Novel Research Paper Summariser Based on Deep Learning: Proof of Concept

Suja Sreejith Panickar, Prabal Singh, Saurav Bhise, Devanshu Mehta, Pratham Pokharkar, Zuhayr Khalid

Dept. of CET, Dr. Vishwanath Karad, MIT World Peace University, Pune

Abstract- The exponential rise of textual data in the modern digital era has made it difficult to efficiently understand and extract crucial information. It is necessary to use a method that is more accurate and efficient because manual summarizing takes time and is prone to mistakes. In order to overcome these issues, this research article examines the possibility of automated text summarizing technology. Automated summarizers provide a more effective and precise method for extracting key information from lengthy texts by utilizing natural language processing techniques and machine learning algorithms. This study aims to look into the use of automated summarization in the context of the growing amount of textual data. Reviewing new research on automatic text summarization methods like transformers and pre-trained encoders is part of the strategy. The results show how efficient automatic summarizing is at lowering storage needs and giving useful information for text mining and data analysis activities.

Keywords-*Automated Text Summarization, Natural Language Processing, Pre-trained Transformers*

1. Introduction

The amount of textual data being produced in the digital age has increased exponentially, making it extremely difficult for people to understand it all and effectively extract the information they need. Even while it is successful, manual summarizing requires a lot of work and is prone to mistakes. This highlights the requirement for automated summarizing technology to offer a more effective and precise method. Furthermore, the expanding data volume has led to storage issues, which automated summarization can resolve by minimizing file sizes. Automated summarizing is a useful tool for academics and analysts alike since accurate and succinct summaries provide insightful data and text mining.

1.1. Motivation

The overwhelming volume of textual data generated each day, which makes it more and more challenging for people to understand and extract crucial information, is the driving force behind this study article. The sheer amount of data cannot be handled by manual summarization, which is time-consuming, error-prone, and labour-intensive. The necessity for automated summarizing technology develops in order to get over these restrictions and enable effective information processing.

1.2. Need of the Study

The increasing amount of textual data as well as the shortcomings of manual summarizing make it clear that automated text summarization is necessary. The development of automated summarization techniques during the last three to four years, including transformers and pre-trained encoders, has been highlighted in recent literature and has shown encouraging results.

1.3. Research Objectives

This study's goals are divided into two categories. In order to overcome the difficulties caused by the growing amount of textual data, it is first necessary to investigate the possibility of automated summarization. Reduced storage needs and the availability of succinct, accurate summaries are two examples of this. The second goal is to emphasize the importance of automated summarization in text mining and data analysis jobs, which enables researchers and analysts to quickly get insights and make wise decisions. These goals will help this research article enhance automatic text summarizing techniques and their use in a variety of fields.

1.4. Organization of the Paper

The structure of the paper is as follows. An overview of the available research on automatic text summarization is given in Section 2. The research's methodology and data are covered in Section 3 of the article. The results and analysis of the experiments are presented in Section 4. The research's usefulness in the context of text mining and data analysis is discussed in Section 5 along with its practical ramifications. The paper is concluded in Section 6 along with a description of possible future research topics.

2. Literature Survey

2.1. Abstractive Summarization

In [1] recurrent attentive summarizer (RAS) is a model for abstractive phrase summarising that employs a conditional recurrent neural network (RNN), more precisely an RNN-LSTM. The RAS builds on the previous feedforward neural network model in [2] by taking into account input word position information and using prior words and sentences during training to construct the next word in the summary. Both models were trained on datasets of sentence-summary pairs.

Using an encoder-decoder RNN, Lopyrev [3] developed a simpler attention technique for producing headlines for news stories. There were two types of attention processes tested: basic and sophisticated. The last layer of the encoding and decoding in the basic mechanism was separated into two pieces for computing the attention weight vector and context vector, respectively. No such divide was formed in the complicated mechanism, and the attention weight vector and context vector were computed in the final layer without fragmentation. During testing, a beam search was used to lengthen the probability sequence.

[4], like articles [5] and [6], uses encoder-decoder RNN and sequence-to-sequence models to map input to target sequences. Three approaches for global attention are proposed: dot product scoring, bilinear form, and scalar value determined from the RNN encoder's hidden states projection. The research employs LSTM cells rather than GRU cells and investigates three models with distinct topologies.

The first model employs unidirectional LSTMs in both the encoder and decoder, the second employs bidirectional LSTMs in both the encoder and the decoder, and the third employs a bidirectional LSTM encoder and an LSTM decoder with global attention. The context vector created at each time step in the encoder-decoder model is determined by whether the attention mechanism is local or global. The preceding decoder output is included in the decoder input. The output of the LSTM decoder is converted to a dense vector prediction via an affine transformation. The study discusses the training time required to match the number of hidden states to the vocabulary size.

Khandelwal [6] performs abstractive summarization on tiny datasets using a sequence-to-sequence model with an LSTM encoder and decoder. After receiving the encoder's hidden representations and transferring them to the SoftMax layer, the decoder creates the output summary. According to the article, sequence-to-sequence models do not memorise information, limiting generalisation. To overcome this issue, the study recommends utilising imitation learning to determine whether to use the golden token (i.e., reference summary token) or previously created output at each stage.

The RC-Transformer (RCT) was used to solve the long-sequence inadequate semantic representation of abstractive text summarisation systems based on an RNN encoder-decoder framework [7]. To solve the issue of a lack of sequential information at the word level, which is critical for abstractive text summarisation, the RCT-Transformer utilises two encoders. According to Cai et al., the model uses a beam search at the decoder and is quicker than RNN-based models. Additional benefits include the transformer's parallel processing capability and ability to obtain global context semantic links.

Paper [8] proposes a two-stage fine-tuning strategy in which they fine-tune the encoder first on the extractive summarising problem and subsequently on the abstractive summarization task. Previous research (Gehrmann et al., 2018; Li et al., 2018) demonstrates that applying extractive aims might improve abstractive summarization performance. Also, because this two-stage strategy is theoretically quite basic, the model may benefit from knowledge shared between these two jobs without fundamentally changing its design.

2.2. Dataset Pre-processing

Datasets were pre-processed by PTB tokenization in the model provided by Rush et al., with "#" replacing all digits, all letters converted to lowercase letters, and "UNK" replacing words that appeared less than 5 times [2]. Due to a lack of restrictions for producing output, the model was trained using any input-output pair. The Gigaword datasets were used for training, whereas DUC2003 and DUC2004 were used for summarisation assessment [2]. Chopra et al.'s suggested model was trained on the Gigaword corpus using sentence separation and tokenization [9]. Each article's title was coupled with the first sentence to produce sentence-summary pairs. The same data preparation techniques as in [2] were used in [9].

After analysing the data, Lopyrev and Jobson et al. used Gigaword to train the model. Tokenisation and character conversion to lowercase were the most important pre-processing processes in the Lopyrev model for both the text and the headline [3]. Furthermore, just the first paragraph's characters were maintained, and the headline's length was set between 25 and 50 words. Furthermore, no-headline articles were ignored, and the symbol was utilised to substitute uncommon terms.

2.3. Extractive Summarization

The method used in paper [8] trained extractive models using a greedy technique similar to Nallapati et al. (2017) to produce an oracle summary for each page. The programme provides an oracle composed of many words that maximises the ROUGE-2 score in comparison to the gold summary. It utilises the model to acquire the score for each phrase before predicting summaries for a fresh document. Then, in order of greatest to lowest score, rate these sentences, and choose the top three sentences as the summary.

3. Methodology

3.1. Approaches to Text Summarization

To create a summary from a lengthy text, two methods are used: abstractive summarization and extractive summarization.

Extractive summarization: Extractive summarization is picking out key words or sentences from the source text and combining them to create a summary. The most pertinent sentences, which are typically those that contain keywords or phrases that are frequently used in the text, are found using statistical or rule-based algorithms in this technique. Extractive summarization, which is frequently employed in news items and academic publications, can result in a summary that is strikingly identical to the original text.

Abstractive summarization: Abstractive summarizing entails creating a summary that can include brand-new words or phrases that are absent from the original text. This method uses deep learning techniques for natural language processing (NLP) to provide a summary that accurately captures the meaning of the original text. Because abstractive summarizing requires the algorithm to comprehend the context and meaning of the original text, which can be difficult for machines to do, it is more difficult than extractive summarization.

3.2. Architecture

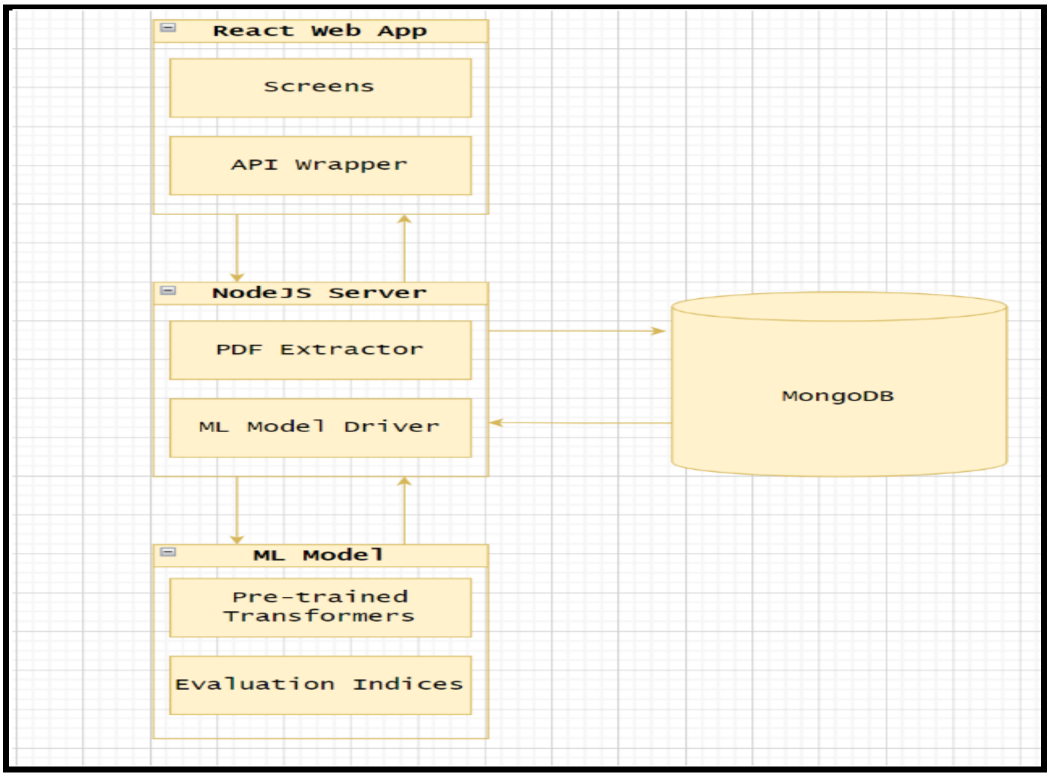


Figure 1. Low Level Design

Our basic design shows a simple web interface connected to a server which is connected to a database. The server interacts with the ML model to receive inputs and generate the summarized text as its output.

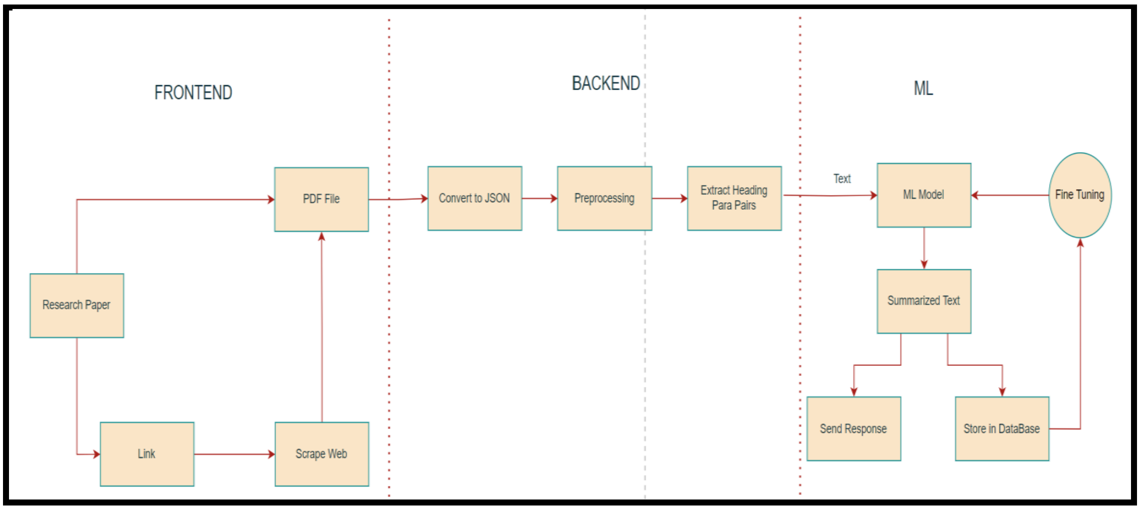


Figure 2. Activity Diagram

Data Collection: First, a study paper is collected, which may be received in two forms: a web link and a PDF file. The web link is used to scrape the web and get the PDF file of the paper.

Front-end UI: The PDF file is uploaded to the web application along with some information about the domain of the paper being uploaded. The file is then sent to the backend for processing.

Pre-processing: The PDF is received from the front-end and converted to an HTML format. The text from the PDF is also extracted and converted into a JSON format. To prepare the data for future analysis, the transformed HTML and JSON files are pre-processed. This stage may include cleaning the data, deleting superfluous pieces, and organising the content.

Heading-Paragraph Pair Extraction: The pre-processed data is utilised to extract heading-paragraph pairings. This phase entails locating the document's headers or sections and extracting the paragraphs related with each heading.

Machine Learning (ML) Model: The intermediate stage of processed JSONis passed as an input, which is further decomposed and processed into more accurate format, eliminating any anomalies of paragraphs or false text flagged as topic. Upon completion of Final text pre-processing, the text is furthermore decomposed into tokens and sentence embeddings, and to deal with the problem of large bodies of text, extractive summarisation is performed deploying the tf-idf method for sentence ranking in document and subsequently in corpus to generate important processable piece of text from the larger content, which can further be transformed into a sentence embedding or processed as an input to the transformer model for further accurate summarisation. This ML model analyses the text and creates a content summary.

The created summary is returned to the front-end, where it may be displayed or utilised for additional processing.

Database Storage: The result of the summarization is saved in a database for future reference. training or retrieval.

Fine-tuning: The summary data that has been saved may be utilised to fine-tune the ML model with domain specific literature to develop more precise summaries for scientific literature rich content. This iterative technique seeks to enhance the model's performance by incorporating input and modifying its parameters.

NLP has hugely contributed to variety of applications worldwide [13-16], to mention few are – social media analytics [17-20], healthcare analytics [21-23], image sentiments analytics [24-26], education [27-28].

3.3. Dataset

BBC Dataset [12]: The BBC dataset is a collection of news stories from the British Broadcasting Corporation (BBC). It comprises of news items from a range of topics, including politics, business, entertainment, sports, and technology. Every article in the collection typically has a headline, a summary or introduction, the main text of the piece, and extra metadata like the publishing date. The dataset offers researchers a wide variety of news stories to work with and is frequently used for tasks including text categorization, topic modelling, and text summarization.

ScisummNet dataset [11]: The ScisummNet dataset consists of scientific paper summaries matched with full-text articles. It is utilised for training and testing text summarization models. Researchers may use this information to create algorithms that provide brief and useful summaries, assisting in literature review and keeping up with scientific progress. It permits the investigation of domain-specific issues and approaches for scientific text summarising.

3.4. Pre-processing

3.4.1. Cleaning Text

Removal of special characters and punctuation: This stage entails getting rid of any extra special characters and punctuation that don't add anything to the text's meaning.

Lowercasing: By changing all text to lowercase, you may normalize the data and prevent duplicate entries based on case.

3.4.2. Tokenization

Dividing the text into separate tokens or words: Tokenization separates the text into digestible chunks, like words or sub words, which can then be processed and examined separately.

3.4.3. Stop Word Elimination

Elimination of common terms Stop words are frequently used terms that have little sense when analysed within the context of text. Eliminating them decreases noise and boosts processing effectiveness.

3.4.4. Stemming or lemmatization

Words that have been inflected or derived are reduced to their basic form through lemmatization or stemming, which enables more accurate analysis and grouping of related words.

3.4.5. Content Removal from the Site

Filtering away information that is not informative: In this phase, you should get rid of everything that does not add anything to the meaning of the text, including HTML tags, URLs, numerals, or other elements.

3.4.6. Sentence Segmentation

Separating the text into sentences: Sentence segmentation separates the text into sentences, allowing for a more detailed analysis and the extraction of significant information at the sentence level.

3.5. Models

3.5.1. BART (Bidirectional and Auto-Regressive Transformers)

Text summarization is just one of the many natural language processing (NLP) tasks that BART, a transformer-based approach, is frequently employed for. In order to accurately capture the context and produce coherent and illuminating summaries, it integrates both bidirectional and auto-regressive components. By conditioning the model on the input text and refining it to provide succinct and coherent summaries, BART can be fine-tuned on summarization datasets to produce abstractive summaries.

3.5.2. T5 (Text-To-Text Transfer Transformer)

A flexible transformer-based model called T5 was created by Google. It was developed using a "text-to-text" paradigm that allows a variety of NLP tasks, such as text summarization, to be recast as text production tasks. T5 is capable of producing abstractive summaries and has attained state-of-the-art performance on a number of benchmarks. For researchers and practitioners working on text summarizing problems, T5 is a popular option due to its adaptability and potent performance.

3.5.3. PEGASUS

PEGASUS is a transformer-based approach for abstractive text summarization created by Google Research. It uses a framework for self-supervised learning that combines pre-training and fine-tuning. PEGASUS uses a variety of document-level and sentence-level reconstruction aims and is trained on large-scale datasets to produce high-quality summaries. It can produce coherent and illuminating summaries across a variety of disciplines and has achieved outstanding performance on common benchmarks for summarizing.

4. Applications of the model

In the research work on automated text summarization utilizing the BBC and ScisummNet datasets, the models BART, T5, and PEGASUS will be used to produce abstractive summaries of news stories.

On the combined dataset of BBC and ScisummNet dataset, BART, T5, or PEGASUS can be fine-tuned until the models learn to comprehend the context and key information inside the articles and produce succinct and insightful summaries. These summaries can retain coherence and readability while capturing the essential concepts, prominent details, and important information from the source text.

The use of these models can be advantageous to numerous parties, including information seekers, journalists, and researchers, in a number of ways:

- Review of Literature: Review several research papers efficiently.
- Document Indexing and Search: Use summaries to improve document retrieval.
- Knowledge Discovery and Mining: Extract useful information from enormous amounts of paper.
- Trend Analysis: Monitor trends and developments in specialised sectors.
- Personalised Research Alerts: Stay up to speed on current research in certain fields.
- Summaries can be used as teaching materials in the classroom.
- When BART, T5, or PEGASUS are used in the context of the study paper, accurate and succinct summaries can be automatically generated from datasets from the BBC and CNN, offering useful tools for information processing, knowledge extraction, and effective news consumption.

5. Results and Analysis

A sample research paper [10] was used for testing the efficiency of our model.

Initially, each section of the research paper is treated as a key and its adjoining text/data is treated as a value. Therefore, creating a key value pair.

Figure 3 is the page which greets the user upon opening. The user can upload a pdf file of their choice, and select the corresponding fields provided that gives context to the research paper uploaded.

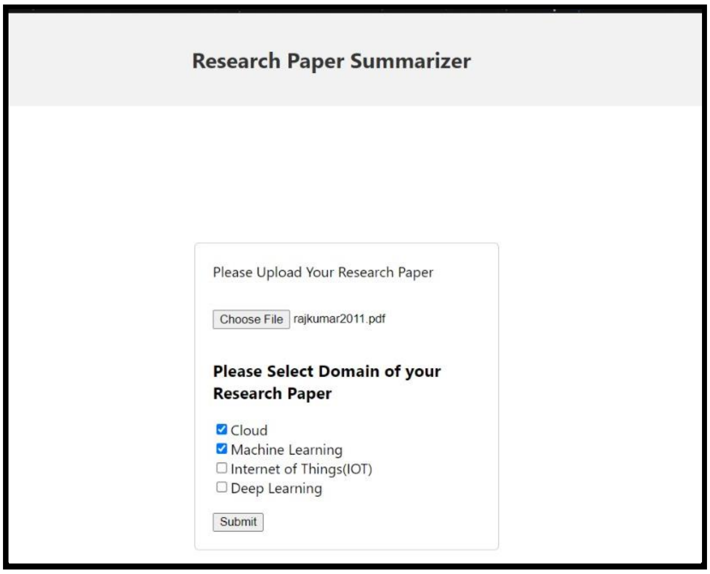


Figure 3. The initial page of the web interface

The following summary was obtained after running the model:

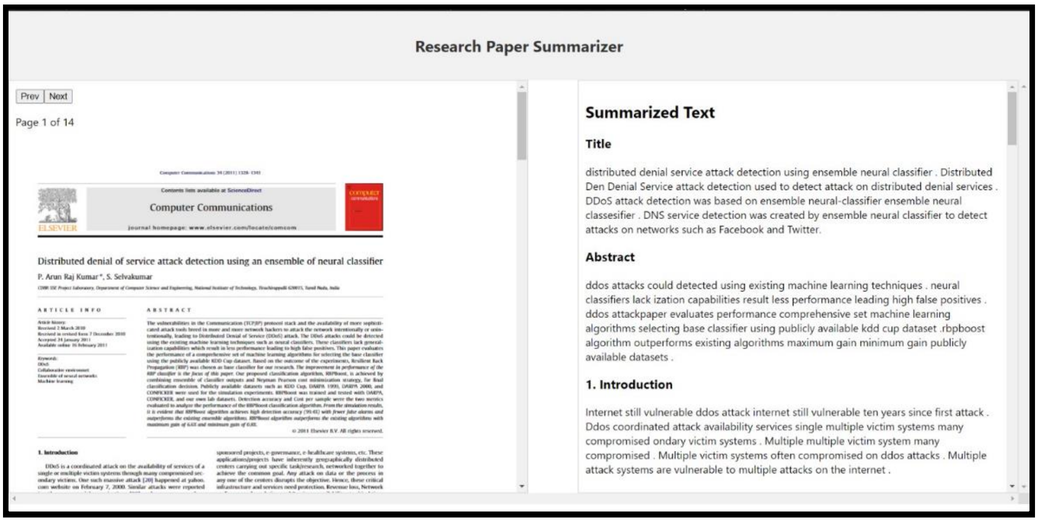
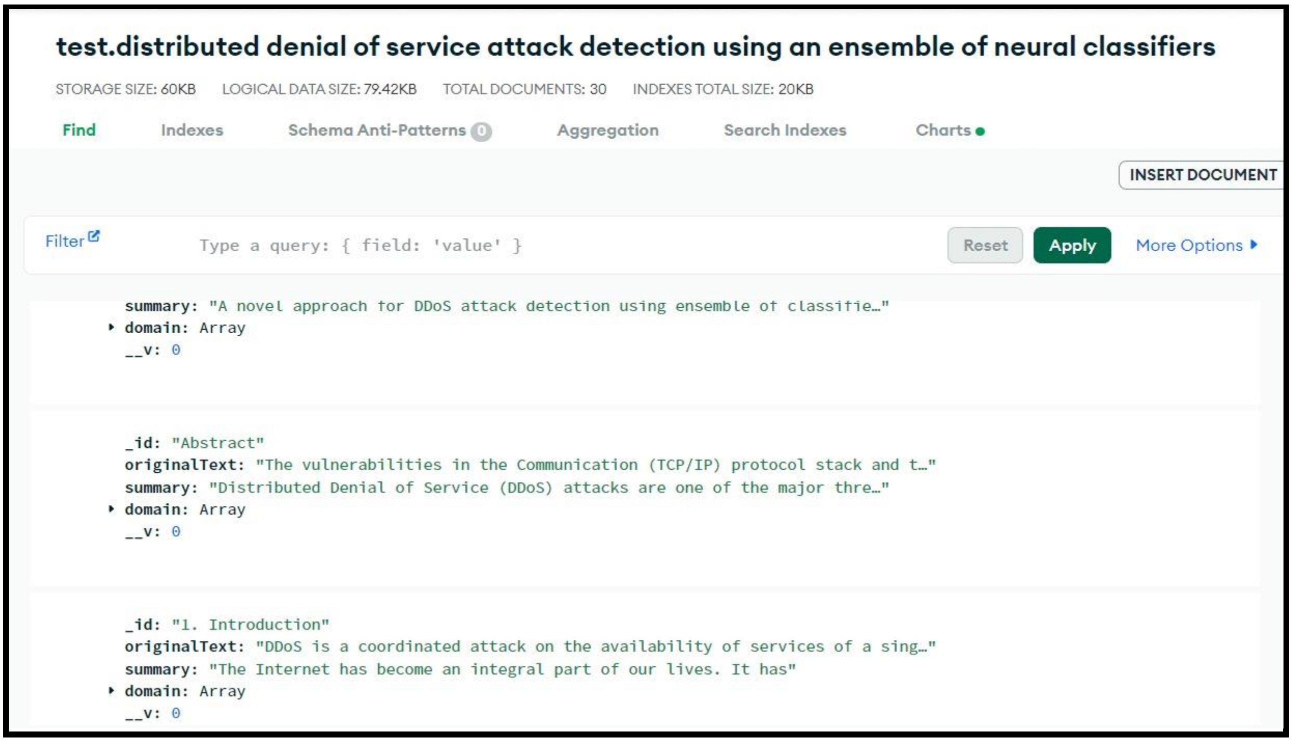


Figure 4. The output on the web interface

We are creating a database to finetune



6. Conclusion

The research paper illustrates the potential of automated text summarization methods, including transformers and pre-trained encoders, to deal with the problems brought on by the growing amount of textual data. With the use of these methods, you may extract the most important data and create succinct summaries more precisely and efficiently. Reduced storage needs, improved information processing, and easier text mining and data analysis activities are just a few examples of how automated summarization has practical applications. These techniques offer insightful information for researchers and analysts by utilizing natural language processing and machine learning.

7. Limitations and Future Scope

- Loss of complicated information: The existing automated text summarizing algorithms could have trouble effectively capturing and portraying complex information throughout the summation process, which could lead to the loss of crucial specifics or nuances.
- Difficulty managing topic-related texts: It might be difficult for automatic summarizing systems to manage texts that are precisely relevant to a given topic or domain. It may be necessary to make future improvements to the capability of extracting and summarizing important information from texts that are specific to a given topic.
- Lack of scientific domain-specific datasets: Accessing or making use of specialized datasets that are particular to scientific fields may be difficult for research paper. The effectiveness and generalizability of the model in summarizing technical or scholarly papers may be impacted by this constraint.
- Work required for annotation and data augmentation: The datasets needed for training and fine-tuning the summarization models require a significant amount of annotation and data augmentation work. This process requires a lot of labour, which can be time- and resource-intensive.

- Bias and ethical concerns: Automated summarization systems may be biased by design, which raises issues that need to be looked into more and prevented. Automated text summary has the potential for biased summaries, and it has ethical issues that call for careful consideration and mitigating measures.

These drawbacks highlight the necessity for domain-specific datasets, the difficulties in processing complex data, the need for fine-tuning methods, and ethical issues. These are all areas that need more investigation and development in automated text summarization.

Future Scope

The stated shortcomings and the development of the area can be the main goals of future study in automated text summarization. This entails developing methods for maintaining more complex context and information in the summaries, enhancing the processing of texts relevant to a certain area, and improving the customizability and adaptability of summarizing models. Automated summarization's precision and efficacy could be increased by investigating cutting-edge methods like utilizing contextual embeddings or reinforcement learning. The stability and dependability of automated summarization systems can also be ensured by incorporating human input and evaluation procedures into the development process. To promote the use of automated summarization technology in real-world applications and address the systems' broader societal effects, cooperation between researchers, industry practitioners, and policymakers is required.

References

1. Dima Suleiman, Arafat Awajan, "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges", Mathematical Problems in Engineering, vol. 2020, Article ID 9365340, 29 pages, 2020. https://doi.org/10.1155/2020/9365340
2. A. M. Rush, S. Chopra, and J. Weston, “A neural attention model for abstractive sentence summarization,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 2015.
3. K. Lopyrev, Generating news headlines with recurrent neural networks, 2015, https://arxiv.org/abs/1512.01712.
4. R. Nallapati, B. Zhou, C. N. dos Santos, C. Gulcehre, and B. Xiang, “Abstractive text summarization using sequence-to-sequence RNNs and beyond,” in Proceedings of the CoNLL-16, Berlin, Germany, August 2016.
5. E. Jobson and A. Gutiérrez, Abstractive Text Summarization Using Attentive Sequence-To-Sequence RNNs, p. 8, 2016.
6. U. Khandelwal, P. Qi, and D. Jurafsky, Neural Text Summarization, Stanford University, Stanford, CA, USA, 2016.
7. T. Cai, M. Shen, H. Peng, L. Jiang, and Q. Dai, “Improving transformer with sequential context representations for abstractive text summarization,” in Natural Language Processing and Chinese Computing, J. Tang, M.-Y. Kan, D. Zhao, S. Li, and H. Zan, Eds., pp. 512–524, Springer International Publishing, Cham, Switzerland, 2019.
8. https://doi.org/10.48550/arXiv.1908.08345
9. S. Chopra, M. Auli, and A. M. Rush, “Abstractive sentence summarization with attentive recurrent neural networks,” in Proceedings of the NAACL-HLT16, pp. 93–98, San Diego, CA, USA, 2016.
10. P. Arun Raj Kumar, S. Selvakumar,Distributed denial of service attack detection using an ensemble of neural classifier, Computer Communications, Volume 34, Issue 11, 2011,

Pages 1328-1341, ISSN 0140-3664, https://doi.org/10.1016/j.comcom.2011.01.012. (https://www.sciencedirect.com/science/article/pii/S0140366411000600)

11. https://cs.stanford.edu/~myasu/projects/scisumm_net/
12. https://www.kaggle.com/competitions/learn-ai-bbc/data
13. Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544-551.
14. Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*.
15. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019, June). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)* (pp. 54-59).
16. Blodgett, S. L., Barocas, S., Daumé III, H., & Wallach, H. (2020). Language (technology) is power: A critical survey of" bias" in nlp. *arXiv preprint arXiv:2005.14050*.
17. Kunte, A. V., & Panicker, S. (2019, November). Using textual data for personality prediction: a machine learning approach. In *2019 4th international conference on information systems and computer networks (ISCON)* (pp. 529-533). IEEE.
18. Kunte, A., & Panicker, S. (2020). Personality prediction of social network users using ensemble and xgboost. In *Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2019* (pp. 133-140). Springer Singapore.
19. Ahire, K., Bagul, M., Dhanawate, S., & Panicker, S. S. (2021). A novel proof of concept for Twitter analytics using popular hashtags: experimentation and evaluation. In Proceedings of International Conference on Communication and Artificial Intelligence: ICCAI 2020 (pp. 313-322). Springer Singapore.
20. Kunte, A. V., & Panicker, S. (2019, October). Analysis of machine learning algorithms for predicting personality: brief survey and experimentation. In *2019 global conference for advancement in technology (GCAT)* (pp. 1-5). IEEE.
21. Panicker, S. S., & Gayathri, P. (2019). A survey of machine learning techniques in physiology based mental stress detection systems. *Biocybernetics and Biomedical Engineering*, *39*(2), 444-469.
22. Bharate, M., & Panicker, S. S. (2022). HealthCare Data Analytics: A Machine Learning-Based Perspective. In *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2022* (pp. 333-353). Singapore: Springer Nature Singapore.
23. Khire, S., Ganorkar, P., Apastamb, A., & Panicker, S. (2021). Investigating the Impact of Data Analysis and Classification on Parametric and Nonparametric Machine Learning Techniques: A Proof of Concept. In *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020* (pp. 211-227). Springer Singapore.
24. Panicker, S. (2020, June). Use of machine learning techniques in healthcare: A brief review of cardiovascular disease classification. In *2nd International Conference on Communication & Information Processing (ICCIP)*.
25. Salunke, V., & Panicker, S. S. (2021). Image sentiment analysis using deep learning. In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020* (pp. 143-153). Springer Singapore.
26. Suja Panicker. Intelligent Attention-based transformer models for Text Extraction: A proof of concept. International Conference on Health Informatics, Intelligent Systems, and Networking Technologies (HINT’24) (in press)

27. *Ferreira Mello, R., André, M., Pinheiro, A., Costa, E., & Romero, C. (2019). Text mining in education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(6), e1332.*
28. Chen, X., Xie, H., Zou, D., & Hwang, G. J. (2020). Application and theory gaps during the rise of artificial intelligence in education. Computers and Education: Artificial Intelligence, 1, 100002.