Multi-model Summarization on Extending T5 Transformer for Text, Audio, and Images

Kompella Bhargava Kiran(✉), ManojKumar Marumudi, Aniketh Varma Mudunuri, Sainadh Kondaveeti, Rohith Kondapalli, Kali I. Pradeep, and B. Ch. S. N. L. S. Sai Baba

Department of Computer Science and Engineering, Vishnu Institute of Technology, Bhimavaram, Andhra Pradesh, India
bhargav.kiran@gmail.com

Abstract. This research proposes an integrated approach to text summarization, extending the T5 transformer model's capabilities to include not only textual data but also audio and images. Leveraging the T5 transformer's versatility in natural language processing, we fine-tune the model on a substantial corpus of text documents for effective text summarization. Additionally, we integrate audio and image processing modules into the system, employing speech-to-text conversion for audio and feature extraction techniques for images. Our unified summarization framework seamlessly handles multi-modal data, allowing users to input text, audio, or images, and generates cohesive and informative summaries. This innovative approach enhances the T5 transformer's capabilities and addresses the growing demand for summarization in diverse data formats.

Keywords: T5 Transformer · Text Summarization · Multi-Model Summarization

1 Introduction

The process of condensing a textual document into a brief summary, highlighting its main concepts, is referred to as text summarization. This process involves utilizing software to reduce the length of a text document, providing an abstract or summary of the actual material. The main objective of text summarization is to extract crucial information from extensive text bodies. However, due to the investment of a lot of time on the nature of the process and the continuous influx of information, this task is becoming increasingly challenging.

There are two primary ways of approaching to text summarization:

Extractive Summarization & Abstractive Summarization: Extractive methods [1] means the kind of summarization [2] by selecting key phrases or sentences from the original text and piecing them together to create a condensed version as shown in Fig. 1. On the other hand, Abstractive Summarization [3] relies on condensing and paraphrasing portions of a document using sophisticated natural language techniques [4]. Abstractive

P. Pareek et al. (Eds.): IC4S 2024, LNICST 599, pp. 52–64, 2025.
https://doi.org/10.1007/978-3-031-77081-4_6

methods [5], particularly those based on the advanced models of deep learning [6, 7], can generate fresh words and phrases to accurately reflect the source text's content, helping overcome grammatical errors as shown in Fig. 2.

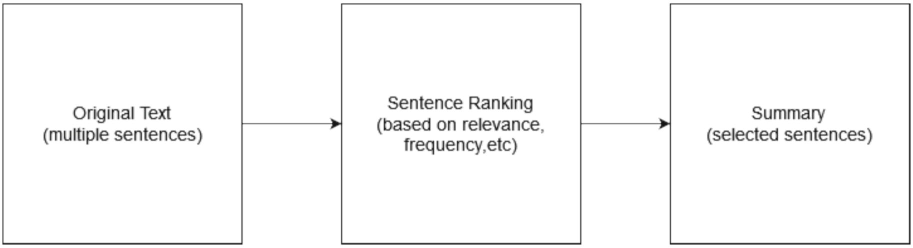

Fig. 1. Extractive Text Summarization

Fig. 2. Abstractive Text Summarization

The inclusion of three distinct data formats—PDF files, audio transcripts, and text extracted from images—extends the project's applicability. PDFs often contain rich textual information, and our project focuses on extracting and summarizing key insights from these documents [8]. Furthermore, the conversion of spoken words in audio files to text enables the summarization of spoken content, opening avenues for applications in podcast summarization, voice note analysis, and more.

The overarching goal of our research is to compare and develop a tool for automatic summarization, leveraging principles of natural language processing. With the growing need for automatic summarization to eliminate manual labor, the focus is on creating a tool that can efficiently produce summaries from input formats like text, images, audio. This research aims to extract or generate a summary from lengthy documents, catering to various applications such as writing headlines for news channels, creating briefings, note-taking, and generating highlights for speeches. The advanced techniques of Deep learning, especially in the field of natural language processing, plays a crucial role in achieving the objectives of this research.

2 Literature Review

Chaitan Raju, Subash Voleti, and Teja Rani M. [9] collaborated on a project focused on text extraction and summarization through natural language processing and the Google Text-to-Speech model. One drawback of their approach was its reliance on extractive summarization, lacking paraphrasing capabilities, and utilizing the TextRank [10] algorithm. In contrast, T5, a transformer-based model, employs graph-based methodologies and heuristics. T5 demonstrates superior ability in capturing contextual relationships and dependencies among words within sentences or documents compared to TextRank. Additionally, T5's training on extensive datasets enables it to develop nuanced language representations, whereas TextRank depends on heuristics alone. T5's adaptability for specific domains or tasks through fine-tuning sets it apart from TextRank, rendering it a more sophisticated and efficient approach to text summarization.

Batra, Pooja; Chaudhary, Sarika; Bhatt, Kavya; Varshney, Saloni; and Verma, Srishti conducted a project titled "A Review - Abstractive Text Summarization Techniques using the NLP." An identified limitation of their approach was its reliance on abstractive summarization with LSTM [11] and RNN models [12]. Transformers, however, surpass LSTM and RNN architectures in text summarization by adeptly learning the intricate relationships between words in sentences or documents, leading to more precise and informative summaries.

2.1 Automated Document Summarization

The automation of document summarization within the field of natural language processing (NLP) has become a focal point for researchers. Initially, methods relied on statistical features such as frequency and centrality to rank sentences [13] for summarization. With the introduction of machine learning in the 1990s, a shift towards extractive summarization occurred, wherein essential phrases were extracted from texts without full semantic comprehension. Deep learning techniques [14], including encoder-decoder classifiers and LSTM networks [15], furthered progress in extractive summarization. Advancements were also made in abstract summarization [16], utilizing RNNs [17] and LSTM models [18]. The emergence of transformer [19] architectures like BERT and BART in 2017 revolutionized text summarization, marking a significant milestone in NLP progress [20].

2.2 Semantic Style Rendering

Semantic Style Rendering within NLP involves modifying text styles [21] while preserving original content, crucial for applications like creative writing and sentiment analysis. Initially, manual rule-based methods were predominant but were limited in adapting to evolving styles. The emergence of deep learning, particularly seq2seq models [22], revolutionized the field by enabling encoding [23] and generation of text. More recently, generative adversarial networks (GANs) [24] have been employed, with a generator crafting desired styles and a discriminator distinguishing generated from authentic text. These advancements automate style modification, improving adaptability to changing linguistic trends and broadening applicability across various NLP tasks.

3 Design

The following steps in Fig. 3 depicts the end to end implementation of the proposed application:

Start: The process begins with selecting the type of data to summarize. The available options are TEXT, IMAGE, PDF, and AUDIO.

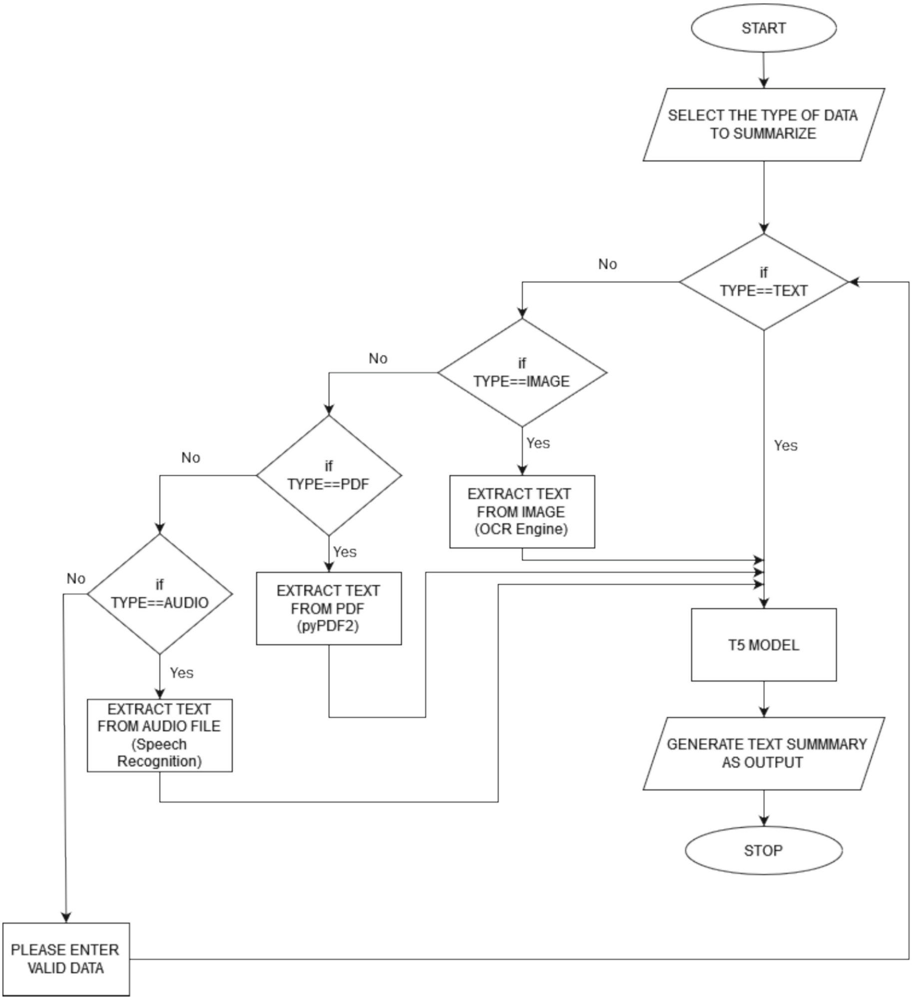

Fig. 3. Flow of end-to-end application

Image and PDF Data: For IMAGE and PDF data types, the application first extracts text from the respective files. Converting images to text via OCR involves managing font, size, and image quality variations. The process includes scanning, preprocessing for clarity, recognizing multi-language scripts, employing pattern matching or feature extraction for accurate text recognition, and converting the extracted text into digital formats, often creating annotated PDFs to display the original image alongside the recognized text.

After extracting text, it follows the same path as text data: reaching the T5 model for summary generation.

Audio Data: When the selected data type is AUDIO, the application extracts text from the audio file. Again, the process continues to the T5 model for generating a text summary. Converting spoken audio into text involves two key functions: 'convert_mp3_to_wav' and 'audio_to_text'. The former converts MP3 to WAV, while the latter transcribes speech to text using Google's service. The process encompasses typical stages: audio input, preprocessing, feature extraction, acoustic and language modeling, decoding, post-processing, and output. These stages collectively enable accurate transcription and formatting of spoken content for various applications.

Error Handling: At the bottom left corner, there's an error handling mechanism labeled as "PLEASE ENTER VALID DATA." This ensures that the user provides valid input.

4 Implementation

Our project aims to summarize text comprehensively, showing transformer models' adaptability. We provide a unified solution for extracting and summarizing content from diverse sources.

4.1 Dataset

The CNN/DailyMail Dataset comprises over 300k distinct news articles in English, authored by some journalists from CNN and also from the Daily Mail. Initially designed for machine-based reading and comprehension, as well as abstractive question answering through which the current iteration of the dataset accommodates both extractive and abstractive summarization tasks.

Data Fields

1. **Unique identifier (ID):** A sequence of characters presenting the SHA1 hash in hexadecimal format derived from the URL source of the story.
2. **Article Body:** A textual representation encompassing the content of the news report.
3. **Article Highlights:** A textual representation capturing the salient features of the article, articulated by its author.

Data Splits. The CNN/DailyMail dataset comprises three divisions: training, validation, and testing. Provided in Table 1. are the statistics pertaining to Version 3.0.0 of the dataset.

Our implementation involves extracting textual content from PDFs, converting audio to text, and using OCR for text in images. This feeds into the T5 model, known for NLP tasks. Text summarization is vital for distilling large volumes into concise summaries. T5 handles different input and output formats effectively. Our project's inclusion of PDFs, audio transcripts, and OCR for images extends its applicability. We address technical challenges in handling varied data and emphasize efficient information summarization. The synergy between T5 and diverse input formats reflects NLP's evolution towards inclusive solutions.

Table 1. The statistics pertaining to Version 3.0.0 of the dataset

Dataset	Number of Instances in Split
Training Dataset	287k
Validation Dataset	13k
Testing Dataset	11k

4.2 Preprocessing and Tokenization

Preprocessing plays a crucial role in T5 text summarization, ensuring that the input data is appropriately formatted for the model. Here are key preprocessing steps commonly applied in T5 text summarization:

Data Cleaning. It's common to perform standard text cleaning steps, such as removing irrelevant characters, punctuation, or special symbols, to enhance the quality of the input data.

Input Formatting. T5 is designed for a kind of text to text framework, where both the input and the output are treated as sequences. The input text is typically formatted to start with a specific task prefix, such as "summarize:" to instruct the model to perform text summarization.

Tokenization. The input text is tokenized using the T5 model's tokenizer. This involves breaking down the text into tokens, which could be words, subwords, or characters. Let Tokenize(x) represent the tokenization function, where x is the input text.

x = "The T5 model is a powerful tool for text summarization."

Tokenize(x) = ["The", "T5", "model", "is", "a", "powerful", "tool", "for", "text", "summarization", "."]

4.3 Proposed Framework

The project aims to develop a web-based tool for text summarization utilizing abstractive techniques to extract pertinent information from documents, while filtering out irrelevant content. This involves leveraging the CNN Daily Mail dataset alongside T5 (Text - to - Text Transfer Transformer), an advanced model that converts language complexities into a text to text format. CNN/DailyMail dataset comprises 311,971 recent articles, partitioned into training, testing, and validation sets. Prior to model construction, the dataset undergoes preprocessing, including tokenization, lowercase conversion, and removal of stopwords and quotations.

The T5 model undergoes training using the training dataset and is evaluated using Rouge Score. Following the training phase, testing is conducted using reserved text data. Subsequently, the model is implemented to develop a web application, enabling users to generate summaries from textual inputs as depicted in the Fig. 4.

4.4 Encoding and Decoding in Text Summarization Using T5 Model

In the era of text summarization using the T5 (Text-To-Text Transfer Transformer) model [25], encoding and decoding refer to crucial steps in the process of transforming raw text into a format suitable for the model and obtaining meaningful summaries as output.

In the T5 model architecture [26], encoder layers process encoded input text through transformer mechanisms, attending to the input's vectors to identify important words and relationships. The decoder then generates the summary token by token [27], attending to the encoded input to ensure relevance and coherence. Decoding can utilize strategies like beam search or greedy decoding, with beam search considering multiple candidate tokens and greedy decoding selecting the token with the highest probability. This conditional generation mechanism allows T5 to focus on salient information while crafting summaries [28]. Overall, T5's encoder and decoder [29] components work in tandem to transform raw text into concise and informative summaries.

Encoding. Encoding involves converting the input text into a format of numerical representation that the model can make mathematics operations on. This is achieved through tokenization, where the text is partitioned down into smaller units termed to be tokens, and each token is mapped to a unique numerical ID. Encoding allows the model to process textual information in a structured numerical format. The encoded representation captures the semantic and contextual information of the input text.

$$\mathrm{Encode}(x) = [101, 1109, 2047, 1110, 170, 3102, 6991, 1111, 3793, 113, 119, 102]$$

Numerical Representation. Each token is assigned a unique numerical ID. The entire sequence of the tokens is then coined as a tensor, a numerical matrix, where each entry corresponds to the ID of the corresponding token. Let Encode(x) represent the encoding function, which converts tokens into numerical IDs.

Decoding. Decoding is the reverse process of encoding. It involves converting the model's numerical output back into human-readable text. In the context of text summarization, decoding is applied to the generated summary.

Token Decoding. The model's output, typically a sequence of token IDs, is decoded using the inverse mapping of the tokenizer. Each token ID is converted back into its corresponding word or subword. Let TokenDecode(y) represent the token decoding function, where y is the model's output sequence of token IDs.

$$y = [101, 1109, 2047, 1110, 170, 3102, 6991, 1111, 3793, 1130, 119, 102]$$

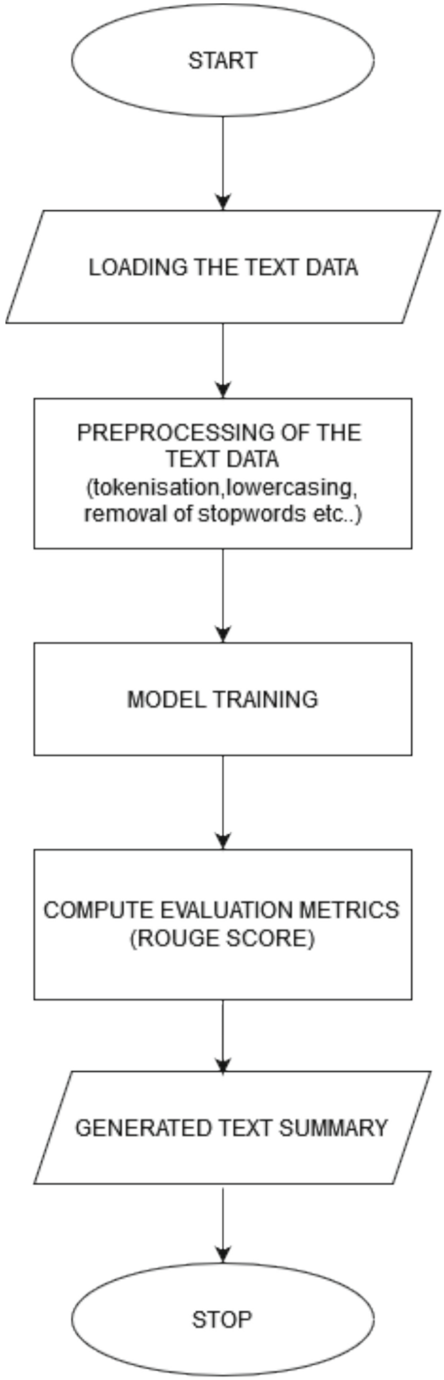

Fig. 4. Flow of Summarization Model

TokenDecode(y) = ["The", "T5", "model", "is", "a", "powerful", "tool", "for", "text", "summarization", "."]

Reconstruct(y) = "The T5 model is a powerful tool for text summarization.

Decoding translates the model's numerical output into a human-understandable format. The decoded summary provides a concise representation of the key information in the input text. TextReconstruction: The decoded tokens are combined to reconstruct the summarized text. Let Reconstruct(y) represent the function that combines decoded tokens to reconstruct the summarized text.

5 Evaluation

Evaluation metric deals with various versions of Rouge scores that exist, such as Rouge-N, Rouge-L, Rogue-S, Rouge-N evaluates the overlap of n-grams between the generated summary and the reference summary, while Rouge-L computes the longest common subsequence between them. Rouge-S, a derivative of Rouge, concentrates on sentence-level similarity.

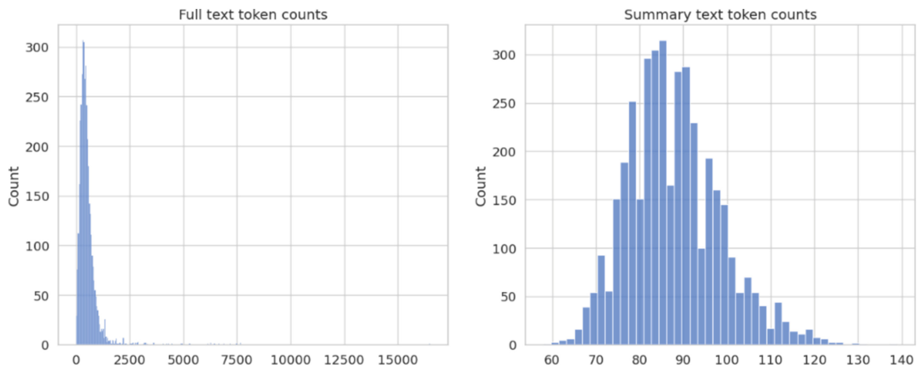

Fig. 5. The chart representation for the count of tokens presents before and after the summarization

In Fig. 5, the image consists of two histograms side by side comparing "Full text token counts" on the left and "Summary text token counts" on the right. - Both histograms are plotted against count on y-axis but have different scales for x-axis due to differing lengths of full texts and summaries. - The histogram for full text shows a sharp peak at the beginning indicating most articles have a token count less than 2500; it then sharply declines. - In contrast, summary text histogram has a more normal distribution centered around approximately 100 tokens, indicating summaries are relatively concise and consistent in length.

ROUGE-1 measures how much the words in the candidate summary match those in the reference summary, using individual words (uni-grams). Recall shows the proportion of words from the reference summary that are in the candidate summary, while Precision shows how many of the words in the candidate summary are actually in the reference summary.

$$\text{ROUGE} - 1 = \frac{\sum_{\text{summary} \in \text{Model}} \text{Count of Unigrams(summary)} \cap \text{Count of Unigrams(Reference)}}{\sum_{\text{summary} \in \text{Model}} \text{Count of Unigram(summary)}} \quad (1)$$

ROUGE-2 evaluates similarity by looking at pairs of consecutive words (bi-grams) in both the candidate and reference summaries.

$$\text{ROUGE} - 2 = \frac{\sum_{\text{summary} \in \text{Model}} \text{Count of Bigrams(summary)} \cap \text{Count of Bigrams(Reference)}}{\sum_{\text{summary} \in \text{Model}} \text{Count of Bigram(summary)}} \quad (2)$$

ROUGE-L calculates Precision and Recall by finding the longest sequence of words (LCS) that appear in both the candidate and reference summaries, even if they're not right next to each other.

ROUGE - 1. Assesses the unigram correspondence between the summaries generated by the model and the collection of human generated reference summaries.
ROUGE - 2. Evaluates the bigram match between the summaries produced by the model and a group of human-generated reference summaries.
ROUGE - 3. Determines the longest common subsequence agreement between the model-generated summaries and the reference summaries created by humans.

$$ROUGE - L = \frac{\sum_{summary \in Model} Longest\ Common\ Subsequence(summary,\ Reference)}{Total\ Number\ of\ References\ Summaries} \quad (3)$$

The Rouge scores from Table 2 highlight varying summarization capabilities, with T5 exhibiting superior performance, achieving 36.53 on drug reviews and 41.2 on CNN/DailyMail, compared to an abstractive model's 32.65 and a seq-to-seq baseline's 28.83. Specifically focusing on abstractive text summarization using the CNN/Daily Mail dataset, the T5 Transformer model achieves a ROUGE-L score of 41.2 after training on 287,113 news items, indicating high-quality generated summaries and showcasing its efficacy. Comparison with other datasets [30], such as the "Drug reviews dataset," where the T5 model achieved a ROUGE-L value of 36.53, further emphasizes its effectiveness. Strategies for improvement include utilizing larger training datasets, employing more complex model architectures, and fine-tuning with domain-specific data. Overall, the T5 transformer model's adaptability, efficiency, and remarkable performance position it as a leading solution for abstractive text summarization tasks.

Table 2. Rouge Values Comparison for Text Generation Models

Model	Rouge Value
Abstractive Model	32.65
Seq-to-seq attn baseline	28.83
T5 with Drug reviews dataset	36.53
T5 with CNN/DailyMail dataset	41.2

6 Conclusion

In summary, the text summarization project utilizing the T5 model has delivered promising outcomes, demonstrating the model's robust performance across various data formats including PDFs, audio transcripts, and images. The project's success is highlighted by its high ROUGE scores, indicating the model's effectiveness in producing concise and informative summaries. While the T5 model showcases versatility and adaptability, some limitations have been noted, particularly in dealing with complex formatting and noisy audio transcripts. To enhance the project's capabilities, future efforts will concentrate on refining preprocessing methods, exploring multimodal summarization, and addressing deployment considerations for real-time usage. Continuous evaluation of model performance and the development of an interactive user interface will further contribute to the project's advancement and usability. Ultimately, this project establishes a groundwork for advancing text summarization techniques and opens avenues for future research in natural language processing and multimodal data analysis.

Future Work. In addition to future work, several avenues for improvement and expansion in the domain of text summarization are proposed. Advanced preprocessing techniques are suggested, aiming to enhance Optical Character Recognition (OCR) for extracting text from images and to improve the handling of complex PDF layouts, thereby elevating data quality. The idea of incorporating augmented training data is explored to expose the summarization model to a broader spectrum of linguistic styles and content structures. Experimentation with fine-tuning approaches is recommended, particularly tailored to address challenges encountered during audio transcript summarization. Moreover, the potential extension of the project to support multimodal summarization, integrating information from various data sources for generating more comprehensive and contextually rich summaries, is proposed. A focus on user interaction is emphasized through the development of an interactive user interface, potentially integrating advanced visualization tools to facilitate intuitive exploration and evaluation of summarization results. Strategies for performance optimization, including distributed training and model compression techniques, are suggested to manage resource constraints effectively. Continuous evaluation mechanisms are advocated for, ensuring ongoing assessment of the model's performance with new datasets and iterative adjustments to the training process. Lastly, deployment considerations such as model inference speed are highlighted to ensure real-time applicability in scenarios where rapid summarization is crucial. By addressing these aspects, the field of text summarization can progress towards more robust, versatile, and user-friendly solutions.

References

1. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
2. Ferreira, R., et al.: Assessing sentence scoring techniques for extractive text summarization. Expert Syst. Appl. **40**, 5755–5764 (2013)
3. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, vol. 27 (2014)

4. Bollegala, D., Okazaki, N., Ishizuka, M.: A machine learning approach to sentence ordering for multidocument summarization and its evaluation. In: Dale, R., Wong, K.F., Su, J., Kwong, O.Y. (eds.) IJCNLP 2005. LNCS, vol. 3651, pp. 624–635. Springer, Heidelberg (2005). https://doi.org/10.1007/11562214_55
5. Chopra, S., Auli, M., Rush, A.M.: Abstractive sentence summarization with attentive recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 93–98 (2016)
6. Bonthu, S., Rama Sree, S., Krishna Prasad, M.H.M.: Automated short answer grading using deep learning: a survey. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2021. LNCS, vol. 12844, pp. 61–78. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84060-0_5
7. Anand, D., Wagh, R.: Effective deep learning approaches for summarization of legal texts. J. King Saud Univ. Comput. Inf. Sci. **34**, 2141–2150 (2022)
8. Nallapati, R., Zhai, F., Zhou, B.: SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the AAAI Conference on Artificial Intelligence (2017)
9. Voleti, S., Raju, C., Rani, T., Swetha, M.: Text summarization using natural language processing and google text to speech API. Int. Res. J. Eng. Technol. (IRJET) **7**, 3705–3711 (2020)
10. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
11. Hou, L., Hu, P., Bei, C.: Abstractive document summarization via neural model with joint attention. In: Huang, X., Jiang, J., Zhao, D., Feng, Y., Hong, Y. (eds.) NLPCC 2017. LNCS, vol. 10619, pp. 329–338. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73618-1_28
12. Nayeem, M.T., Fuad, T.A., Chali, Y.: Neural diverse abstractive sentence compression generation. In: Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D. (eds.) ECIR 2019, Part II. LNCS, vol. 11438, pp. 109–116. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15719-7_14
13. Erkan, G., Radev, D.R.: LexRank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
14. Conroy, J.M., O'leary, D.P.: Text summarization via hidden Markov models. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 406–407 (2001)
15. Alguliev, R.M., Aliguliyev, R.M., Isazade, N.R.: Multiple documents summarization based on evolutionary optimization algorithm. Expert Syst. Appl. **40**, 1675–1689 (2013)
16. Baralis, E., Cagliero, L., Mahoto, N., Fiori, A.: GraphSum: discovering correlations among multiple terms for graph-based summarization. Inf. Sci. **249**, 96–109 (2013)
17. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252 (2016)
18. Mohsen, F., Wang, J., Al-Sabahi, K.: A hierarchical self-attentive neural extractive summarizer via reinforcement learning (HSASRL). Appl. Intell. **50**, 2633–2646 (2020)
19. Rush, A.M., Chopra, S., Weston, J.: A neural attention model for abstractive sentence summarization. arXiv preprint arXiv:1509.00685 (2015)
20. Zhang, H., Xu, J., Wang, J.: Pretraining-based natural language generation for text summarization. arXiv preprint arXiv:1902.09243 (2019)
21. Ban, H.: Stylistic characteristics of English news. In: Proceedings of the 5th Japan-Korea Joint Symposium on Emotion and Sensibility, Daejeon, Korea, June 2004 (2004)
22. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019)

23. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 1–67 (2020)
24. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
25. Zolotareva, E., Tashu, T.M., Horváth, T.: Abstractive text summarization using transfer learning. In: ITAT, pp. 75–80 (2020)
26. Ranganathan, J., Abuka, G.: Text summarization using transformer model. In: 2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 1–5 (2022)
27. Bonthu, S., Sree, S.R., Prasad, M.H.M.K.: Improving the performance of automatic short answer grading using transfer learning and augmentation. Eng. Appl. Artif. Intell. **123**, 106292 (2023)
28. Sukhavasi, V., Dondeti, V.: Effective automated transformer model based sarcasm detection using multilingual data. Multimed. Tools Appl., 1–32 (2023)
29. Laskari, N.K., Reddy, K.A.N., Indrasena Reddy, M.: Seq2Code: transformer-based encoder-decoder model for python source code generation. In: Congress on Intelligent Systems, pp. 301–309 (2022)
30. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 (2017)