 5: Sentence Tokenization: as determined by sentence frequency
Step 6: SUMMARY
Save the model for later usage in step

V. RESULTS

We were so near the anticipated results. Using this straightforward web application, obtaining a summary from text is a piece of cake [20]. These are the outcomes. These data, while meeting our predictions, nevertheless seem to be lacking some crucial information. We intend to upgrade it in the future for a better user experience. The number of words before and after summarizing can also be found
VI. CONCLUSION

Reducing the amount of time you spend reading can significantly increase your productivity. Python and SpaCy's natural language processing capabilities can help you save time without compromising the accuracy of the content you read, whether it be papers or academic journals. This is merely one of the techniques for producing text summaries by figuring out the key phrases utilising the key words. N-grams, a part of speech tagger, and the nltk library are further options for performing lexical analysis. We plan to keep up with and develop these packages as more resources become available.

FUTURE SCOPE

The proposed work does not include a notebook-wide summarizer. With potential future effort, we may perhaps improve the summarizer's quality and make it more effective[20].

ACKNOWLEDGEMENT

"This work is supported by the Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women, Bhimavaram, India."

DECLARATION

| Funding/Grants/Financial Support | No, I did not receive. |
| Conflicts of Interest/Competing Interests | No conflicts of interest to the best of our knowledge. |
| Ethical Approval and Consent to Participate | No, the article does not require ethical approval and consent to participate with evidence. |
| Availability of Data and Material/Data Access Statement | Not relevant. |
| Authors Contributions | All authors have equal participation in this article. |

REFERENCES


AUTHORS PROFILE

Dr. Dharmaiah Devarapalli is working as a Professor, Computer Science and Engineering at Shri Vishnu Engineering College for Women(A), Bhimavaram, Andhra Pradesh, India-534202. He has completed his Ph.D. from Acharya Nagarjuna University, Guntur, AP, India, in 2014. His area of interest is Computational Intelligence, Deep learning, Image Analytics, Cyber Security, Bioinformatics, and IoT. He published more than 30 publications in reputed journals and 2 Indian patents.

Ms. Srija Padmini Guduri is an undergraduate student in Computer Science and Engineering at Shri Vishnu Engineering College for Women(A), Bhimavaram, Andhra Pradesh, India-534202. Her area of interest is Machine Learning, Deep learning, Data science, and Artificial Intelligence. She published one publication in reputed journals.

Ms. Pericharla Jaya Madhuri is an undergraduate student in Computer Science and Engineering at Shri Vishnu Engineering College for Women(A), Bhimavaram, Andhra Pradesh, India-534202. Her area of interest is Machine Learning, Deep learning, Data science, and Artificial Intelligence. She published one publication in reputed journals.

Ms. Sathi Navya Vahini Reddy is an undergraduate student in Computer Science and Engineering at Shri Vishnu Engineering College for Women(A), Bhimavaram, Andhra Pradesh, India-534202. Her area of interest is Machine Learning, Deep learning, Data science, and Artificial Intelligence. She published one publication in reputed journals.

Ms. Pavani Pasupuleti is an undergraduate student in Computer Science and Engineering at Shri Vishnu Engineering College for Women(A), Bhimavaram, Andhra Pradesh, India-534202. Her area of interest is Machine Learning, Deep learning, Data science, and Artificial Intelligence. She published one publication in reputed journals.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Lattice Science Publication (LSP) / journal and/or the editor(s). The Lattice Science Publication (LSP) / journal and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.