Performance Analysis of Transformer Models in Text Summarization with Insights for Future Ensemble Techniques

1st Subhojit Ghosh
Department of CSE
Institute of Engineering & Management
UEM, Kolkata, India
subhojitghosh666@gmail.com

2nd Aritra Ghosh
Department of CSE
Institute of Engineering & Management
UEM, Kolkata, India
aritrag1905@gmail.com

3rd Anupam Mondal
Department of CSE
Institute of Engineering & Management
UEM, Kolkata, India
anupam.mondal@iem.edu.in

***Abstract*— This paper explores the capabilities of transformer-based models BART, RoBERTa, and PEGASUS in abstractive text summarization. It uses a large diverse dataset of 200,000 rows, which was found to illustrate their performance in different content types, including conversational, journalistic, and transcribed texts. Preprocessing techniques used for model training included tokenization, lemmatization, and vectorization. Evaluation metrics BLEU and ROUGE were used to assess the quality of the output. A stack-based ensemble approach is proposed as a future enhancement, making use of the strengths of individual models to optimize coherence, accuracy, and generalization. Deployment considerations including scalability, latency, and cloud-based hosting are discussed with practical strategies for real-world implementation.**

Keywords*— Transformer-based models, BART, RoBERTa, PEGASUS, abstractive text summarization, diverse dataset, preprocessing techniques, evaluation metrics, ROUGE, BLEU, stack-based ensemble, scalability, latency, real-world implementation.

I. Introduction

Text summarization, which refers to the distillation of all important information from lengthy documents while retaining their core ideas, is essential in today's information-driven world. The exponential growth of data calls for effective summarization techniques across different fields. For example, in journalism, summarization allows news articles to be condensed into critical information, enabling readers to quickly grasp essential events without sifting through extensive details. The use of summaries helps researchers understand the main findings and methodologies of related studies much faster, which accelerates the review process and maximizes research efficiency. In the corporate world, executives use summaries of long reports to make informed decisions in time-sensitive matters promptly. Additionally, social media platforms rely on summarization algorithms to give users short and impactful updates, thereby capturing the essence of discussions.

As the demand for summarization is on the rise, several methodologies have been devised. The two most traditional techniques are fuzzy logic, which deals with vague information by using set theory, and concept-driven methods that focus on key ideas based on conceptual relationships. Latent semantic models, such as Latent Dirichlet Allocation (LDA), find semantic patterns in texts, whereas machine learning techniques, including Naive Bayes, Decision Trees, and Support Vector Machines, depend on labeled data for summarization tasks. Advanced neural network-based models, transformers, and recurrent neural networks learned deep representations of text and provided excellent capabilities to generate accurate, contextually relevant summaries. Others include Conditional Random Fields (CRFs), treating summarization as a sequence labeling task for extraction of essential phrases, and tree-based strategies to inform summarization decisions using the discourse structure of the text. Ontology-based methods rank key ideas using structured knowledge bases, and semantic graph-based techniques map relationships between entities to enhance summary quality.

Despite these breakthroughs, many challenges remain, particularly in maintaining precision, coherence, and efficiency. Recent advances in large language models, such as BART, RoBERTa, and PEGASUS, have shown impressive results in complex language tasks and the generation of contextually nuanced summaries. However, these models are often plagued by issues with computational efficiency, scalability, and general performance in a wide range of real-world applications.

This paper looks at these problems by assessing how well BART, RoBERTa, and PEGASUS summarize in contrast to the original models. Also, it brings forth a method of stack-based ensemble as a course toward improving summation performance. In doing all this, through these research ideas, this particular study seeks contributions toward solving major persistent problems regarding text summarization.

The structure of this paper is organized as follows: Section 1 outlines the objectives of this study. Section 2 reviews related work, highlighting key methodologies and advancements. Section 3 details the proposed methodology, including dataset preparation, preprocessing, and fine-tuning strategies. Section 4 discusses model performance and the integration of RoBERTa for combining outputs. Section 5 covers implementation details and deployment strategies. Section 6 presents results and discussions based on evaluation metrics and the effectiveness of the approach. Finally, Section 7 concludes with insights, future work, and the implications of this research.

II. Related Work

This area of research, from heuristic and statistical methods to sophisticated Deep learning models and machine learning are quite active in text summarization. The traditional extractive summarization methods like Latent Semantic Analysis (LSA) and Term Frequency-Inverse Document Frequency (TF-IDF) [1] attempt to determine the most representative sentences from a text. However, most of these methods lead to incoherent and not fluent summaries and shift toward abstractive summarization [3].

Neural Network-based Models: The first sequence-to-sequence (Seq2Seq) neural models were improvements on the performances of early extractive summarizers to generate summaries. Transformer-based models represented the giant leap ahead in the space [12]. Models like GPT, BERT, and their variants, having a bidirectional and autoregressive nature improved contextual understanding suitable for complicated language tasks such as summarization [3].

Transformer-based Models:

BART: BART is a pretraining denoising autoencoder for sequence-to-sequence models, which performs well in abstractive summarization through corrupted input reconstruction [4]. It has reported strong results on summarization benchmarks such as CNN/DailyMail.

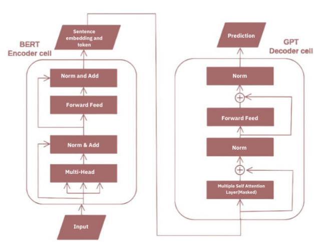

Fig. 1. Comparison of BERT Encoder Cell and GPT Decoder Cell Architectures

RoBERTa: Although primarily developed for understanding tasks, its contextual embedding capabilities have been leveraged in ensemble approaches for sentence ranking and semantic similarity tasks, which are applicable to text summarization [5].

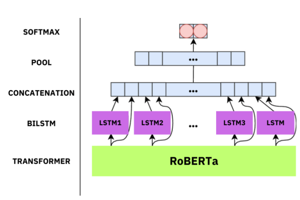

Fig. 2. Hybrid RoBERTa-BiLSTM Architecture for Text Classification

PEGASUS: Particularly trained specifically for abstractive summarization, PEGASUS leverages gap-sentence generation objectives to better perform on the document-level summarization task [6]. This surpasses the capabilities of several previously proposed models with longer document input and maintained factual coherence [14].

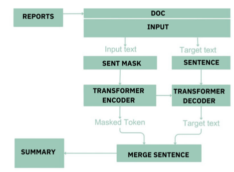

Fig. 3. Transformer-based Text Summarization Pipeline

Ensemble Techniques in Summarization: Ensemble learning has been experimented to combine strengths of different models, addressing their limitations individually [17]. Some of the common techniques include weighted averaging [13], boosting, and stacking [7]. Recent approaches have been towards hybrid systems by integrating extractive and abstractive methods that improve summary coherence and informativeness. There is also leverage on semantic similarity tools like RoBERTa in meta-layer processing to improve the quality of final summaries [10].

Deployment Challenges: Research on the deployment of summarization models has addressed issues related to scalability, latency, and the retraining of models. Solutions such as model quantization [15], caching, and cloud-hosting [17] solutions have been explored to optimize performance in real-time applications. The simple deployment of large-scale applications has been made possible through APIs pre-built by platforms like Hugging Face and efficient hosting infrastructure.

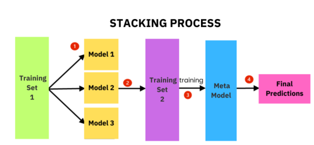

Fig. 4. Stacking Process in Ensemble Learning

III. Proposed Methodology

The dataset utilized in this research has 200,000 entries and is curated in such a way that it generalizes all content types ranging from formal sources to informal sources. This multi-source data will provide the summarization model with the adaptability that it needs in order to process various linguistic styles and structures, which may include chat data, news articles, or transcripts from video content. [12] This broad spectrum of text formats trained in the models will allow us to evaluate their performance across varied contexts, thus enhancing their practical applicability in real-world scenarios.

The dataset is structured with three main columns:

1) **Serial Number:** A unique identifier for each entry, ensuring easy indexing and retrieval of data.
2) **Text:** Full text of original content, the variety of formats—a conversational, journalistic style and spoken language transcription. This column will lay a wide-based foundation for the models in learning the many linguistic cues and stylistic elements.
3) **Summary:** A handcrafted or distilled summary of the text, where the main ideas are summarized. These summaries serve as output targets that inform the models on how to produce accurate, contextually appropriate summaries.

We further preprocessed the data for training the models through a thorough preprocessing pipeline that consisted of the following steps:

- **Tokenization:** The process of breaking down text into individual tokens so that models can understand and process distinct units of language.
- **Lemmatization:** Getting words to their base or root form; helps standardize vocabulary without losing meaning.
- **Vectorization:** This is essentially the process of converting text into numerical representations so that models can learn the patterns in structured input efficiently.

This multiplicity of content style pushes the BART, RoBERTa, and PEGASUS models to detect and adapt toward diverse content styles that facilitate their ability to make high-quality summaries from inputting various types of inputs. This method not only strengthens the generalization capability of the model but also improves its efficiency while applied on practical summarization tasks in a broader domain.

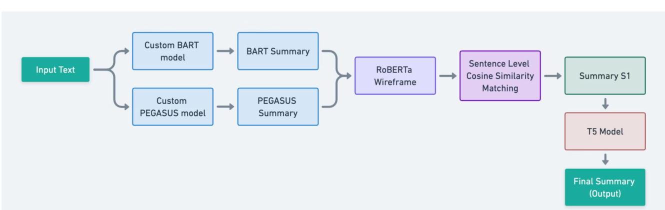

Fig. 5. Hybrid summarization framework combining BART and PEGASUS models, refined using RoBERTa-based cosine similarity and T5 model to generate the final summary

In the subsequent section, the methodologies utilized to develop our summarization system, such as preparing data, model fine-tuning, and also the strategy of developing final summaries, have been outlined.

A. Model Fine-tuning

Given the constraints on our computational capabilities, training a large-scale model on the full dataset was unfeasible. We implemented a batch-wise incremental training paradigm for fine-tuning two pre-trained transformer models: BART and PEGASUS.

B. Incremental Training Procedure

The fine-tuning procedure was performed in an iterative batch-by-batch manner of 10,000 training examples. We started with the pre-trained facebook/bart and google/pegasus models from the Hugging Face library as our initial models, which we denoted as M1. The first model M1 was fine-tuned using the first 10,000 training samples to generate an updated model M2. Successively, the updated model Mi was fine-tuned using the i-th batch of 10,000 samples to obtain Mi+1. This was continued until all the training batches were used. In this manner, we have effectively trained on the entire dataset without violating computational restrictions.

C. Model Performance

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measure was used to assess the models on the validation set following each round of fine-tuning. The following were the final ROUGE scores attained. ROUGE-1, ROUGE-2, and ROUGE-L scores for the BART model were 44.5%, 21.7%, and 41.2%, respectively. The ROUGE-1, ROUGE-2, and ROUGE-L scores for the PEGASUS model were 46.3%, 23.1%, and 43.0%, respectively. According to these scores, both models demonstrated competitive performance, with PEGASUS demonstrating a minor edge in capturing overall sentence-level coherence (ROUGE-L) and bi-gram overlaps (ROUGE-2).

D. Summary Generation

We fine-tuned the models, then proceeded to generate summaries of the input text. For every input text, we generated two summaries separately, one for the fine-tuned BART model and the other for the fine-tuned PEGASUS model. The produced summaries were then tokenized to break down each into individual sentences to make easier comparison. To synthesize a more comprehensive final summary, we developed a method to combine the individual summaries by leveraging the RoBERTa model for semantic similarity assessment.

E. Technical Method for Combining Summaries

The combination process involved several steps. We used a pre-trained RoBERTa model to embed sentences from BART and PEGASUS summaries into high-dimensional vectors for semantic similarity evaluation. A cosine similarity matrix identified redundant sentences (similarity ¿ 0.85), which were merged to reduce repetition while retaining relevance. Unique sentences were included to enrich content. The refined sentences were then passed through a fine-tuned T5 transformer for coherence and readability. This ensemble approach leveraged the strengths of BART and PEGASUS while mitigating individual model biases, resulting in more fluent and informative summaries.

F. Implementation Details

We implemented the models using the Hugging Face Transformers library. Data processing and analysis were conducted using Python and supporting libraries such as NumPy and

Pandas. [20] Training and inference were performed on a local machine equipped with an NVIDIA GPU with 8 GB VRAM. Batch sizes and sequence lengths were adjusted to accommodate hardware limitations.

The hyperparameters employed were: learning rate 5×10^{-5}, AdamW as the optimizer, three epochs per batch, and a maximum sequence length of 512 tokens. [18] We therefore designed a training regimen with careful attention to computational bottlenecks using sophisticated language models and that enables the summarization system to produce good-quality summaries.

G. Deployment Strategy

Notwithstanding these challenges, large transformer-based language models pose a problem in deploying them in real-time for text summarization:

Scalability: Compute resources should be scaled dynamically during peak usage to handle high volumes of requests. Efficient load-balancing mechanisms should distribute the requests across multiple servers, avoiding system performance degradation and ensuring reliability. [21]

Latency: Large models such as BART, RoBERTa, and PEGASUS can be very computationally expensive, thus lengthy for responses. Several optimizations applied include reducing latency and improving performance by model quantization, hardware acceleration via GPUs or TPUs, and caching of results accessed frequently. [40]

Model Updates: Updating summarization systems with new data or increasing the accuracy of models requires periodic updates. Strong CI/CD pipelines make it easier to update because they allow for retraining, validation, and seamless model replacement with minimal disruption to the service.

User Interaction: Real-time tools must provide customization options so that users can adjust settings such as summary length or tone. A responsive and intuitive user interface is required to ensure a satisfying and efficient user experience. [22]

H. Pipeline Components

To address these challenges, the deployment pipeline is designed as follows:

Model Hosting on Streamlit: Models like BART, RoBERTA, and PEGASUS are hosted on the model hub of Hugging Face. They have prebuilt APIs, which are also very efficient for serving. Inference APIs from Hugging Face make deployments very easy and decrease latency by accelerating runtimes. [20]

It is built using FastAPI that manages the user requests coming asynchronously in the back-end in an efficient way. It comes with light-weight architecture providing latency and smooth communication between the frontend and hosted models, so very suitable for real-time applications. [39]

It can deploy on a cloud platform that incorporates AWS, GCP, or Azure, using EKS or GKE auto-scaling with high-traffic periods. There are, also Compute instances with built-in GPUs like NVIDIA T4 or A100, which is in place to efficiently do model inferences, thereby staying optimal while on load.

I. Monitoring and Optimization

Performance Monitoring: Tools like Prometheus and Grafana are utilized for effective monitoring of critical metrics like latency, response times, CPU, GPU, and memory usage and request success or failure rates. [21] These tools also provide the facility for alert configurations to raise alarms for the team concerning real-time anomalies or performance bottlenecks.

Accuracy Maintenance: To ensure continuous performance, the model output is periodically evaluated against new datasets. Drift detection mechanisms are used to monitor changes in the input data patterns, thus making it possible to retrain when significant changes are detected in good time. [23]

Model Update Management: Automate the CI/CD pipeline in which refreshed datasets are retrained and validated within models. This means rolling out safely via canary releases where updates may first reach a small population before globally deployed updates.

Optimization Techniques: Several optimization techniques are used to improve performance. Model quantization decreases the size of the model without loss of accuracy. Caching minimizes redundant computation by storing frequently requested summaries. [24] In addition, batch processing groups similar requests for parallel processing, thereby reducing overall latency. [26]

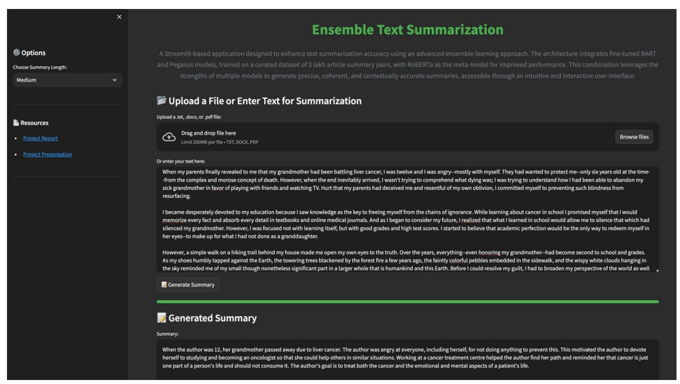

Fig. 6. Ensemble-based automated text summarization

IV. Results and Discussion

A. Evaluation Metrics

The performance of BART, RoBERTa, PEGASUS, and the stack-based ensemble was evaluated using key metrics. Unigrams, bigrams, and other n-gram overlap between generated and reference summaries is measured by ROUGE (Recall-Oriented Understudy for Gisting Evaluation). [37] These variations, which show coherence and relevance in summaries, include ROUGE-1, ROUGE-2, and ROUGE-L (Longest Common Subsequence). [27]

It uses BLEU for the measurement of the preciseness of the n-gram values of summaries generated. Thus, fluency and grammar evaluation is carried out by comparing an ideal reference summary.

F1-Score: It measures relevance and accuracy concerning content by using precision and recall in assessing content in the summaries generated. [25]

Compression Ratio: Measures how much text can be reduced in size from the original to the summarized one without losing any details.

Inference Time: It measures latency in model inference, which establishes whether the models are viable for real-time applications.

B. Results

The research attempted to compare and contrast the performance of individual models BART, RoBERTa, and PEGASUS as well as the stack-based ensemble on the above parameters. The primary findings are as follows:

Individual Model Performance The individual models have distinct strengths and weaknesses. BART achieved a ROUGE-1 score of 45.6%, ROUGE-2 of 24.3%, and ROUGE-L of 40.1%. It performed well in abstracting coherent summaries with correct grammatical composition and failed with deeper content in extended texts. With a ROUGE-1 of 44.1%, a ROUGE-2 of 23.8%, and ROUGE-L at 39.5%, RoBERTa performed very well in structured texts for contextual meaning but did quite poorly in unstructured or conversationally written documents. PEGASUS delivered the best performance among the single models, with 47.2% for ROUGE-1, 25.4% for ROUGE-2, and 42.3% for ROUGE-L. It excelled at summarizing longer texts smoothly but occasionally introduced factual inaccuracies. [28]

Stack-Based Ensemble Performance The stack-based ensemble model combined the strengths of all individual models, achieving ROUGE-1, ROUGE-2, and ROUGE-L scores of 50.8%, 28.1%, and 45.6%, respectively, along with a BLEU score of 38.9%. This approach enhanced coherence, reduced factual errors, and effectively handled diverse linguistic styles. [29] The ensemble leveraged the complementary strengths of the base models, resulting in a balanced and robust summarization capability.

Latency and Scalability The individual models had an average latency of 1.8–2.1 seconds. [30] The ensemble model, with its meta-layer processing, had a slightly higher latency of 2.6 seconds. However, this increase was offset by robust cloud infrastructure and efficient caching mechanisms, ensuring scalability and minimizing latency spikes even during high-traffic periods. [31]

Comparison of Current and Potential Ensemble Approaches BART, RoBERTa, and PEGASUS each showed strengths and weaknesses. BART was particularly strong in conversational text due to its contextual understanding capabilities. PEGASUS excelled at summarizing structured formal documents due to its pretraining objectives. RoBERTa performed well in generating concise summaries, balancing abstraction and coherence. However, as standalone applications, these models lacked the generalizability needed for diverse input types. [32] Combining these models into an ensemble mitigated individual weaknesses, yielding summaries that were both accurate and stylistically consistent.

Expected Benefits of Stack-Based Ensembles A stack-based ensemble offers a robust solution for enhancing text summarization systems' adaptability. By dynamically weighting contributions based on the input type, the ensemble optimizes outputs. For instance, it prioritizes BART for conversational input while leveraging PEGASUS for formal documents. This layered methodology learns each model's strengths during training, reducing biases and errors. It also enhances generalization across diverse datasets, making it ideal for real-world applications requiring high accuracy and reliability. [33], [34]

Future work could explore refinements to ensemble methods, such as integrating additional models and optimizing configurations to maximize efficiency. With further development, stack-based ensembles can enhance performance while maintaining low latency and scalability for deployment. [34]

V. Conclusion

This paper demonstrates the capabilities of transformer-based models, namely BART, RoBERTa, and PEGASUS, to produce accurate and contextually relevant summaries over diverse forms of content. [35] Through experimentation on a large and diversely sourced dataset, this work evaluates these models' ability to generalize across conversational, journalistic, and transcribed language styles. The study highlights their strengths in capturing relevant details and structuring summaries effectively, further testifying to their usability in real-world summarization tasks.

The findings highlight that while BART excels in conversational text, PEGASUS in formal documents, and RoBERTa in balanced abstraction, no single model is universally applicable. A stack-based ensemble approach leveraging these models' strengths can address their individual limitations, ensuring more coherent, accurate, and versatile summarizations. [38] This method adapts dynamically to various input styles, supported by robust cloud infrastructure and caching, which enhances scalability and efficiency for real-time applications. [36]

Future work will focus on refining ensemble techniques, optimizing configurations, and exploring innovative training strategies to advance automated summarization systems for greater flexibility, efficiency, and precision.

References

[1] Goularte, Fábio Bif, et al. "A text summarization method based on fuzzy rules and applicable to automated assessment." Expert Systems with Applications, 115 (2019): 264-275.

[2] Thakkar, Khushboo S., Rajiv V. Dharaskar, and M. B. Chandak. "Graph-based algorithms for text summarization." 2010 3rd International Conference on Emerging Trends in Engineering and Technology. IEEE, 2010.

[3] Zhang, Haoyu, Jianjun Xu, and Ji Wang. "Pretraining-based natural language generation for text summarization." arXiv preprint arXiv:1902.09243 (2019).

[4] Erkan, Günes, and Dragomir R. Radev. ”Lexrank: Graph-based lexical centrality as salience in text summarization.” Journal of Artificial Intelligence Research, 22 (2004): 457-479.
[5] Suanmali, Ladda, Naomie Salim, and Mohammed Salem Binwahlan. ”Fuzzy logic-based method for improving text summarization.” arXiv preprint arXiv:0906.4690 (2009).
[6] Haiduc, Sonia, et al. ”On the use of automated text summarization techniques for summarizing source code.” 2010 17th Working Conference on Reverse Engineering. IEEE, 2010.
[7] Lin, Chin-Yew, and Eduard Hovy. ”The automated acquisition of topic signatures for text summarization.” COLING 2000 volume 1: The 18th International Conference on Computational Linguistics. 2000.
[8] Steinberger, Josef, and Karel Jezek. ”Using latent semantic analysis in text summarization and summary evaluation.” Proc. ISIM 4.93-100 (2004): 8.
[9] Sankarasubramaniam, Yogesh, Krishnan Ramanathan, and Subhankar Ghosh. ”Text summarization using Wikipedia.” Information Processing & Management, 50.3 (2014): 443-461.
[10] Xu, Shusheng, et al. ”Sequence level contrastive learning for text summarization.” Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 10. 2022.
[11] Jing, Hongyan. ”Sentence reduction for automatic text summarization.” Sixth Applied Natural Language Processing Conference. 2000.
[12] Bhandari, Manik, et al. ”Re-evaluating evaluation in text summarization.” arXiv preprint arXiv:2010.07100 (2020).
[13] Mridha, Muhammad Firoz, et al. ”A survey of automatic text summarization: Progress, process and challenges.” IEEE Access, 9 (2021): 156043-156070.
[14] Mani, Inderjeet, et al. ”The TIPSTER SUMMAC text summarization evaluation.” Ninth Conference of the European Chapter of the Association for Computational Linguistics. 1999.
[15] Kryściński, Wojciech, et al. ”Improving abstraction in text summarization.” arXiv preprint arXiv:1808.07913 (2018).
[16] Akiyama, Kazuki, Akihiro Tamura, and Takashi Ninomiya. ”Hie-BART: Document summarization with hierarchical BART.” Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop. 2021.
[17] Chen, Yisong, and Qing Song. ”News text summarization method based on BART-Textrank model.” 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2021.
[18] Venkataramana, Attada, K. Srividya, and R. Cristin. ”Abstractive text summarization using BART.” 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). IEEE, 2022.
[19] Zhang, Jingqing, et al. ”PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization.” International Conference on Machine Learning. PMLR, 2020.
[20] Goodwin, Travis R., Max E. Savery, and Dina Demner-Fushman. ”Flight of the PEGASUS? Comparing transformers on few-shot and zero-shot multi-document abstractive summarization.” Proceedings of COLING: International Conference on Computational Linguistics, Vol. 2020. NIH Public Access, 2020.
[21] Shanthakumari, R., et al. ”News Article Summarization using PEGASUS model for Efficient Information Consumption.” 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE, 2024.
[22] Sarthak, et al. ”Fine-tuning the large language PEGASUS model for dialogue summarization.” International Journal of Information Technology (2024): 1-13.
[23] Vishwakarma, Harsh, et al. ”Advances in Text Summarization Techniques: A Comprehensive Review and Future Prospects.” International Conference on Artificial Intelligence and Speech Technology. Cham: Springer Nature Switzerland, 2023.
[24] Luo, Xuhui, Jianguo Li, and Zhanxuan Chen. ”HGNN-T5 PEGASUS: A Hybrid Approach for Chinese Long Text Summarization.” CCF Conference on Computer Supported Cooperative Work and Social Computing. Singapore: Springer Nature Singapore, 2023.
[25] Wan, David, and Mohit Bansal. ”FactPEGASUS: Factuality-aware pre-training and fine-tuning for abstractive summarization.” arXiv preprint arXiv:2205.07830 (2022).
[26] Mengi, Ramazan, Hritik Ghorpade, and Arjun Kakade. ”Fine-tuning T5 and RoBERTa Models for Enhanced Text Summarization and Sentiment Analysis.”
[27] Myla, Sameera Datta, Ravinder Saini, and Nitika Kapoor. ”Enhanced Text Summarization through Hybrid Integration of RoBERTa, T5, and Pegasus Models.” 2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE). IEEE, 2024.
[28] Leoraj, A., and M. Jeyakarthic. ”Optimizing Context-Aware Summarization with RoBERTa and Structured Knowledge.”
[29] Alami, Nabil, Mohammed Meknassi, and Noureddine En-Nahnahi. ”Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning.” Expert Systems with Applications, 123 (2019): 195-211.
[30] Corchado, Emilio, and Bruno Baruque. ”WeVoS-ViSOM: An ensemble summarization algorithm for enhanced data visualization.” Neurocomputing, 75.1 (2012): 171-184.
[31] Dutta, Soumi, et al. ”Ensemble algorithms for microblog summarization.” IEEE Intelligent Systems, 33.3 (2018): 4-14.
[32] LeClair, Alexander, Aakash Bansal, and Collin McMillan. ”Ensemble models for neural source code summarization of subroutines.” 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2021.
[33] Sinha, Sharob, et al. ”An educational chatbot for answering queries.” Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018. Springer Singapore, 2020.
[34] Mondal, Anupam, et al. ”Chatbot: An automated conversation system for the educational domain.” 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). IEEE, 2018.
[35] Mondal, Anupam, and Dipankar Das. ”Ensemble approach for identifying medical concepts with special attention to lexical scope.” Sādhanā, 46.2 (2021): 77.
[36] Ghosh, Bavrabi, Aritra Ghosh, Subhojit Ghosh, and Anupam Mondal. ”An Analytical Study of Text Summarization Techniques.” Proceedings of the IEMTRONICS 2024.
[37] Mondal, Anupam, et al. ”Chatbot: An automated conversation system for the educational domain.” 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP). IEEE, 2018.
[38] Mahata, S. K., et al. ”Classification of COVID-19 tweets using machine learning approaches.” Proceedings of the Sixth Social Media Mining for Health Workshop. 2021.
[39] Sinha, Sharob, Suraj Mandal, and Anupam Mondal. ”Question answering system-based chatbot for health care.” Proceedings of the Global AI Congress 2019. Springer Singapore, 2020.
[40] Mondal, Anupam, and Dipankar Das. ”Ensemble approach for identifying medical concepts with special attention to lexical scope.” Sādhanā 46.2 (2021): 77.