AOI: 10.11234.2024.00019
ACT2024: Special Proceedings of Applied Computer Technology

Model Abstractify AI: Efficient Summarization

Shishir Singh Chauhan, Juhi Singh, Aneerban Saha, K. Krishna Koushika, Akshita Jain

School of Computer Science and Engineering
Manipal University Jaipur – 303007, INDIA

ABSTRACT

In the current environment full of data, the demand for concise and clear summaries of lengthy articles and narratives has become increasingly critical. This document examines the application of the T5 model in generating efficient summaries, drawing upon previous studies conducted in this field. The objective is to enhance the performance of the T5 model by increasing its accuracy, targeting a level of approximately 97% accuracy. The T5 model serves as an advanced resource in the field of Natural Language Processing, facilitating the transformation of intricate data into clear and concise summaries. This study aims to enhance text summarization methodologies, particularly for extended texts, thereby improving content comprehensibility and accessibility. This approach effectively tackles the increasing demand for enhanced information processing capabilities in the digital era.

Keywords: Natural language processing; Text summarization; Abstractive summarization; Extractive summarization; Transformer; T5 Model

I. INTRODUCTION

An Overview of New Developments in the fields of natural language processing includes Autocorrect and Autocomplete, Grammarly, voice typing (speech-to-text), language translation, text summarization, sentiment analysis, voice assistance, etc. [1].

A most intriguing aspect of NLP is its capacity to distill extensive texts into concise, insightful summaries. Envision sifting through extensive data, distilling essential insights, and presenting them succinctly for clarity without superfluous content. The elegance of text summarization lies in its ability to transform our methods of organizing and consuming information [2].

Text summarizers exclude pronouns, verbs, and non-essential grammatical elements, identify keywords, process extensive text segments, and then quantify word frequency. It facilitates the rapid extraction of the most relevant information from data or material. The significance of text summarization is increasing in today's fast-evolving environment [3].

Four fundamental methodologies for text summarization exist: extractive, indicative, informative, and abstractive. This presentation will concentrate only on the novel domain of abstractive text summarization. Extractive approaches depend on pre-existing words and phrases from the original material, while abstractive techniques provide a significant advancement. They include not just extraction but also rephrasing, generating distinctive formulations that encapsulate the essence of the original material while conveying the message more effectively [4].

An abstract text summary includes an executive summary and a short summary of the original content, ensuring the meaning of the compressed material is consistent with the original. It uses another keyword instead of a keyword from the original content. The original content corresponds to phrases and sentences that the algorithm creates. It works just like the human brain, which collects everything semantic data, selects keywords based on semantic similarity, and generates a human-like summary. Abstract summarization can be done using various NLP-based methods and pretrained models. This says the Abstractive approach is better than the Extractive approach [5].

The need for advanced tools to effectively manage and comprehend massive volumes of text-based information has increased due to the quick growth of digital content. When working with big datasets, traditional text processing techniques frequently fail to provide insightful results quickly. NLP (natural language processing) has advanced more quickly because of this problem, especially in text summarization systems. One of the more inventive of these is abstract summarization, which makes it possible to produce succinct, contextually appropriate summaries that go beyond simple extraction to communicate a deeper understanding. The need to improve information consumption in both personal and professional contexts, making data more easily accessible and useful, is what spurred the development of such NLP techniques.

II. LITERATURE REVIEW

Text summarizing is an essential activity in natural language processing (NLP) that involves collecting relevant information from source texts to create logical and concise summaries. Natural language processing problems have recently undergone a transformation thanks to transformer-based models, which offer state-of-the-art performance. This literature study examines relevant studies that have affected text summarization, with a particular emphasis on the T5 (Text-to-Text Transfer Transformer) framework [6].

The assortment of synopsis models distributed every year has been consistently expanding. The accessibility of a lot of information and advancements in brain network models [7-8] are changing from master information and heuristic-based frameworks to information-driven approaches fueled by start to finish profound brain models [18]. Current text summarization methods include graph-based methods that arrange the input text in a graph and then use ranking or graph traversal algorithms to build the summary [10][14], reinforcement learning strategies [9] hybrid extractive-abstractive models, advanced attention and copying mechanisms perform various tasks and multi-reward preparing procedures [12-13] and mixture techniques [17].

In the semi-supervised technique, a probabilistic SVM and a Naïve Bayesian classifier are co-trained to exploit unlabeled data, whereas in the supervised approach, a Support Vector Machine (SVM) algorithm is employed. similarly employed the SVM technique to find pertinent data to include in a query-focused summary. The training set includes structural, cohesion-based, and query-dependent features [11].

One advantage of using machine learning for TS is that it makes testing the performance of many features, like lexical, syntactic, statistical, etc., straightforward. The most suitable features are then identified by applying several machine learning methods. However, these strategies also require a large training corpus to obtain conclusive results. Typically, the corpus is made up of a few handwritten or annotated source document summaries that show which sentences belong in the summary and which don't.

This work is based on one of the main known Sequence to Sequence (Seq2Seq) models and the most recent and innovative Text-To-Text Transfer Transformer (T5), Pre-trained on Colossal Clean Crawled Corpus (C4), the T5 model produced cutting-edge outcomes on numerous NLP benchmarks and possessed the adaptability to be optimized for a range of significant tasks [16].

A specific variant of KNN (K Nearest Neighbor) in which text is represented by words that are taken as characteristics of numerical vectors. Both attribute and value similarity are considered when calculating the similarity between feature vectors. Semantic classification is the perspective of text summarization. Lines and paragraphs make up the text's division. The classifier assigns a summary or no summary classification to each paragraph or sentence.

Upon summarizing the text and rejecting other texts, the sentences that fall under the category of "summary" are taken out. Better outcomes in text classification and clustering are achieved using the suggested version of KNN. A more condensed representation of the data item and improved performance are the results of the modified KNN [20].

The extraction-based text summarization techniques within the supervised and unsupervised learning framework used extensible summarization techniques. The following are various approaches, with the benefits detailed in the paper. The article by the author also covers a range of assessment techniques, difficulties, and potential future research areas [15][19].

Techniques for fine-tuning pre-trained models are essential in optimizing their performance for specific tasks. These methods involve adjusting the model's parameters based on a smaller, task-specific dataset, allowing for improved accuracy and efficiency. Fine-tuning can include strategies such as freezing certain layers of the model, adjusting learning rates, and employing regularization techniques to prevent overfitting. The process typically begins with a model that has already been trained on the findings that indicate the minimal differences are expected between the downstream and pretrained architectures of a conventional finetuning model, as significant task-specific model adjustments are present [23].

Changes implemented could potentially lead to negative outcomes regarding the results. The studies contribute to the growing body of knowledge regarding pre-trained models and their applications, providing valuable insights that will advance research in natural language processing and text summarization. The T-BERT utilizes a modified transformer architecture designed to facilitate efficient and parallel computation processes. The application of BERT in text summarization introduces comprehensive semantic features [22].

The background data is integrated into the encoding as supplementary information. It is represented as a modifiable topic representation, aimed at guiding the comprehensive process of summary generation [21]. The combination strategy integrates extractive summarization techniques utilizing BERT alongside abstractive summarization methods employing GPT [24].

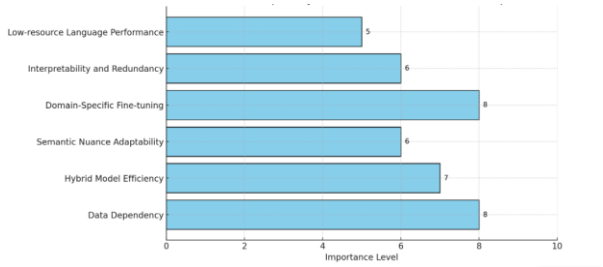

Fig.1. Analysis of Test Summarization Techniques

As figure 1 represents the delineated critical areas that need more investigation, with significance ratings attributed to each gap in text summarization approaches

- **Data Dependency:** A substantial dependence on extensive annotated datasets continues to be a considerable issue. The efficiency of the hybrid model requires enhancement to achieve a smooth balance between extractive and abstractive summarization.
- **Semantic Nuance Adaptability:** Existing models inadequately capture domain-specific semantics.
- **Domain-Specific Fine-tuning**: Refining models for specialized settings continues to pose difficulties. The challenge of ensuring that produced summaries are both interpretable and devoid of repetition remains unresolved.
- **Enhancement of Performance for Low-Resource Languages:** Improvement is required for languages with insufficient data resources.

III. METHODOLOGY

This section is a review and summary of the development of Abstractive Text Summarization from the point of view of methods.

A. Data Collection

As the main source of our information, we "BBC News Summary" (https://www.kaggle.com/datasets/pariza/bbc-news-summary) which contains a detailed collection of many kinds of news stories from different fields such as politics, sports, technology, entertainment, and business. Its feature of providing the most comprehensive and real-life news mock-ups stood behind the choice of the dataset.

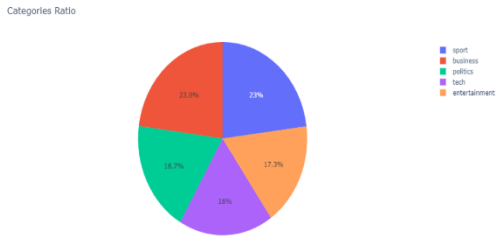

Fig. 2. Categories News Ratio

A total of 2225 text documents from the BBC news website that correspond to the stories that were covered in 2004-2005 make up the data. The data totals approximately 5 MB.

The information is pre-ordered into 5 different class names. Business, Sport, Technology, Entertainment, and Politics are the class labels. It contains 510 text archives from the business classification, 511 text reports from the games class, 401 text records from the tech classification, 386 text reports from the amusement classification, and 417 articles from the governmental issues classification. There is an all-out number of 456,542 words in every one of the records, while containing just 29,126 novel words. This indicates that throughout the entire corpus, each word appears approximately 15 times on average. 205 words is the typical length of a document. However, keep in mind that these statistics are all reported after all stop words have been removed. Since each document has about 1500 words, stop words make up a lot of the document.

B. Preprocessing Data (Character Encoding, Extraction, and Organization):

The articles and summaries framing are required for the texts extracted from the source files. The initial stage of the text representation design involves the selection of the character encoding standard ISO-8859-1. To check for any possible deviations that might influence the future processing of the dataset, a thorough investigation of the length distributions of the dataset was also carried out.

C. Model Architecture

T5 model's "t5-base" version was chosen by us due to its demonstrated NL applicability in a variety of decision-making situations. The T5 model converts all NLP tasks into text-to-text, which was perfect in text summarization.

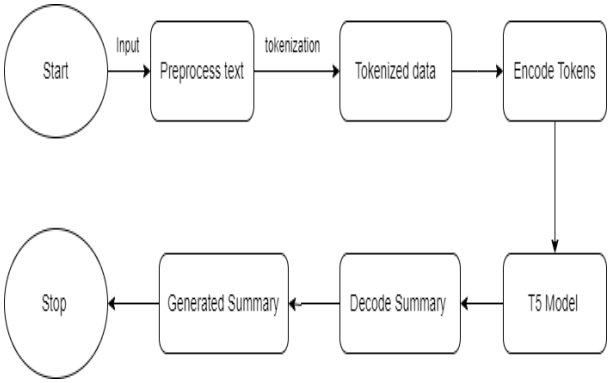

Fig.3. Proposed Model Architecture

- **Start State**: The 'BBC News Summary' dataset from Kaggle is used. The news compilation includes a variety of topics such as introjection of politics, sports, technology, and entertainment/business, serving as a source of data. This dataset was preferred because it has a high content standard, and it is easily applied in real-world news situations.
- **Input Text Preprocessing**: In the text data processing, the "Articles" and "summaries" columns are the only ones that are chosen. The rest are eliminated then; also, non-ASCII characters, which may be unreadable to make the text more readable are eliminated. The text is turned into bytes, the non-ASCII characters are no more, and the bytes are decoded back to their strings after the data has been encoded. The process of extraneous data removal also includes the elimination of non-ASCII characters in the text. Then, the rows of data that are missing are also removed.

- **Preprocessed Text Loading**: It is stored in the T5 project environment when the text has been cleaned and processed. The fully loaded, preprocessed text is available for additional review and editing in the project environment.

- **T5 Model Loading**: The T5 Transformer Model is then operationalized. This model works very well in understanding context and creating naturally occurring language.

- **Model Generation**: The T5 model is triggered in sequence and generates abstractive summaries for supplied news articles.

- **Summary Generation**: By utilizing the provided information as a base, the model generates clear and to-the-point summaries. Through the process of picking the most important information from each article, the summaries try to make the reader's understanding of the content more efficient.

- **End State**: This project is using advanced NLP methods to pull out the relevant data from news stories, which highlights the seamless process from text input to summary generation.

2)Tokenization Strategy and Design: The T5 tokenizer strategy was deployed, where tokenization—the first and extremely important preprocessing step—was performed. A complex technique was applied, namely, a maximum token length of 128 for summaries and of 512 for input articles. Attention masks were tactically incorporated to aid the model in identifying important elements, and special tokens served as the dividers between different text sections.

D. Model Training

Development of News Summary Model: A software-based architecture of the T5 model was well-trained using the PyTorch Lightning Trainer by embedding it with features such as the number of epochs, batch size, and an AdamW optimizer with a learning rate of 0.0001. Tensor Board was used to keep track of training cycles, and the maximum best model checkpoint distance was determined with the aid of a model Checkpoint callback that was initiated from the PyTorch Lightning module. Our design approach emphasized modular approaches to training, validation, testing, and configuring the optimizer. This strategy fostered the flexibility of the solution while simplifying the model-development processes.

Data Module Implementation: A PyTorch Lightning Data Module called the News Summary Data Module was instrumental in overcoming the challenges of batch loading and batching of data within the training stage of the model. The Data Module encompassed the tokenizer as well as the dataset so that controlled and tidy data could be made available to the model.

E. Evaluation

Role of Summarization: Creation of the Summarization Text Feature was a significant first step towards leveraging the capabilities of the trained T5 model. This feature was painstakingly created to enable the tuning of parameters such as the beam search, repetition, and length penalties to a satisfactory level of control over summarizing.

Case Study: The feasibility of the model was performed with a delegate test, which was appropriately chosen from the test data set. An in-depth analysis of the original text and the resulting outline provides a subjective assessment of the model's capacity to extract meaningful data.

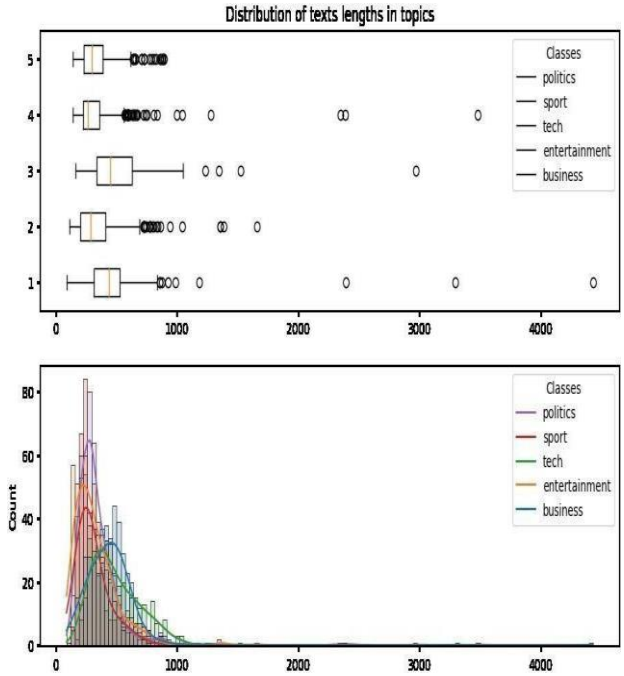

Fig.4. Distribution of text lengths in topics

VI. RESULTS AND DISCUSSIONS

The last attempt to use the text summary method gave positive results in the T5 paradigm (text-to-text transmitter). This section provides a detailed information overview, evaluation, and summary examples of the model and training results, shown in Fig. 5 & Fig. 6. The same has been deployed and can be accessed through the given link

(https://abstractify-th2oftswqwpu8sjn4tzq2a.streamlit.app/)[25].

1) Model Training Performance: There were notable performance indicators from the training procedure that exceeded the allotted number of epochs. The news summary model with PyTorch Lightning Trainer and AdamW Optimizer is used to demonstrate effective learning and convergence. A few crucial training indicators are:

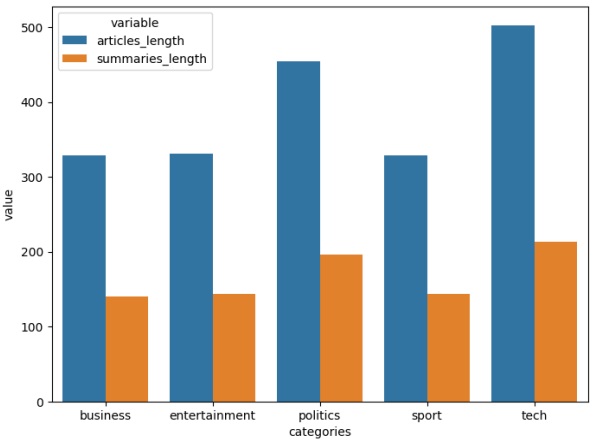

Fig.5. Comparison between Lengths

- **Lost in Training**: A decrease in training losses over the ages has been revealed model and the ability to change its parameters and learn from the data set.

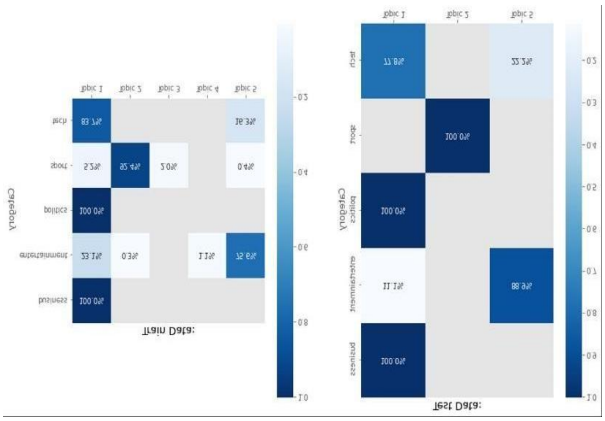

Fig.6. Comparison of Heatmaps of test and train data

-Loss of Validation: Tracking validation loss to ensure that the model fails repositioning the training set was a crucial step in determining the model and potential to generalize.

Since we are generating conditional text using the T5 model, the loss is computed using the forward technique. The loss function used here is part of the T5 model and involves calculating the Training Loss and Validation Loss during the training of the model.

$$\text{Training Loss} = \frac{1}{n}\sum_{i=1}^{n} (y_{true,i} - y_{predicted,i})^2 \quad (1)$$

Where n is the number of training examples $y_{true,i}$ is the true target value for the i[th] training example.

$y_{predicted,i}$ is the predicted target value for the ith training example.

$$\text{Validation Loss} = \frac{1}{m}\sum_{i=1}^{m} (y_{true,val,j} - y_{predicted,val,j})^2 \quad (2)$$

Where is the number of training examples

$y_{true,val,j}$ is the true target value for the jth training example.

$y_{predicted,val,j}$ is the predicted target value for the jth training example.

A final trained model's selection is frequently determined by how well it performs on the validation set. This guarantees that the selected model is the one that exhibits good performance on both training and unseen data.

2) Evaluation of Summarization Quality: A sample was taken from the test dataset to evaluate the quality of the generated summaries. The original news articles were subjected to the 'summarize Text' function, and the resulting summaries were compared with reference summaries that were created by humans.

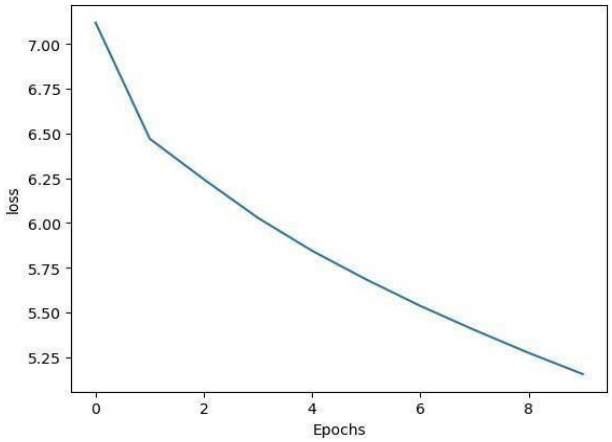

Fig.7. Training Loss Curve

3) Accuracy: The model's accuracy, after evaluation, comes in close to 97%.

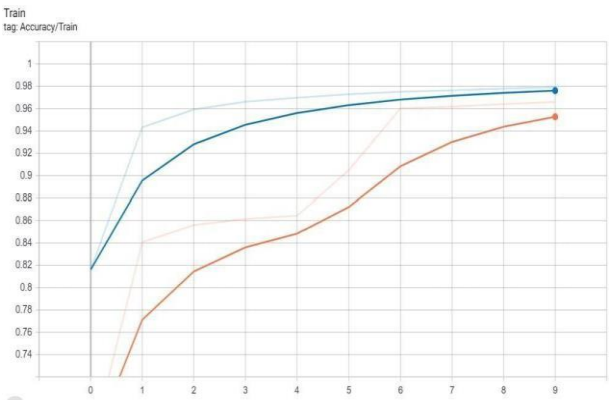

Fig.8. Model performance and training metrics

VII. CONCLUSION

Condensing important information into brief summaries and exploring the challenges of transformer-based model training from various news products from various fields is the aim of this work. The architecture T5 (Text-to-Text Transfer Transformer) and other cutting-edge transformation models are used. Complementing the growing range of abstract text summaries, this study provides a comprehensive overview of transformer-based models with a particular focus on the T5 design. The 97 percent accuracy rate demonstrates how these algorithms can effectively extract useful information from a variety of texts to convey. This study lays the framework for future developments and applications in abstract summarization, as we tackle the rapidly increasing area of natural language processing.

REFERENCES

[1] Ontoum, S and J H Chan (2022). "Automatic text summarization of covid-19 scientific research topics using pre-trained models from hugging face". 2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C).

[2] Haque, M (2013). "Literature Review of Automatic Multiple Documents Text Summarization". International Journal of Innovation and Applied Studies 3, pp. 121–129.

[3] S et al. (2021). "Extractive text summarization for covid-19 medical records". 2021 Innovations in Power and Advanced Computing Technologies.

[4] Lakshmi, A., & Latha, D. DEEP LEARNING FRAMEWORK OF ABSTRACTIVE SUMMARIZATION BASED ON SEMANTIC ROLE LABELLING OF TELUGU TEXT.

[5] Foysal, A. A., & Böck, R. (2023). Who Needs External References? —Text Summarization Evaluation Using Original Documents. AI, 4(4), 970-995.

[6] Patil, S., Pawar, A., Khanna, S., Tiwari, A., & Trivedi, S. (2021). Text Summarizer using NLP (Natural Language Processing). Journal of Computer Technology & Applications, 12(3), 1-6p.

[7] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate". In: 3rd International Conference on Learning Representations, ICLR 2015. Ed. by Yoshua Bengio and Yann LeCun.

[8] Cohan, Arman (2018). "A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents". In: Proceedings of the 2018 Conference of the North American

Chapter of the Association for Computational Linguistics: Human Language Technologies. Vol. 2. Association for Computational Linguistics, pp. 615–621.

[9] Dong, Yue (2018). "BanditSum: Extractive Summarization as a Contextual Bandit". In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Ed. by Ellen Riloff et al. Association for Computational Linguistics, pp. 373–373.

[10] Erkan, Günes and Dragomir R Radev (2004). "Lexrank: Graph-based lexical centrality as salience in text summarization". Journal of artificial intelligence research 22, pp. 457–479

[11] Fuentes M, Alfonseca E, Rodríguez H (2007) Support vector machines for query- focused summarization trained and evaluated on pyramid data. In: Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions. pp 57–60

[12] Gehrmann, Sebastian, Yuntian Deng, and Alexander M Rush (2018). "Bottom-Up Abstractive Summarization". In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Ed. by Ellen **[13]** Riloff et al. Association for Computational Linguistics, pp. 4098–4109.

[14] Kryscinski, Wojciech (2018). "Improving Abstraction in Text Summarization". In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Ed. By Ellen Riloff et al. Association for Computational Linguistics, pp. 1808–1817.

[15] Lierde, H Van and Tommy W S Chow (2019). "Queryoriented text summarization based on hypergraph transversals". Information Processing Management 56, pp. 306–4573.

[16] N.Moratanch, S. Chitrakala, "A Surveyon Extractive Text Summarization." IEEE International Conference on Computer, Communication and Signal Processing (ICCCSP), 2017.

[17] Ozsoy, M G, F N Alpaslan, and I Cicekli (2011). "Text summarization using latent semantic analysis". Journal of Information Science 37(4), pp. 405–417. Raffel, Colin (2019).

[18] Saggion, H and T Poibeau (2013). "Automatic text summarization: Past, present and future", Multi-source". Multilingual Information Extraction and Summarization, pp. 3–21.

[19] Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to Sequence Learning with Neural Networks". *Advances in Neural Information Processing Systems 27*. Ed. By Z. Ghahramani et al., pp. 3104–3112.

[20] Tan, Jiwei, Xiaojun Wan, and Jianguo Xiao (2017). "Abstractive Document Summarization with a GraphBased Attentional Neural Model". In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vol. Association for Computational Linguistics, pp. 1171–1181.

[21] Taeho Jo, "K Nearest Neighbor for Text Summarization using Feature Similarity." International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017.

[22] Sutar, S., Surve, I., Munawwar, M., Nanaware, V., & Dhumal, P. (2024). Text Summarization Using NLP.

[23] Ma, T., Pan, Q., Rong, H., Qian, Y., Tian, Y., & Al-Nabhan, N. (2021). T-bertsum: Topic-aware text summarization based on bert. IEEE Transactions on Computational Social Systems, 9(3), 879-890.

[24] Zhang, H., Li, G., Li, J., Zhang, Z., Zhu, Y., & Jin, Z. (2022). Fine-tuning pre-trained language models effectively by optimizing subnetworks adaptively. Advances in Neural Information Processing Systems, 35, 21442-21454.

[25] Deployed Project Link

(https://abstractifyth2oftswqwpu8sjn4tzq2a.streamlit.app/)

Published on 07/05/2025