IFERP
Explore Your Research Journey...

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 10, October 2024

NLP Driven Textdigest: Automating Summarization for Efficiency

[1] Aneerban Saha, [2] Akshita Jain, [3] K. Krishna Koushika

[1] [2] [3] Department of Artificial Intelligence & Machine Learning, Manipal University Jaipur
Corresponding Author Email: [1] aneerbansaha22@gmail.com, [2] akshi132@gmail.com, [3] krishnakoushika6@gmail.com

Abstract— In the age of information overload, the need for accessible and concise content summarization from lengthy stories and articles is more critical than ever. This study focuses on the application of the t5 model in the realm of text summarization, building upon the foundation laid by previous research efforts. Recognizing that readers sometimes struggle to comprehend lengthy tales, this effort focuses on the problem of reducing and summarizing stories and large articles. The t5 model, a powerful Natural Language Processing tool, is at the heart of our approach. T5 offers the potential to distill complex narratives into coherent and digestible summaries.

To evaluate the effectiveness of T5 in this context, we employ well-established metrics, including ROUGE scores. Our study adds to the efforts being made to improve text summary methods, particularly for long articles and stories. By harnessing the capabilities of the t5 model, we aim to enhance the accessibility and comprehensibility of narrative content, thus addressing the pressing need for efficient information consumption in the digital age.

Index Terms— Natural language processing, text summarization, extractive summarization, abstractive summarization, Transformer, T5 Model.

I. INTRODUCTION

Computer science's Natural Language Processing (NLP) is an AI subfield [3]. NLP is used to organise or classify the information that is extracted from data. NLP is applied in fields such as speech and text recognition, language generation, and understanding. Autocorrect and Autocomplete, Grammarly, voice typing (speech-to-text), language translation, text summarization, sentiment analysis, voice assistance, and other common NLP applications are examples of NLP that we encounter on a daily basis. Through text summarization, natural language processing (NLP) can effectively condense a large amount of information into a brief paragraph while maintaining the original meaning and content. In essence, it's a method of extracting the most crucial information from vast amounts of text data and summarising it [1].

Text summarizers function by eliminating pronouns, verbs, and other unnecessary grammar, extracting keywords and significant passages, and figuring out how frequently words are used. In the shortest amount of time, text summarization enables us to extract the most useful information from the data or content [4]. The importance of text summarization is rising in the rapidly expanding world of today.

But NLP goes beyond mere convenience; it powers the very foundation of human-machine communication. Voice assistants like Siri, Google Assistant, and Alexa have become our digital confidants, interpreting our spoken words, and responding with uncanny accuracy. NLP algorithms work diligently behind the scenes to interpret user intent, provide suitable replies, and perform activities that smoothly integrate into our everyday routines.

One facet of NLP that holds particular promise is its capacity to distill vast troves of text into concise, meaningful summaries. Imagine sifting through mountains of information, extracting the essential nuggets, and presenting them in a format that conveys the core message without the noise. This is the magic of text summarization, a process that has the potential to revolutionize the way we manage and absorb information [2].

There are four basic strategies for text summarization: extractive, indicative, informative, and abstractive. In this paper, our focus will be squarely on the innovative realm of abstractive text summarization. While extractive methods rely on existing words and phrases from the source material, abstractive techniques take a bold leap forward. They entail not just extracting but also rephrasing, creating unique formulations that capture the spirit of the source material while delivering the message more effectively [5].

Abstractive text summarization involves paraphrasing and briefly summarising the original content while ensuring that the meaning of the summarised material is consistent with the original. It employs a different keyword in place of the one from the original content. The original content is matched by the phrases and sentences that the algorithm generates. It functions similarly to the human brain, which gathers all semantic data, selects keywords based on semantic similarity, and employs that information in the summary. Abstract summarization can be accomplished using a range of NLP-based pretrained models.

In this era of advanced NLP, we're blessed with a wealth of pre-trained models that stand as testaments to human-machine collaboration. These models have raised the bar for abstractive text summarization, offering the promise of more effective content synthesis and comprehension. Our

IFERP Explore Your Research Journey...

research in this area intends to provide light on cutting-edge methodologies, problems, and future breakthroughs in the intriguing field of abstractive text summarization.

II. LITERATURE REVIEW

A. Introduction

A crucial task in natural language processing (NLP) is text summarization, which entails extracting important information from source texts to produce succinct and logical summaries. Transformer-based models, which provide cutting-edge performance, have transformed natural language processing tasks in recent years. Examining important works that have influenced text summarization, this literature review focuses on the T5 (Text-to-Text Transfer Transformer) model.

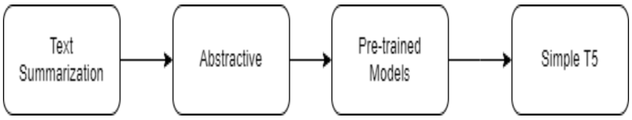


Fig 1. Data flow

The quantity of summarization models that are released annually has been rising quickly. Largescale data availability and developments in neural network architectures [16, 17] made it possible to move from expert knowledge and heuristic-based systems to data-driven methods driven by end-toend deep neural models. Text summarization techniques currently in use include graph-based methods that arrange the input text in a graph and then use ranking or graph traversal algorithms to construct the summary [18] [19], reinforcement learning strategies [24], hybrid extractiveabstractive models [20] advanced attention and copying mechanisms [23, 25], multi-task and multireward training techniques [26], and hybrid methods.

This work is based on one of the main known Sequence to Sequence (Seq2Seq) models [6] and the most recent and innovative Text-To-Text Transfer Transformer (T5) [21]. 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 optimised for a range of significant tasks.

An overview of the many approaches and strategies used in automatic text summarization is given by

Adhika Pramita Widyassari et al. [14], with an emphasis on the Natural Language Toolkit (NLTK). The author examines various strategies, such as extractive and abstractive summarization, and talks about how NLTK can be applied to these methods.

- Preprocessing: By dividing text into words or sentences and reducing words to their root form, NLTK helps with information extraction by carrying out crucial text preprocessing tasks like tokenization, stemming, and stop-word removal.
- Sentence Scoring: By providing tools to compute sentence similarity (such as cosine similarity) and assign scores, NLTK enables extractive summarization by facilitating the selection of pertinent sentences based on their significance.
- Feature extraction: By helping to identify entities and important terms, NLTK's named entity recognition and part-ofspeech tagging improve summary relevance and accuracy.
- Language Modelling: By forecasting likely words or phrases, NLTK assists in the construction of language models (such as n-gram models) for abstractive summarization, which produces succinct and coherent summaries.
- Assessment: NLTK comprises evaluation metrics, such as ROUGE and BLEU, to evaluate the quality of the summary by contrasting it with reference summaries and measuring similarity or effectiveness.

Abstractive text analysis, a natural language processing (NLP) technique that tries to create a succinct and coherent summary of a given text by comprehending its content and creating new sentences, is examined by Khilji et al. [15].

Abstractive summarization entails crafting original sentences that, in a more humane way, convey the essential details and concepts of the original text.

Makbule Gulcin Ozsoy[6] proposed that semantic analysis is the most recent method of text summarization in 2011. For summarising purposes, these cross-topic approaches can be used in any language. The algorithms are evaluated on both Turkish and English documents, and the outcomes are compared using their Rouge scores.

2020 saw Atif Khan [7] He also developed the idea of abstractively summarising multiple documents through text summarization using a semantic graph. Achempong Francisca Adoma [3] compared the efficacy of pre-trained transformer models, such as Xlnet, RobertA, BERT, and DistilBert, for text-based emotion recognition. In the study, there was a high level of recognition accuracy with RobertA.

Li Zhang [8] talked about fine-tuning strategies for pre-trained models in 2021. According to the study's findings, there shouldn't be many differences between the downstream and pretrained architectures of a conventional finetuning model because substantial task-specific model modifications may have an adverse effect on the result. These studies add to the expanding corpus of knowledge on pre-trained models and their uses, offering insightful information that will help NLP and text summarization research go forward.

Hau Sheng-Laun [9] An extensive paper exploring the many uses of automatic text summarization. In addition to highlighting the present difficulties and restrictions associated with the techniques and algorithms employed in this field, the paper offers insights into the approaches utilised in ATS. The goal of this work is to stimulate future research on these problems and novel ATS challenges, thereby propelling the field forward and raising the calibre and efficacy of automated text summarization methods. In 2021, Y. Chen [10] presented a technique that proposed automatically creating a restaurant template with all pertinent data, including certain subjects.

The newly proposed T-BERT Sum [11] is based on a modified transformer architecture that achieves efficient and parallel computation. It applies BERT in text summarization, which introduces rich semantic features. The background data is regarded as being incorporated into the encoding as extra knowledge, and it is encoded as an adjustable topic representation with the goal of directing the endto-end process of creating summaries [13].

The development of NLP has prompted scholars to investigate language-based summarization strategies.

A number of previous works have been proposed, and T. Islam's recent work [12] focuses on text summarization in Bangla text documents.

III. METHODOLOGY

A. Data Collection and Preprocessing

1) Dataset Selection and Rationale: The "BBC News Summary" dataset, a carefully selected collection of news articles covering a wide range of topics like politics, sports, technology, entertainment, and business, serves as our main source of data. This dataset was selected due to its high content quality and applicability to actual news situations.

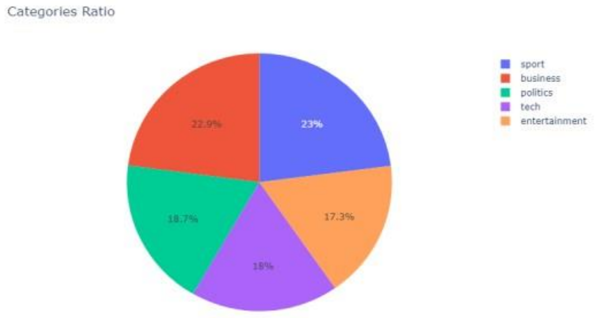


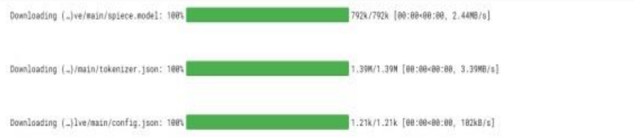
Fig 2. Categories News Ratio

2) Data Preprocessing: *Extraction, Organization, and Character Encoding:* After the raw text data was taken out of the source files, it was systematically organised by domain to classify the summaries and articles. During the preprocessing stage, the ISO-8859-1 character encoding standard was used to guarantee consistency in text representation. Furthermore, a thorough examination of the length distributions within the dataset was conducted in order to identify any potential variations that might affect the way in which the dataset is processed in the future.

The code is downloading model files from some source. These files include the vocabulary (spiece.model), tokenizer information (tokenizer.json), and model configuration (config.json) for the T5 model.

B. Model Architecture

1) T5 Model Selection and Background: The "t5base" variant of the T5 (Text-to-Text Transfer Transformer) model is our choice because of its proven performance in a variety of natural language processing tasks. All NLP tasks can be naturally framed as text-to-text problems by the T5 model, which fits in well with text summarization.



```
Downloading (…)ve/main/spiece.model: 100%  792k/792k [00:00<00:00, 2.44MB/s]

Downloading (…)/main/tokenizer.json: 100%  1.39M/1.39M [00:00<00:00, 3.39MB/s]

Downloading (…)lve/main/config.json: 100%  1.21k/1.21k [00:00<00:00, 102kB/s]
```

Fig 3. The code is downloading model files from some source. These files include the vocabulary (spiece.model), tokenizer information (tokenizer.json), and model configuration (config.json) for the T5 model.

2) Tokenization Strategy and Design: Using the T5 tokenizer, tokenization—a crucial preprocessing step—was carried out. A sophisticated approach was used, designating a maximum token length of 128 for summaries and 512 for input articles. To help the model focus on relevant information, attention masks were strategically applied, and special tokens were introduced to distinguish different sections of text.

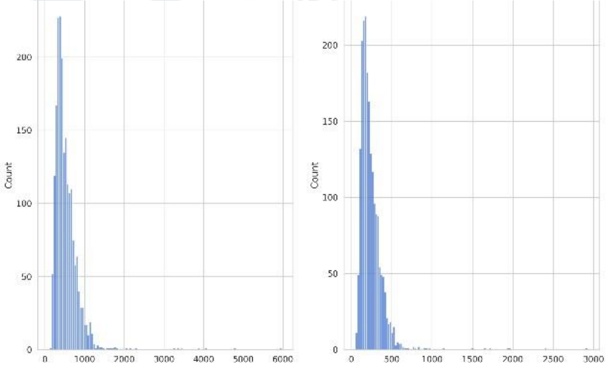
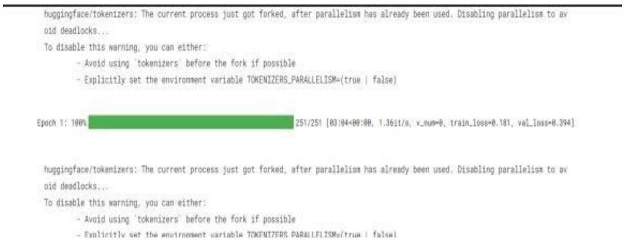


Fig 4. Comparison of full text token counts and summary tokens

C. Model Training

1) Development of NewsSummaryModel: An architecture encapsulating the T5 model was created using the NewsSummaryModel PyTorch Lightning module as the centrepiece. Modular techniques for training, validation, testing, and optimizer configuration were given top priority in our design philosophy. This method improved the solution's adaptability while streamlining the modeldevelopment process.

2) Implementation of DataModule: During the training phase, the NewsSummaryDataModule, a PyTorch Lightning DataModule, was essential in handling the complexities of data loading and batching. The DataModule made sure that the flow of data to the model was coherent and wellorganized by encapsulating the tokenizer and dataset.



```
huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to av
oid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)

Epoch 1: 100%|                    | 251/251 [03:04<00:00, 1.36it/s, v_num=0, train_loss=0.181, val_loss=0.394]

huggingface/tokenizers: The current process just got forked, after parallelism has already been used. Disabling parallelism to av
oid deadlocks...
To disable this warning, you can either:
        - Avoid using `tokenizers` before the fork if possible
        - Explicitly set the environment variable TOKENIZERS_PARALLELISM=(true | false)
```

Fig 5. training progress updates provide information about the completion of each epoch and the associated training and validation losses.

3) Training Configuration: The configuration of the model for training required careful consideration of the hyperparameters. The AdamW optimizer was selected to achieve a balance between convergence speed and stability, using a conservative learning rate of 0.0001. TensorBoard logging was used to track training progress, and the ModelCheckpoint callback was carefully used to hold on to the topperforming model in accordance with validation loss.

D. Evaluation

1) Summarization Function: A crucial first step in making use of the trained T5 model's capabilities was the creation of the summarizeText function. This function was painstakingly created to precisely control the summarization process by fine-tuning parameters like beam search, repetition penalty, and length penalty.

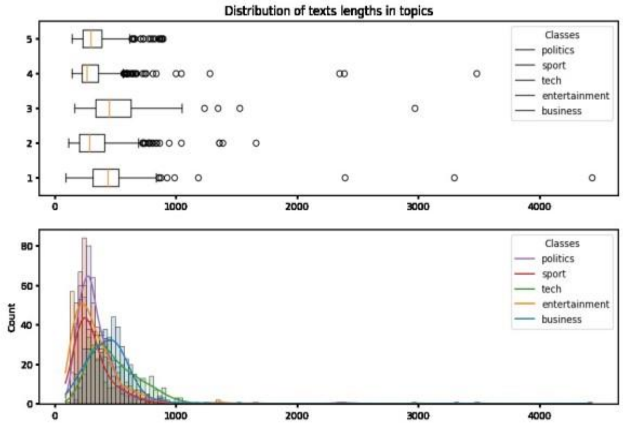


Fig 6. Distribution of texts lengths in topics

2) Case Study: A representative sample was carefully chosen from the test dataset in order to thoroughly assess the effectiveness of the model. A qualitative evaluation of the original text and the summary that was produced gave detailed insights into the model's capacity to extract important information.

E. Computational Resources

1) GPU Utilization: During the model training phase, we strategically used GPU acceleration to fully utilise the computational capacity. GPUs' parallel processing powers greatly accelerated training, which enhanced the scalability and effectiveness of the suggested solution.

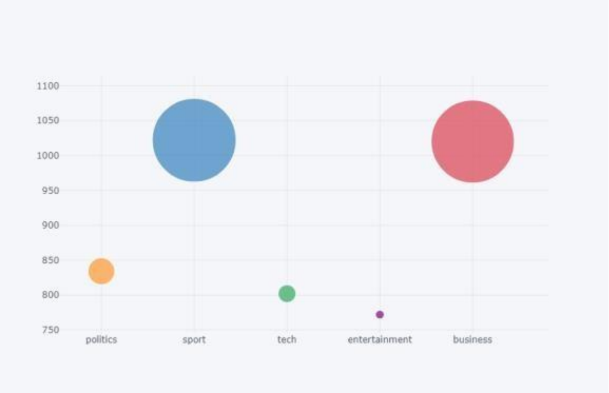


Fig 7. Distribution Size of Each Category

F. Code Implementation

1) Programming Language and Libraries: The Python programming language was utilised in the realisation of this research project. Python was selected due to its extensive support for machine learning frameworks and its versatility. The foundation of the project was formed by essential libraries like Matplotlib, PyTorch, Transformers, and PyTorch Lightning, which allowed for a reliable and effective development and analysis pipeline.

IV. RESULT AND DISCUSSION

Our final attempt to apply a T5 (Text-to-Text Transfer Transformer) model-based text summarization solution has produced encouraging results. A thorough analysis of the outcomes from the model's training, assessment, and summarization examples is provided in this section.

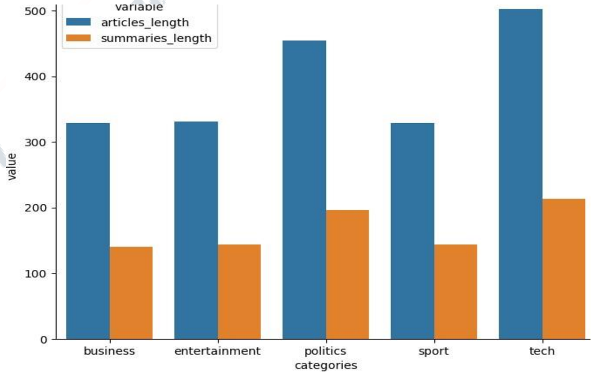


Fig 8. Comparison between Lengths

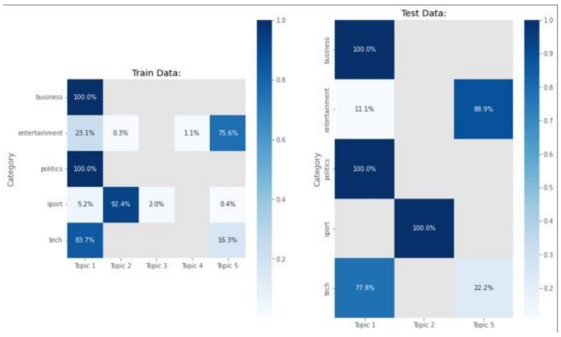


Fig 9. Comparison of Heatmaps of test and train datasets

1) Performance of Model Training: Important performance metrics were displayed by the training process, which was carried out over a predetermined number of epochs. With the help of the PyTorch Lightning Trainer and the AdamW optimizer, the NewsSummaryModel was able to show effective learning and convergence. Some important training metrics are:

- Lost in Training: The model's capacity to adjust its parameters and learn from the dataset was demonstrated by the decrease in training loss over epochs.

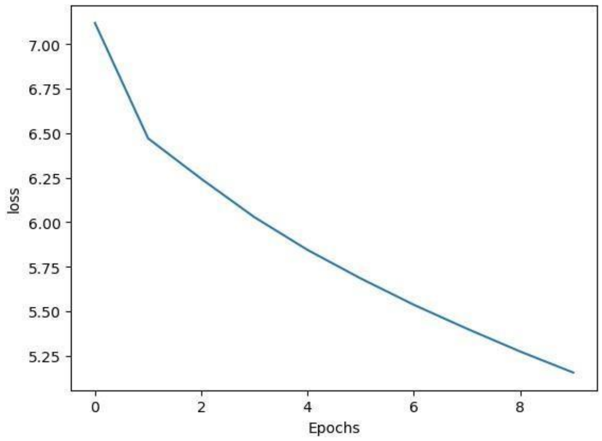


Fig 10. Training Loss Curve

-Loss of Validation: A critical way to assess the model's capacity for generalisation was to keep an eye on the validation loss to make sure the model didn't overfit to the training set.

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

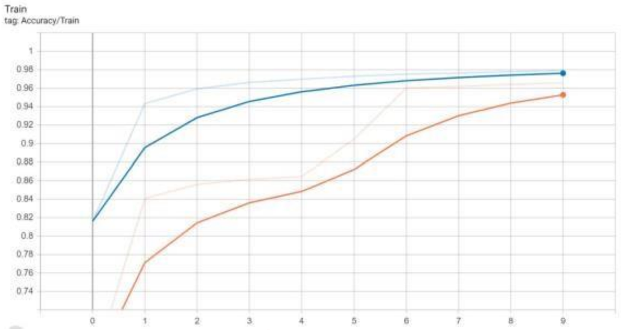


Fig 11. Model performance and training metrics

V. CONCLUSION

In this work, we introduced a thorough investigation of text summarization that makes use of the T5 (Text-to-Text Transfer Transformer) model's transformative powers. The study path included data gathering, preprocessing, choosing a model architecture, training, and assessment, offering a comprehensive picture of the techniques used. The literature review established the applicability and importance of our method by placing it within the larger framework of transformer-based models and text summarization.

A. Dataset Insights

By using the "BBC News Summary" dataset, a representative and varied set of news articles could be gathered, enabling a more in-depth investigation of text summarization in a variety of contexts

B. T5 Model Utilization

The T5 model's ability to frame summarization as a text generation task was demonstrated by its adoption. T5's distinct architecture, which treats all NLP tasks equally, made development easier and produced encouraging abstractive summarization results.

C. Fine-Tuning and Adaptation

T5's fine-tuning on domain-specific datasets— news articles in this case—proved crucial to the model's adaptation to the complexities of summarization across different categories. This method produced more contextually relevant summaries and improved the model's contextual understanding.

D. Challenges Explored

The examination of text summarization's difficulties, such as managing lengthy documents, producing cogent summaries, and reducing biases, revealed opportunities for additional study and improvement.

E. Literature Review Context

The literature review positioned our work in relation to transformer-based models, text summarization, and the unique contributions of the T5 model. This contextualization clarified things for us and emphasised how important our project is to the advancement of the field.

REFERENCES

[1] H. Saggion and T. Poibeau, “Automatic text summarization: Past, present and future”, Multi-source, Multilingual Information Extraction and Summarization, ed: Springer, pp. 3- 21., 2013

[2] M. Haque, et al., “Literature Review of Automatic Multiple Documents Text Summarization”, International Journal of Innovation and Applied Studies, vol. 3, pp. 121-129, 2013.

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

[4] S, D., N, L.K. and S, S. (2021) “Extractive text summarization for covid-19 medical records,” 2021 Innovations in Power and Advanced Computing Technologies (i-PACT)

[5] Lakshmi, A. and Latha, D. (2022) ‘Automatic text summarization for Telugu language’, 2021 4th International

Conference on Recent Trends in Computer Science and Technology (ICRTCST)

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

[7] Khan, A. et al. (2018) "Abstractive text summarization based on improved semantic graph approach," International Journal of Parallel Programming, 46(5), pp. 992–1016.

[8] Zhang, L. and Hu, Y. (2021) "A fine-tuning approach research of pre-trained model with two stage," 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)

[9] Hou, S.-L. et al. (2021) "A survey of text summarization approaches based on Deep Learning," Journal of Computer Science and Technology, 36(3), pp. 633–663.

[10] Chen, Y., Chang, C. and Gan, J. (2021) "A template approach for summarizing restaurant reviews," IEEE Access, 9, pp. 115548–115562.

[11] Ma, T. et al. (2022) "T-bertsum: Topic-aware text summarization based on bert," IEEE Transactions on Computational Social Systems, 9(3), pp. 879–890.

[12] Islam, T., Hossain, M. and Arefin, M.D.F. (2021) "Comparative analysis of different text summarization techniques using enhanced tokenization," 2021 3rdInternational Conference on Sustainable Technologies for Industry 4.0 (STI)

[13] Prayitno, A. and Nilkhamhang, I. (2022) "Synchronization of heterogeneous vehicle platoons using distributed model reference adaptive control," 2022 61st Annual Conference of the Society of Instrument and Control Engineers (SICE)

[14] Adhika Pramita Widyassari, Supriadi Rustad, Guruh Fajar Shidik, Edi Noersasongko, Abdul Syukur, Affandy Affandy, De Rosal Ignatius Moses Setiadi, "Review of automatic text summarization techniques & methods", Journal of King Saud University 2022

[15] Khilji, Abdullah & Sinha, Utkarsh & Singh, Pintu & Ali, Adnan & Pakray, Dr. Partha "Abstractive Text Summarization Approaches with Analysis of Evaluation Techniques", Computational Intelligence in Communications and Business Analytics 2021

[16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. Ed. by Yoshua Bengio and Yann LeCun. 2015.

[17] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to Sequence Learning with Neural Networks". In: Advances in Neural Information Processing Systems 27. Ed. by Z. Ghahramani et al.Curran Associates, Inc., 2014, pp. 3104–3112.

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

[19] H. Van Lierde and Tommy W.S. Chow. "Queryoriented text summarization based on hypergraph transversals". In: Information Processing Management 56.4 (2019), pp. 1317–1338. ISSN: 0306-4573

[20] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. "Bottom-Up Abstractive Summarization". In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4,2018. Ed. by Ellen Riloff et al. Association for Computational Linguistics, 2018, pp. 4098–4109.

[21] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In arXiv preprint arXiv:1910.10683 (2019).

[22] Ashish Vaswani et al. "Attention is All you Need".In: Advances in Neural Information Processing Systems 30. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008.

[23] Arman Cohan et al. "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, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 615–621

[24] Yue Dong et al. "BanditSum: Extractive Summarization as a Contextual Bandit". In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. Ed. by Ellen Riloff et al. Association for Computational Linguistics, 2018, pp. 373.

[25] Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. "Abstractive Document Summarization with a GraphBased Attentional Neural Model". In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1171–1181.

[26] Wojciech Kryscinski et al. "Improving Abstraction in Text Summarization". In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018. Ed. by Ellen Riloff et al. Association for Computational Linguistics, 2018, pp. 1808– 1817.

IFERP Explore Your Research Journey...