Indian Journal of Artificial Intelligence and Neural Networking (IJAINN)
ISSN: 2582-7626 (Online), Volume-2 Issue-1 December 2021

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 summarizing systems can now quickly retrieve summary phrases from input documents. However, it has 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 to 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 summarizing uses 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.

Manuscript received on 11 October 2021 | Revised Manuscript received on 20 November 2021 | Manuscript Accepted on 15 December 2021 | Manuscript published on 30 December 2021.
*****Correspondence Author(s)
Dr. Dharmaiah Devrapalli*, Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women(A), Bhimavaram (A.P), India. E-mail: devarapalli.dharma@gmail.com, ORCID ID: https://orcid.org/0000-0002-5804-5880.
Srija Padmini Guduri, Department of Computer Science and Engineering, Shri Vishnu Engineering College for women, Bhimavaram (A.P), India. E-mail: guduri.srija@gmail.com, ORCID ID: https://orcid.org/0009-0000-3653-1469
Pericharla Jaya Madhuri, Department of Computer Science and Engineering, Shri Vishnu Engineering College For Women, Bhimavaram (A.P), India. E-mail: jayamadhuri345@gmail.com, ORCID ID: https://orcid.org/0009-0005-8547-7928
Sathi Navya Vahini Reddy, Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women, Bhimavaram (A.P), India. E-mail: navyasathi444@gmail.com, ORCID ID: https://orcid.org/0009-0009-7846-8854
Pavani Pasupuleti, Department of Computer Science and Engineering, Shri Vishnu Engineering College for women, Bhimavaram (A.P), India. E-mail: pavanipasupuleti9@gmail.com, ORCID ID: https://orcid.org/0009-0005-4044-7960

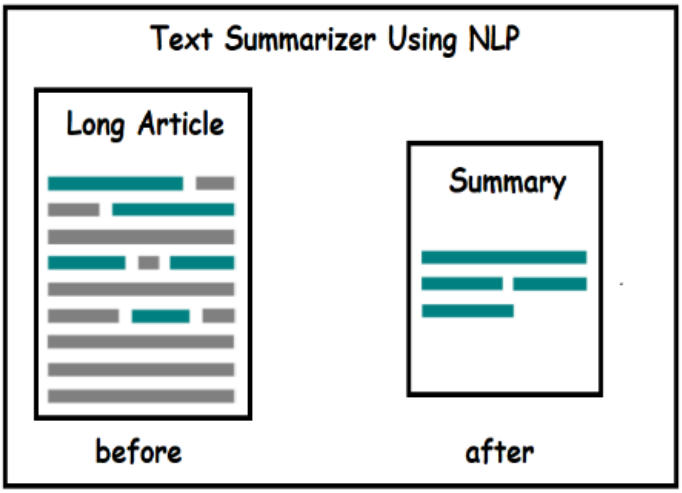

Fig. 1. Text To Summary

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.

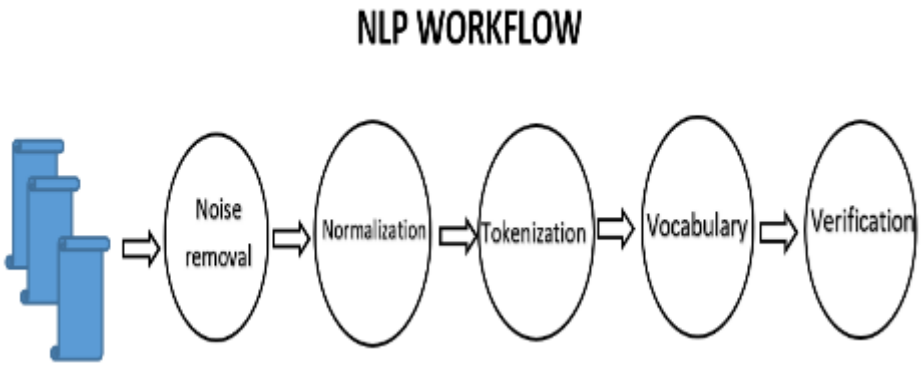

Fig. 2. NLP Workflow

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].

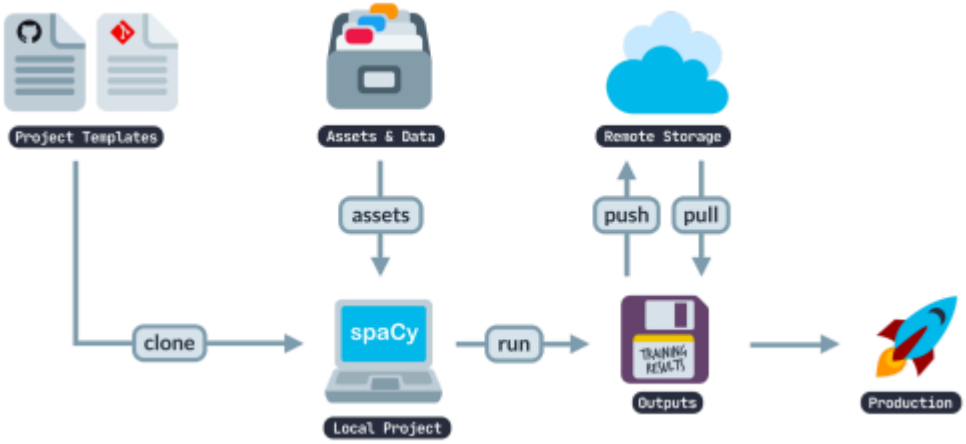

Fig. 3. spaCy

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

Retrieval Number: 100.1/ijainn.B38731212222
DOI: 10.54105/ijainn.B3873.122121
Journal Website: www.ijainn.latticescipub.com

Published By:
Lattice Science Publication (LSP)

0. [14] 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−Pprocessing 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].

Fig. 4. Tokenization

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].

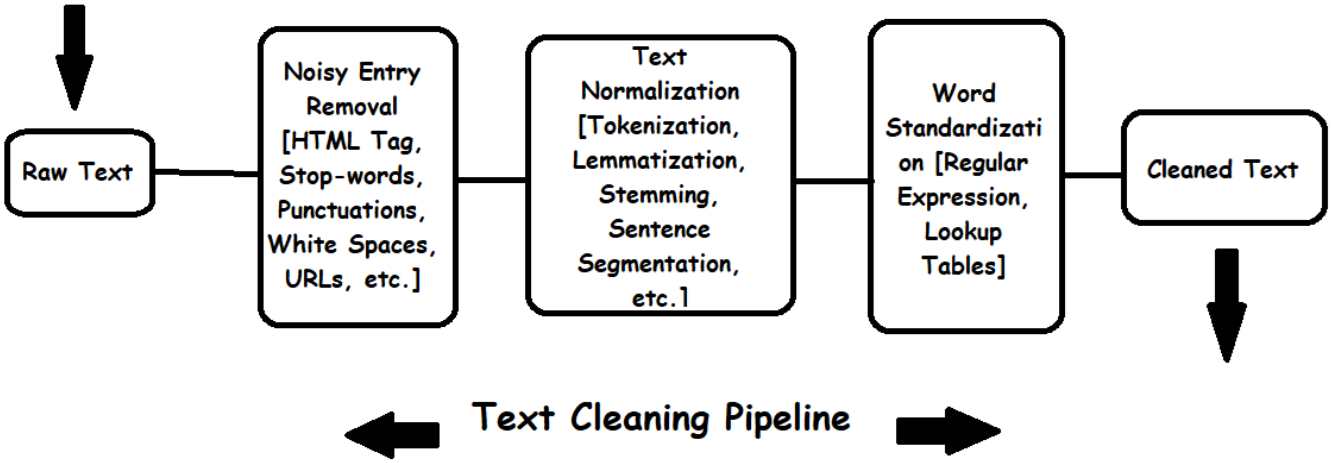

Fig. 5. Text cleaning

2) Tokenization of Words: Tokenize each Word in Sentences [7].

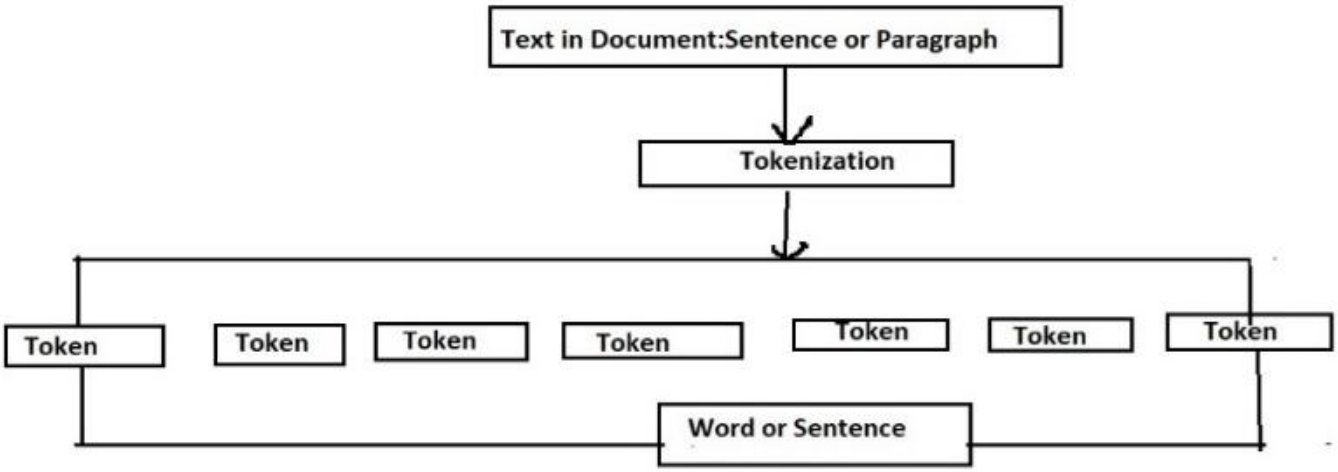

Fig. 6. Word Tokenization

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]

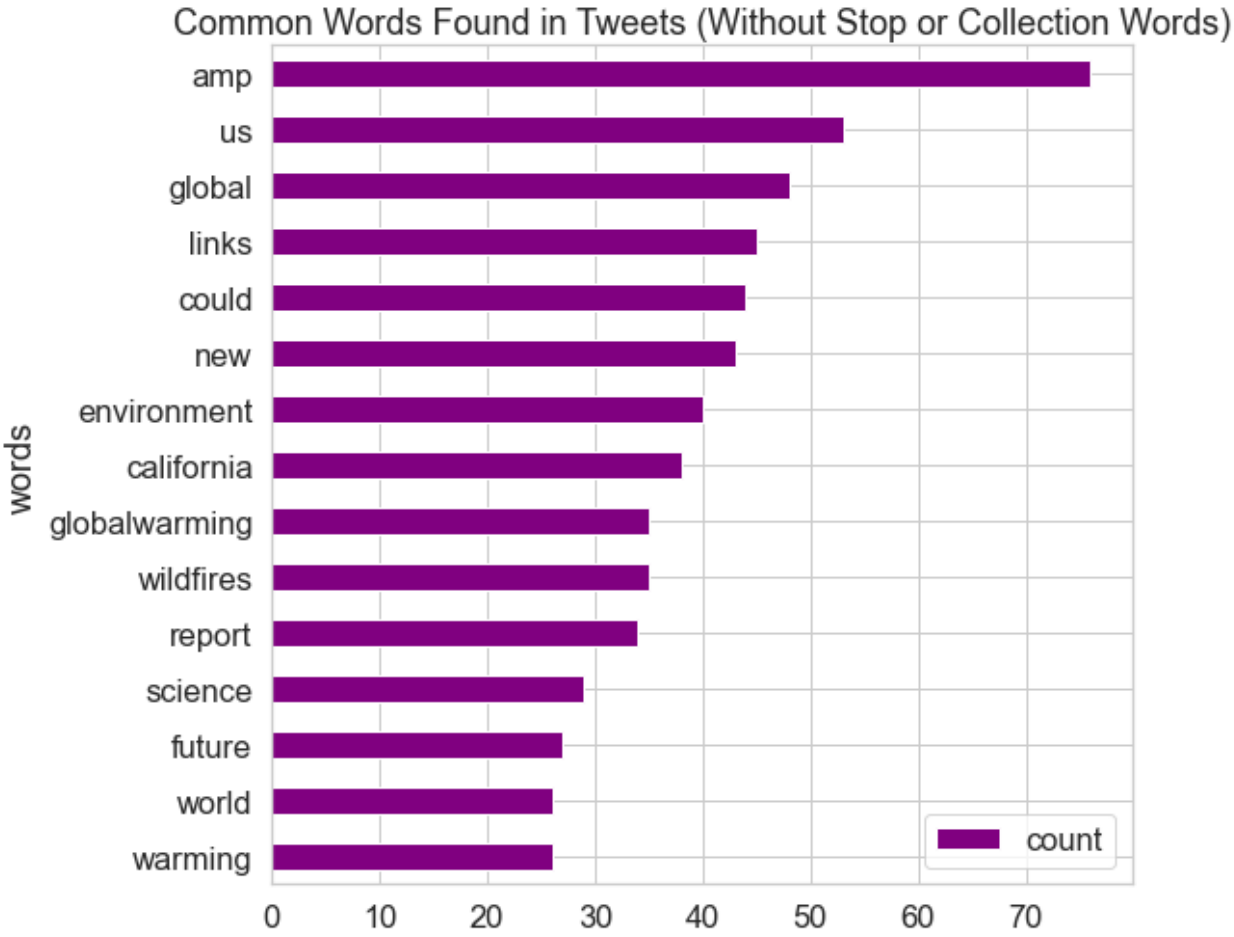

Fig. 7. Word Frequencies

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.

Retrieval Number: 100.1/ijainn.B38731212222
DOI: 10.54105/ijainn.B3873.122121
Journal Website: www.ijainn.latticescipub.com

Published By:
Lattice Science Publication (LSP)

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 normalised 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

Fig. 8. Initial web page

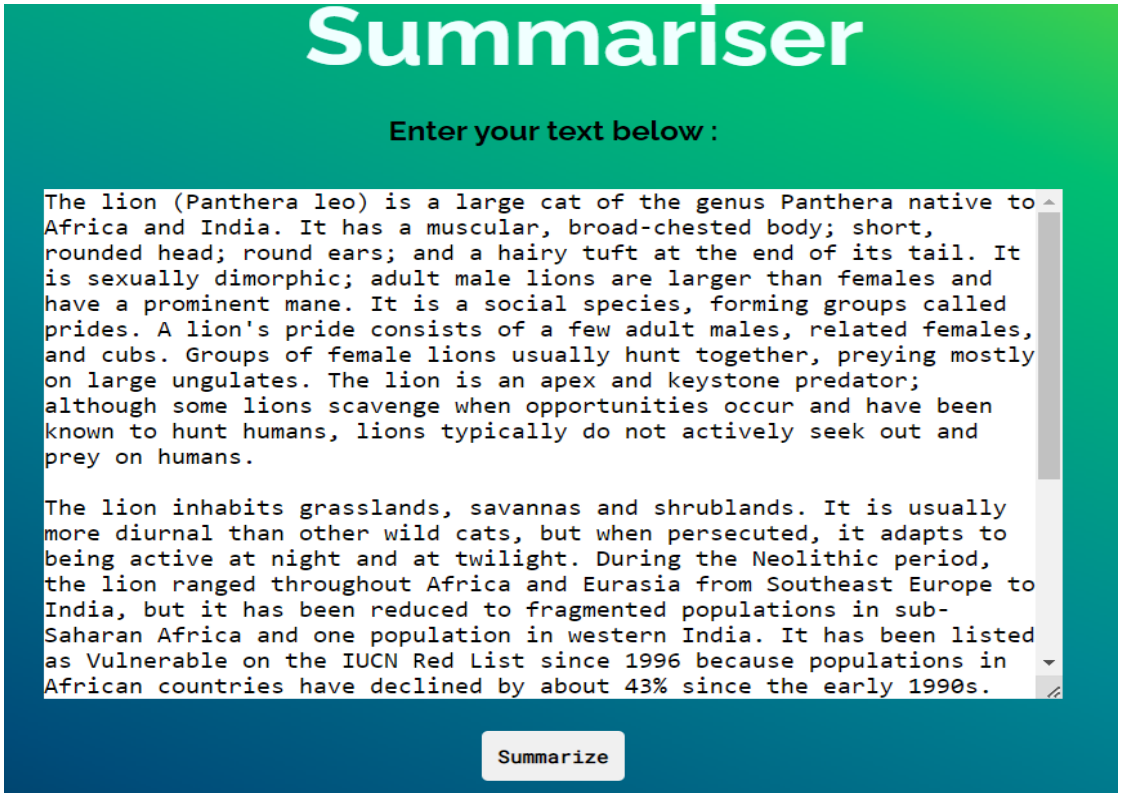

Fig. 9. Entering text into it

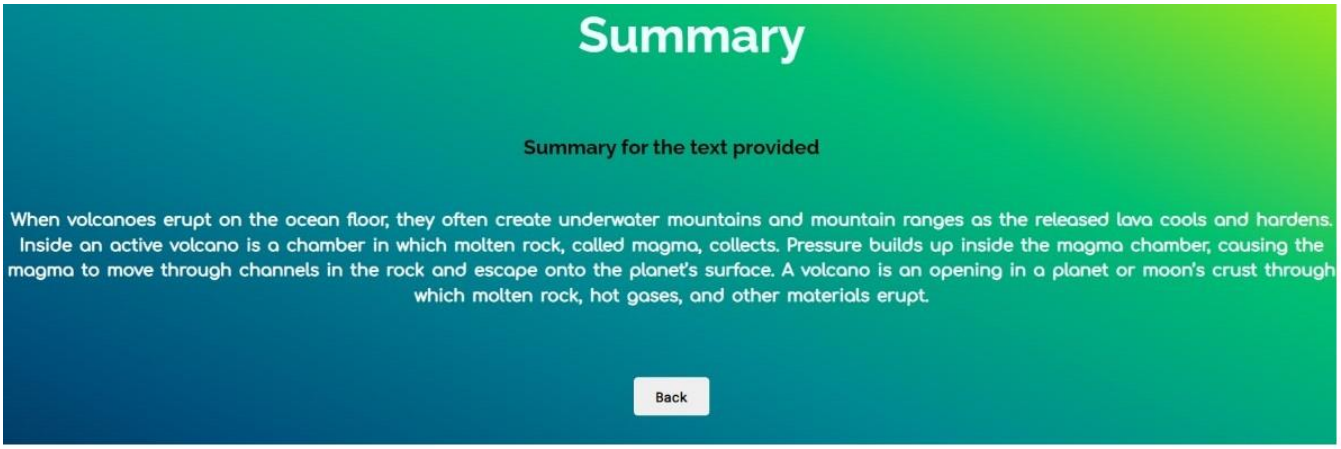

Fig. 10. Summary Obtained

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

1. Deepa, R., Konshi, J., Haritha, A. and Shobini, K. (n.d.). Automatic Text Summarization System. [online] Available at: https://www.ripublication.com/ijaerspl2019/ijaerv14n5spl_04.pdf [Accessed 19 Jun. 2022].
2. Johnson, M.E. (2018). Automatic Summarization of Natural Language. arXiv:1812.10549 [cs, stat]. [online] Available at: https://arxiv.org/abs/1812.10549 [Accessed 30 Jun. 2022].
3. .Maybury, M. (1999). Advances in Automatic Text Summarization. [online] Google Books. MIT Press. Available at: https://books.google.co.in/books?hl=en&lr=&id=YtUZQaKDmzEC&oi=fnd&pg=PA81&dq=EduardHovyand Chin Yew Lin.Automated+text+summarization in SUMMARIST. MIT Press [Accessed 30 Jun. 2022].
4. .Mahdipour, E. (2014). Automatic Persian Text Summarizer Using Simulated Annealing and Genetic Algorithm. International Journal of Intelligent Information Systems, 3(6), p.84. doi:10.11648/j.ijiis.s.2014030601.26. [CrossRef]
5. Srikanth, P. and Deverapalli, D. (2017). CFTDISM:Clustering Financial Text Documents Using Improved Similarity Measure. [online] IEEE Xplore. doi:10.1109/ICCIC.2017.8524466. [CrossRef]
6. Vipul Dalal Latesh L. G. Malik Semantic Graph Based Automatic Text Summarization for Hindi Documents Using Particle Swarm Optimization. Smart Innovation, Systems and Technologies book series (SIST,volume 84).
7. .Kupiec, J.M. and Schuetze, H. (n.d.). System for genre-specific summarization of documents. [online] Available at: https://patents.google.com/patent/US6766287B1/en [Accessed 30 Jun. 2022].
8. Doran, W., Stokes, N., Carthy, J. and Dunnion, J. (n.d.). Comparing Lexical Chain-based Summarisation Approaches Using an Extrinsic Evaluation. [online] Available at: http://www.oriyana.cz/id32402/jazyk/jazykove(2da/aplikovana(1_lingvistika/Ontologie/WordNet/Conference_2004/103.pdf [Accessed 30 Jun. 2022].
9. Goldstein, J., Kantrowitz, M., Mittal, V. and Carbonell, J. (1999). Summarizing text documents. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99. doi:10.1145/312624.312665. [CrossRef]
10. Luhn, H.P. (1958). The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, [online] 2(2), pp.159–165. doi:10.1147/rd.22.0159. [CrossRef]
11. 11.Kedar Bellare, Anish Das Sharma, Atish Das Sharma, Navneet Loiwal and Pushpak Bhattacharyya.Generic Text Summarization Using Wordnet. Language Resources Engineering Conference (LREC 2004), Barcelona, May, 2004.
12. Satapathy, S.C. and Joshi, A. (2017). Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 2. [online] Google Books. Springer. [CrossRef]
13. Deepa, R., Konshi, J., Haritha, A. and Shobini, K. (n.d.). Automatic Text Summarization System. [online] Available at: https://www.ripublication.com/ijaerspl2019/ijaerv14n5spl_04.pdf [Accessed 19 Jun. 2022].
14. A. Nenkova and K. McKeown, "A survey of text summarization tech-niques," in Mining text data. Springer, 2012, pp. 43–76. [CrossRef]
15. D. Sakhare, R. Kumar, and S. Janmeda, "Development of embed-ded platform for sanskrit grammar-based document summarization," inSpeech and Language Processing for Human-Machine Communications.Springer, 2018, pp. 41–50. [CrossRef]

16. X. Mao, H. Yang, S. Huang, Y. Liu, and R. Li, "Extractive summarization using supervised and unsupervised learning," Expert Systems with Applications, vol. 133, pp. 173–181, 2019. [CrossRef]
17. R. Z. Al-Abdallah and A. T. Al-Taani, "Arabic single-document text summarization using particle swarm optimization algorithm," Procedia Computer Science, vol. 117, pp. 30–37, 2017. [CrossRef]
18. P. B. Baxendale, "Machine-made index for technical literature—an experiment," IBM Journal of Research and Development, vol. 2, no. 4. [CrossRef]
19. R. Oak, "Extractive techniques for automatic document summarization:a survey," International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 3. [CrossRef]
20. S. Chitrakala, N. Moratanch, B. Ramya, C. R. Raaj, and B. Divya,"Concept-based extractive text summarization using graph modelling and weighted iterative ranking," in International Conference on Emerging Research in Computing, Information, Communication and Applications.

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 30 publications in repute 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, Andra 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, Andra 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, Andra 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, Andra Pradesh, India-534202. Her area of interest is Machine Learning, Deep learning, Data science, and Artificial Intelligence. She published one publication in reputed journals.