Legal Text Summarization Using Bart and Explainable AI Techniques

Teh Xiao Thong[1], Halizah Basiron[1*], Abdul Syukor Mohamad Jaya[1], Fitrah Rumaisa[2]

[1]Fakulti Kecerdasan Buatan dan Keselamatan Siber, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, 76100, Melaka, Malaysia

[2]Universitas Widyatama, Jl. Cikutra 204 A, Bandung, 40125, Indonesia

***Corresponding Author**

DOI: https://dx.doi.org/10.47772/IJRISS.2025.91100607

Received: 11 December 2025; Accepted: 18 December 2025; Published: 26 December 2025

ABSTRACT

Summarizing lengthy documents, especially in the legal domain, posed significant challenges for humans and automated systems. Human efforts entailed considerable time and effort, while automated systems sometimes faltered in decision-making, leading to ambiguity in the generated summaries. This research explored the use of text summarization in legal documentation coupled with an explainability feature. It addressed the challenges of condensing lengthy legal texts and improving automated summarization systems' transparency. The research involved gathering legal documents, developing a Bidirectional and Auto-Regressive Transformers (BART) summarization model, and integrating explainability within the system, visualizing the attention mechanism. The system performance, which included BERTScore, cosine similarity, and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score between human-generated and system-generated summaries, and evaluation by target users, led to several engaging insights on legal summarization. The model demonstrated moderate performance, where user feedback indicated satisfaction with its functionality but highlighted the need for user interface improvements. Future improvements were suggested, including refining model training, enhancing the user interface, and adding features like adjustable summary lengths and language translation.

Keywords: Text summarization, Natural language processing, Explainable artificial intelligence, Legal Field, Bidirectional and auto-regressive transformers

INTRODUCTION

Reviewing legal documents such as supreme court case documents often require specialized knowledge, and reading through the entire document to capture the critical information is time-consuming. As the volume of legal documents increases, extracting essential details without delving into the whole content becomes crucial. Hence, summarization provides a solution by providing flexibility and convenience to readers. In addition, explainable artificial intelligence (XAI) can ensure the system produces concise summaries and provides transparent justifications for the decisions made, enhancing trust and comprehension for legal professionals.

Legal professionals often struggle with obstacles when driving through documentation, which includes time and effort. Reading and comprehending pages of documents can be a cumbersome process, which might lead to potential oversights or missed critical details. Moreover, clear explanations are necessary and crucial behind automated summarization, where transparency and accountability are paramount. This research aims to develop a system that can summarize legal documentation with suitable explanations. The system must ensure the generated summaries accurately capture the critical points of the original document and provide a clear rationale for including specific information.

The domain of this research falls within the intersection of several fields: the legal domain, natural language processing (NLP), machine learning, and explainable artificial intelligence (XAI). This research operates within the legal fields, explicitly dealing with legal case documents, and involves NLP techniques for processing and

analyzing textual data. Incorporating explainability features suggests using machine learning and XAI algorithms for summarisation and providing insights or explanations about the summarized content. This research also delves into interpretability and transparency aspects to ensure the generated summaries and explanations are understandable and trustworthy to stakeholders. The related work has been separated into two major categories: existing systems and techniques, and the kinds of literature are reviewed respectively.

Existing Systems

A hybrid method for automatic text summarization of legal cases using k-means clustering techniques and term frequency-inverse document frequency (TFIDF) word vectorizer is proposed by Varun Pandya [1]. The process involves data preprocessing to clean the document, clustering similar sentences using k-means, and extracting sentences to form a summary. The k-means algorithm groups sentences, which are then vectorized with TF-IDF. Clustering minimizes intra-cluster distances and maximizes inter-cluster distances, with optimal clusters determined. Sentences are ranked based on TF-IDF score and title similarity, with top-ranked sentences selected for the final summary. The dataset comprises Australian legal cases from Auslii. Evaluation is done by using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metrics to compare results with three automated tools. Pandya's method showed promising results, where the proposed method performs favourably well against other existing methods, as detailed in a comparative table.

Anand and Wagh have also proposed simple generic techniques using neural network architecture, which are feed-forward neural networks (FFNN) based summary and long short-term memory (LSTM) based summary [2]. Their approaches require no manual features or domain knowledge and can be applied across various domains. The process involves generating labelled data using summary information from court judgment headnotes and utilizing this data to extract essential sentences for summarization. Different similarity techniques are employed to compute sentence labels, with sentence embeddings (SSE) performing best. FFNN transforms sentences into vectors, calculates probabilities, and selects the top-ranked sentences for the summary. LSTM, combined with convolutional neural networks (CNN), selects sentences with the highest importance likelihood based on LSTM output scores. Evaluation using ROUGE scores on Supreme Court of India judgment documents demonstrates the effectiveness of both methods. The result table shows that LSTM performs better in many cases.

Research on the comparison of extractive and abstractive legal case document summarization was done by Shukla and his team [3]. This research aims to analyze the performance of various summarization methods on legal case judgment documents and explore effective evaluation techniques. Extensive experiments with several abstractive and extractive summarizations, including supervised and unsupervised approaches, have been carried out over three legal summarization datasets. Some examples of the methods are Luhn, Pacsum_bert, Maximal Marginal Relevance (MMR), Bidirectional and Auto-Regressive Transformers (BART), Bidirectional Encoder Representations from Transformers - Bidirectional and Auto-Regressive Transformers (BERT-BART), and Legal-Pegasus etc. The datasets, Indian-Abstractive, Indian Extractive, and UK-Abstractive datasets, are developed from case documents from the Indian and United Kingdom Supreme Courts. The analyses, including ROUGE, BERTScrore, and evaluations by legal practitioners, aim to provide insights into legal summarization and long document summarization in general, contributing to advancements in this field.

Shifting the focus to another system, the Neural Networks for Text Summarization, with a Keras implementation of an attention-based sequence-to-sequence (seq2seq) model, is explored, emphasizing the success of the attention mechanism in the context [4]. Like other systems, data preprocessing is done as the first implementation step. A model with encoder-decoder architecture, which has global attention, is built, and an embedding layer to convert words into appropriate vector representations is used, learning along with the seq2seq model. Attention mechanisms in encoder-decoder neural networks enable the generation of a context vector at each timestep by considering the decoder's current hidden state and a subset of the encoder's hidden states. The dataset used in this study is the Amazon Fine Food dataset found on Kaggle. Since the original and generated summaries are short, the performance evaluation is done by comparing both.

An automatic abstractive text summarization model based on a hybrid attention mechanism has been introduced by Zhe Wang, where it incorporates a sentence-level attention mechanism to guide word-level attention distribution, adjusting the weight of sentence-level attention to mitigate high variance issues in word-level attention for shorter documents [5]. The methodology of this study introduces a hybrid-attentional model using encoder-decoder networks with recurrent neural networks (RNN). It incorporates attention mechanisms to improve decoder focus and a pointer-generator network for word generation or copying. Additionally, a dynamic hybrid attention mechanism adjusts attention values at both word and sentence levels to enhance summary quality based on document length. Evaluation of the approach using the ROUGE score on a large scale Chinese short text summarization (LCSTS) dataset demonstrates the effectiveness of the proposed method in capturing critical information and generating concise summaries. Other research on automatic abstractive text summarization using deep learning can also be refer to [6] and [7].

Techniques

Text summarization creates a short, accurate, and fluent summary of a longer text document [8]. This process is crucial for managing the vast volume of online text data, facilitating more efficient discovery and consumption of relevant information. There are two main forms of text summarization methods: abstractive and extractive. Extractive summarization combines existing sentences without any alterations to create a summary, while abstractive summarization involves text generation where the machine writes its own sentences [9],[10]. Extractive summarization is more rigid due to directly copying sentences from the source text, potentially resulting in awkward reading. Conversely, text generation in abstractive summarization initiates a better human writing style, enhancing coherence and readability with concise and coherent output. There are several prominent examples of both methods: Luhn, Latent Semantic Analysis (LSA), TextRank, LexRank, PositionRank, and TopicRank for extractive summarization. In contrast, abstractive summarization includes BART and pretraining with extracted gap sentences for abstractive summarization (PEGASUS) [11].

BART, a denoising autoencoder for the pretraining sequence-to-sequence model, is introduced by Mike Lewis and his team [12]. BART is trained to reconstruct original text from corrupted versions using a Transformerbased architecture, which can be seen as a generalization of models like BERT and generative pre-trained transformer (GPT). The study evaluates various text corruption methods and demonstrates BART's effectiveness in text generation, comprehension, abstractive dialogue, question-answering, summarization, and machine translation. Additionally, ablation experiments within the BART framework are conducted to assess factors influencing end-task performance. On the summarization task, BART shows an outperformance over two datasets (CNN/DailyMail and XSum) surpassing other existing methods. The resulting summaries are fluent and grammatically correct, indicating that BART's pretraining has effectively learnt a robust blend of natural language comprehension and generation.

Erkan and Radev have presented a stochastic graph-based method for determining the relative importance of textual units, particularly in text summarization [13]. The technique is named LexRank. It computes sentence importance based on eigenvector centrality in a graph representation of sentences, using intra-sentence cosine similarity. In this study, LexRank is implemented into the MEAD summarization system [14]. The dataset used in the experiments consists of DUC 2003 and 2004 data sets, which involve generic summarization of news document clusters. For evaluation, the ROUGE metric, specifically ROUGE-1, which represents the unigrambased ROUGE score, was used as it aligns closely with human judgements.

A study by Kamya Singh and his team investigates using BERTbased techniques for summarization and sentence similarity checks to enhance important question-answering systems [15]. The proposed approach combines BERT-based summarization with semantic similarity checking to extract key information and predict crucial questions. Experiments on benchmark datasets have been done, showing that this method surpasses traditional machine learning and deep learning techniques, achieving state-of-the-art performance. The approach was also effective in real-world applications like medical diagnosis, legal case analysis, and financial forecasting.

There are several methods to evaluate the performance of a text summarization system, and one of the approaches is the ROUGE score. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation score [16],[17]. It

is a set of metrics commonly used for text summarization tasks to generate a concise summary of a longer text automatically. It was designed to evaluate the quality of machine-generated summaries by comparing them to human reference summaries. ROUGE has variants like ROUGE-N focusing on n-gram overlap, ROUGE-L on the longest common subsequence (LCS), and ROUGE-S on skip-bigram overlap. The ROUGE score ranges from 0 to 1, with higher values indicating better summary quality. It's widely used for objectivity but may not fully capture semantic meaning or coherence.

Another method used to evaluate the quality of text summarization is BERTScore [18],[19]. This method measures the similarity between the summary and the original text. It addresses issues encountered by n-grambased metrics using contextualized token embeddings from models like BERT to compute similarity. The process involves representing sentences with contextual embeddings, measuring cosine similarity, token matching for precision and recall, considering word importance using IDF, and rescaling values for readability. For a basic level BERTScore calculation, the output will be precision, recall, and F1 score [20]. BERTScore enhances text similarity measurement, making it more accurate and balanced, with potential applications in various domains of natural language processing. However, this method has its pros and cons. For example, BERTScore can handle different types of texts, but it can be biased towards models that are more similar to its underlying model.

In essence, ensuring the transparency and interpretability of the summaries is crucial, where explainable artificial intelligence (XAI) plays an important role. Some examples of XAI methods in NLP include visualizing attention mechanisms in neural networks, generating textual explanations for model predictions, and interpreting the reasoning behind the models' decision-making process [21]. Vaswani introduced the attention mechanism in 2017 [22]. In traditional Deep Learning models like LSTMs and RNNs, longer inputs pose challenges for retaining relevant information, prompting the need for attention mechanisms to signal the model about focus areas [23]. However, transformer models, utilizing self-attention across all encoder and decoder layers, circumvent this issue [23]. Attention mechanisms are widely used in text summarization across diverse domains like news, reviews, scientific papers, legal documents, and social media posts, where models such as the Pointer-generator network, Transformer, and BART exemplify this trend [24].

Research on an open-source tool for visualizing attention mechanisms in transformer-based language models is proposed by Jesse Vig [25]. The tool offers three levels of granularity: attention head, model, and neuron views. Its application has been demonstrated on BERT and GPT-2 models. The tool aids in interpreting model decisions and identifying patterns, such as model bias detection, recurring pattern identification, and neurons to model behaviour linkage. This allows a comprehensive understanding of how the model attends to different input parts and how individual neurons contribute to attention computation. It enhances model interpretability, enables targeted improvements through user manipulation, and offers versatility for various analysis tasks and model types.

On the other hand, a theoretical analysis of local interpretable model-agnostic explanations (LIME) has been done by Garreau and Luxburg [26]. This explainer is commonly used to provide interpretability to machine learning models. The study derives closed-form expressions for the coefficients of the interpretable model when the function to explain is linear, demonstrating that LIME can uncover meaningful features proportional to the function's gradient. It aids in understanding model decisions, improving trust, and facilitating compliance with regulations. However, it also highlights potential limitations of LIME where poor parameter choices may cause the algorithm to overlook important features.

METHODS

The system proposed in this research is the text summarization system with the explainability feature using NLP and machine learning techniques. Figure 1 shows the architecture of the proposed system. It involved backend and frontend parts connected by the Flask framework. In the frontend part, the user interacts with the system through a website. The user can upload a legal document to the system. Once the document is sent to the backend part, the system will undergo preprocessing, such as chunking and tokenizing. Then, the pre-processed document will be passed to the trained BART model to generate a summary. Additionally, the attention weight of the tokens

in the document is visualized to produce a highlighted original document, showing important sections or terms. The higher the weight, the deeper the highlight colour, the more important the word token is. Once the outputs are generated, both will be sent back to the frontend via Flask, and the user can see the result on the website. The user can also download the outputs to keep the results.

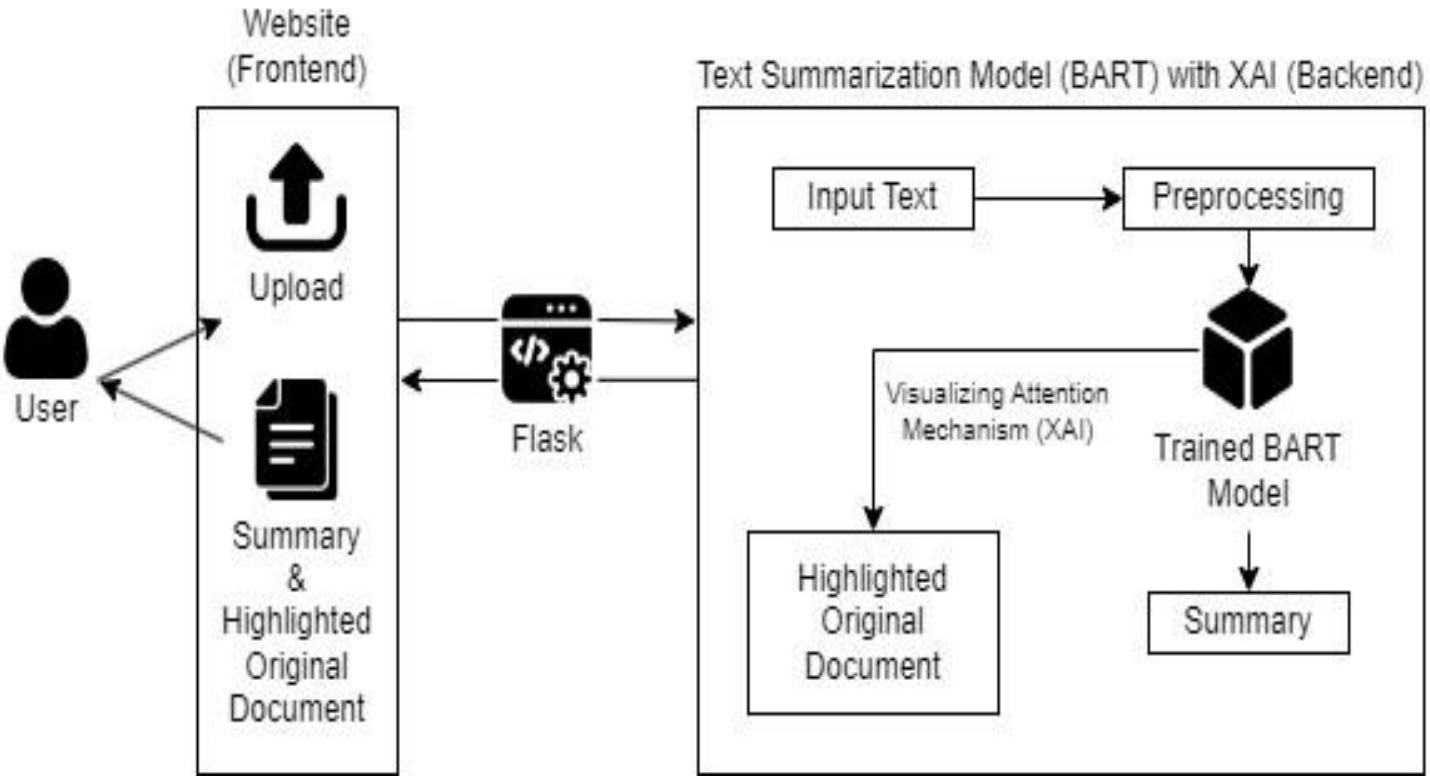

Figure 1. The architecture of the proposed system

Dataset

The dataset used is Indian Supreme Court case documents and their abstractive summaries. The system's text summarisation model requires two datasets for training and testing purposes. There are a total of 7130 documents in the original dataset. One hundred (100) documents from the dataset are randomly chosen as testing datasets; the remaining will be the training dataset. All the data is in .txt format. The dataset was downloaded from Zenodo, a research data repository [27].

Training Process

Figure 2 shows the training process of the proposed text summarization model. The necessary libraries, such as NumPy, NLTK, Pandas, TQDM and BART_utilities, are imported. BART_utilities is the file containing libraries that utilize PyTorch Lightning. The paths used in the following process are defined, including the dataset, root path, and output path. The training data, including judgment and summary, are read from specified directories and stored in lists. A total of 7030 data (a pair of judgments and summary considered as 1 data) is used in the training process.

A pre-trained sentence transformer is initialized and loaded, specifying that it should run on CUDA (Compute Unified Device Architecture). This is because CUDA significantly boosts training and running a model. Next, the model will check both the judgement and summary lists. If there is a document in the lists, the model will split paragraphs into individual sentences and store them in lists (for example, l1 for judgment sentences and l2 for summary sentences). The cosine similarity between two lists of sentences is calculated. Based on the result, chunks of text from judgment and their corresponding summaries will be generated. Once no more documents exist in the judgment and summary lists, the training chunks and summaries are generated and stored in an Excel file. The file is then read, and the columns are renamed into source and target data.

The environment for using the BART model is set up to continue the training process, including loading a pretrained BART model and tokenizer. Special tokens are added to the tokenizer, and the token embeddings in the BART model are resized. The data module and lightning model with specified parameters (BART model is used) are initialized. A PyTorch Lightning trainer is set up with GPU (Graphics Processing Unit) acceleration and other training parameters. After all settings are done, the training starts until it reaches the maximum number of epochs. The training process ends with saving the model weight into the checkpoint.

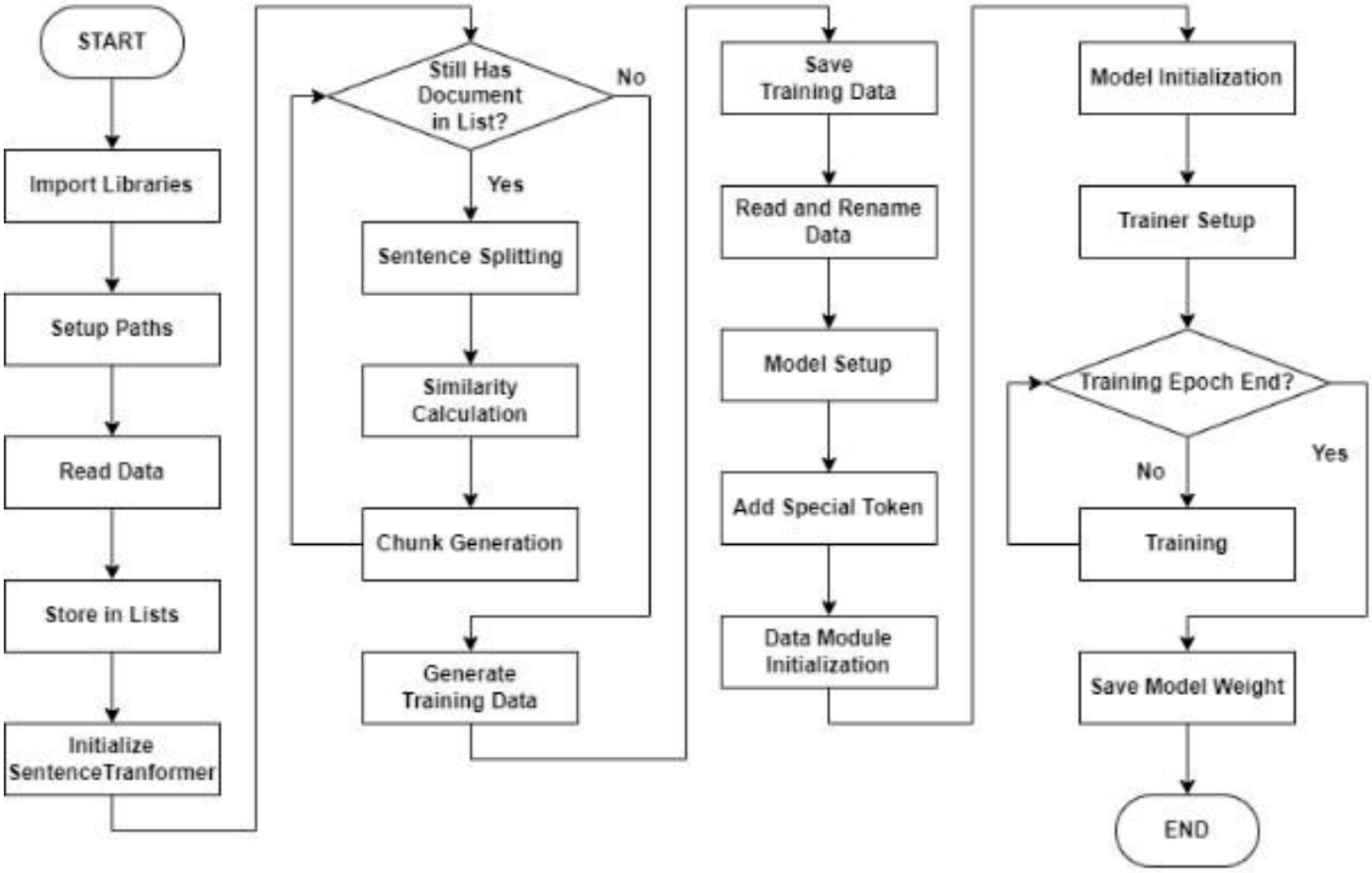

Figure 2. The flowchart of the model training process

Testing Process

Figure 3 illustrates the testing process of the proposed text summarization model. Similar to the training process, it starts with importing libraries, setting up paths, and reading documents. The environment for using the BART model is set up by loading the pre-trained BART model and tokenizer. Special tokens are added to the tokenizer, and the token embeddings in the BART model are resized. The lightning model with specified parameters is initialized, and the saved model weight is loaded (trained BART model).

The required summary length is set to 15% of the length of the original document. This is because the summary length should usually be 10% to 15% of the original text length or even shorter than the range [28]. The model next retrieves the document's name and content. The word count of the document and the required summary length are calculated. The model splits the document into nested chunks of sentences with a maximum chunk length of 1024 words. The required summary length per chunk and percentage of document length for summary is calculated.

The testing process continues by generating summaries for each chunk using the BART model on GPU. The generated summaries are concatenated into a single string. If the length is more than the required length, truncation needs to be done; otherwise, the final summary will be written in the specified output file, which is the end of the testing process.

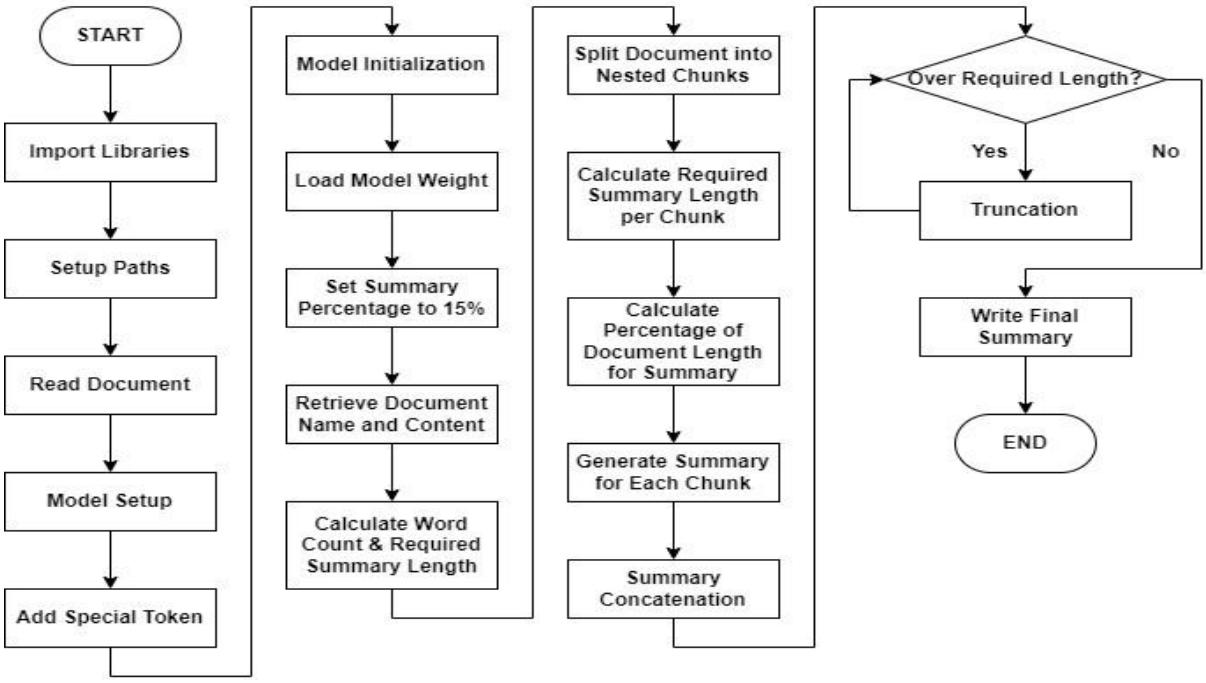

Figure 3. The flowchart of the model testing process

2.4. Explainability Feature

Figure 4 displays the process of how the explainability feature is done. The process is similar to the testing process until the read document step. These steps can be skipped if continued after the summary generation. The attention weights are extracted from the model's output, where the last layer's attention is targeted.

The original document is split into individual tokens to visualize the attention weights. The important tokens are highlighted based on the attention weights by setting highlight colours proportional to the attention weights. The higher the attention weights, the more important the token and the more obvious the highlight colour is. A PDF (Portable Document Format) file containing the original document text with highlighted tokens based on attention weights is created. The generated PDF file is saved to the specified output directory.

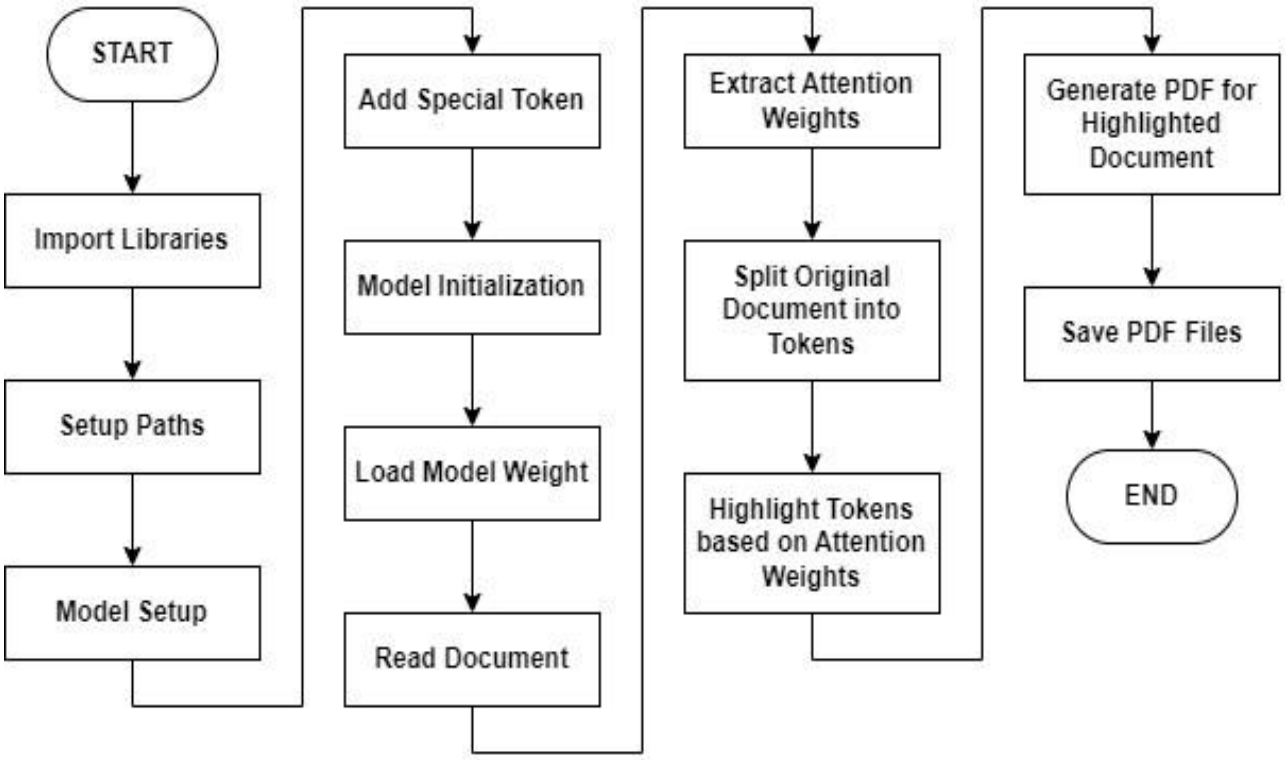

Figure 4. The flowchart of the explainability feature process

RESULTS

The methods used to evaluate the model performance in this research, which is the BART text summarization model, are BERTScore (including precision, recall, and F1- score), cosine similarity, ROUGE score, and user feedback. The first three evaluations are done by comparing the generated summaries and the reference summaries, while the last method is the feedback received from the target users.

BertScore

In the context of BERTScore, precision reflects how many of the tokens generated by the model are similar to the tokens in the reference summary, recall reflects how well the generated summary covers the tokens from the reference summary, and F1-score combines both to give an overall sense of how similar the generated text is to the reference text in terms of token similarity. High precision but low recall indicates that the model generates very accurate tokens, but it might miss out on important content. High recall but low precision indicates that the model includes the most relevant content and adds unnecessary or irrelevant tokens. A high F1 score suggests a good balance between capturing the important content and avoiding irrelevant content in the generated summaries. For this method, the BERTScore of every single testing data and the average are calculated. The overall precision, recall, and F1-score are 0.6241, 0.5976, and 0.6091, respectively. From the score calculated, the model can be proved to generate entirely accurate tokens without missing some important content. However, the performance can still be improved to get a higher score.

Cosine similarity

Cosine similarity is a metric used to measure how similar the documents are, irrespective of their size [29]. This metric is advantageous as the two similar documents could still have smaller angles even if they are far apart by the Euclidean distance because of the size. The smaller the angle, the higher the similarity. The cosine similarity value is allocated between 0 and 1; the larger the value, the more similar the two documents are. The cosine similarity of the reference summaries and summaries generated by the proposed system and the overall accuracy based on cosine similarity is calculated. A threshold of 0.8 is defined to calculate the accuracy, meaning only summaries with a cosine similarity of more than 0.8 will be considered. From the given result, the accuracy is 0.94, showing 94 documents from 100 testing data with cosine similarity higher than 0.8.

Rouge score

ROUGE-1, ROUGE-2, and ROUGE-L are used to evaluate the proposed application. ROUGE-N measures the overlap of n-grams between the reference summaries and system-generated summaries, where ROUGE-1 refers to the overlap of unigrams (every single word), and ROUGE-2 refers to the overlap of bigrams (two consecutive words). At the same time, ROUGE-L is based on the length of the longest common subsequence. The average ROUGE score has been calculated, which ROUGE-1 is 0.4911, ROUGE-2 is 0.2434, and ROUGE-L is 0.2449. The scores are considered moderate except ROUGE L, which is lower than 0.3 [30], [31]. However, it cannot be concluded that the overall performance of the application is poor because ROUGE score relies on reference summaries. The ROUGE score primarily focuses on the proportion of relevant information preserved in a summary, which may not always be the most crucial aspect in evaluating the system. Researchers may sometimes prioritize how accurately key details are captured, or fluency, which assesses the coherence and naturalness of the generated summary.

Users' feedback

A survey has been prepared for the users to rate the system's functionality. The target users are those who study or work in legal fields. Ten (10) target users are chosen and will continue with the questions about the system's functionality. A simple demographic is prepared in the survey to give an essential impression of the target users, especially in the years they are involved in the legal field, the frequency they deal with legal documents and their experience using text summarization tools. This can help answer the functionality questions better because different experiences can have different opinions on the system. Figure 5 is the bar chart showing how accurately the users find the summaries generated by the system. The rating started from 1 (very inaccurate) to 5 (very

accurate). Most of the target users (6 respondents, 60%) find the summaries generated to be accurate, and one of them (10%) said they are very accurate. In comparison, three respondents (30%) found the summaries generated to be only moderate.

Figure 6 displays the bar chart of target users' opinions on the importance of the highlighted words, rating from 1 (very unimportant) to 5 (very important). Half of the target users (5 respondents, 50%) think the highlighted words are important, followed by three respondents (30%) who feel the importance of highlighted words is average. One respondent (10%) voted for each for the range of unimportant and very important. Figure 7 shows the bar chart about the target users' understanding of the legal documents by only reading the generated summaries. The rating is between 1 (very poorly) to 5 (very well). Six respondents (60%) from the target users can understand well, and one respondent (10%) can understand very well. 30% (3 respondents) of the target users only moderately understand the legal documents if they only read the generated summaries.

Open-ended questions are also prepared for the respondents for extra comments or feedback. Most respondents think the system is already good at present, but some also provide their opinions on future improvement. The most significant progress to be made in enhancing the user interface and content layout is increasing the line spacing, being more creative in colours and design, and ensuring the layout is more organized for easy viewing. Users also request extra functions, such as language translator features and an adjustable summary for users to select the summary length. Overall, the feedback highlights the users' satisfaction with the current system but also points out some valuable suggestions for enhancing the user interface (UI), layout, and functionality to improve the user experience (UX) further.

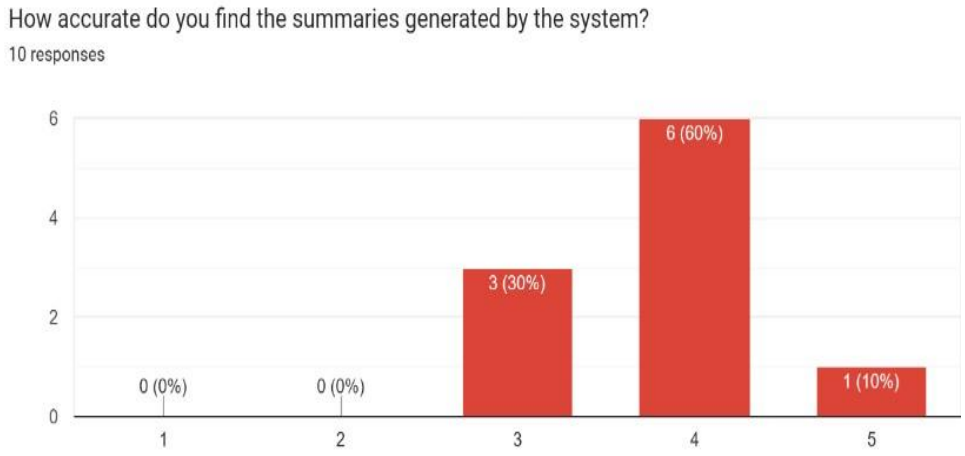

Figure 5. The bar chart of how accurately the users find the summaries generated by the system

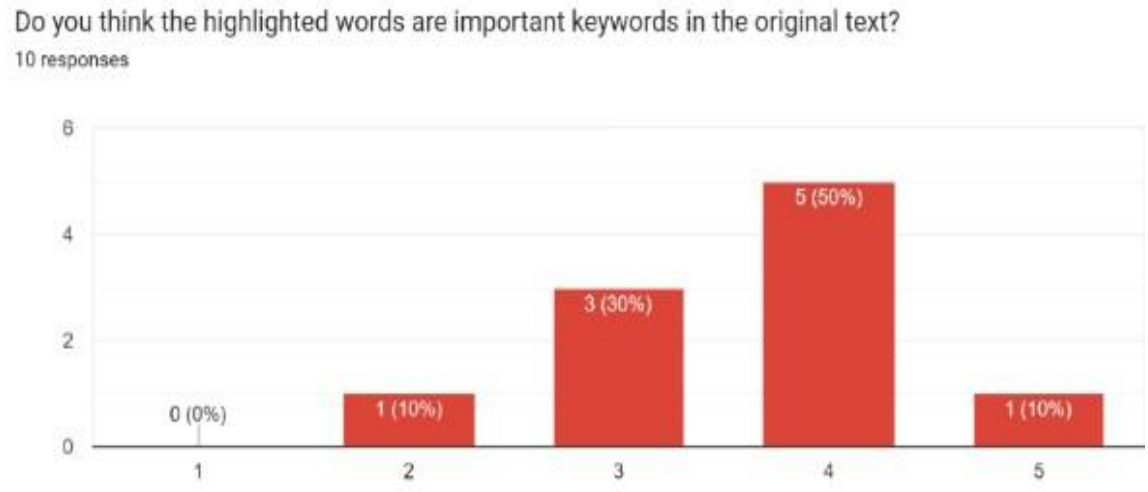

Figure 6. The bar chart of the importance of highlighted words

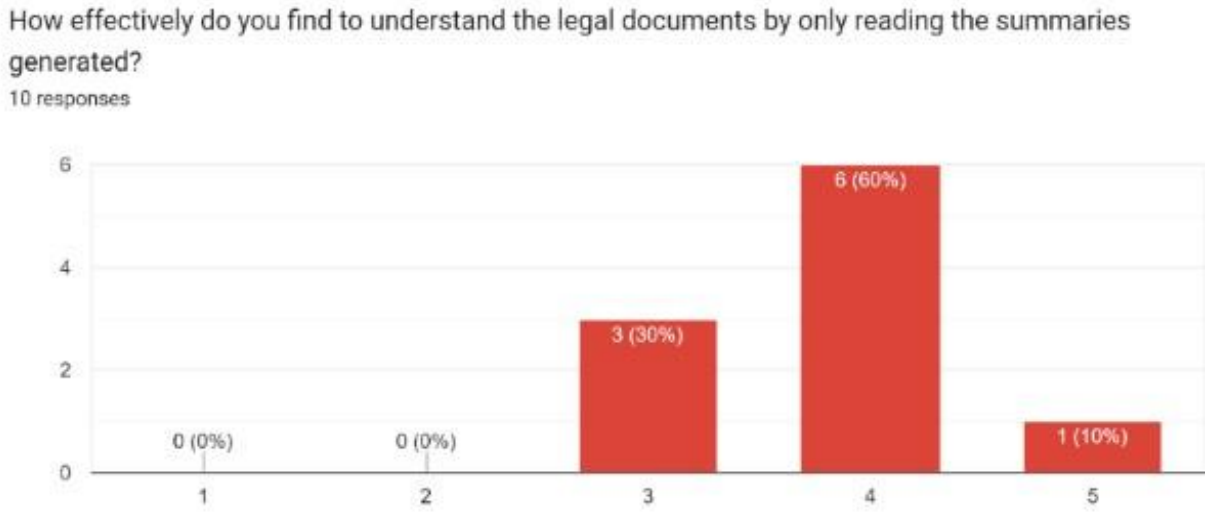

Figure 7. The bar chart of how accurately the users find the summaries generated by the system

CONCLUSION

According to the results, the research successfully meets the set objectives by delivering a functional and explainable AI-driven text summarization system. The system demonstrates significant potential, especially with the integration of XAI. It also reveals some areas for improvement, such as performance optimization and user interface enhancement. The proposed improvements, including adjusting training parameters, upgrading the user interface, and adding new features like adjustable summary length and a language translator, will further enhance the system's functionality and user satisfaction. To conclude, the research contributes valuable insights and tools to the AI and legal field and creates a basis for further development. With the recommended improvements, the system has a high potential to become a leading tool in its domain, offering users a more powerful, customizable, and accessible text summarization experience.

ACKNOWLEDGEMENTS

The authors would like to thank the Applied Intelligent Computing (APIC) research group, the Center of Advanced Computing Technology (C-ACT), and Fakulti Kecerdasan Buatan dan Keselamatan Siber, Universiti Teknikal Malaysia Melaka (UTeM) for their incredible support in this research.

REFERENCES

1. V. Pandya, "Automatic Text Summarization of Legal Cases: A Hybrid Approach," in 5th International Conference on Advances in Computer Science and Information Technology, 2019, doi: 10.5121/csit.2019.91004.
2. D. Anand, and R. Wagh, "Effective deep learning approaches for summarization of legal texts," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 5, pp. 2141–2150, May 2011, doi: 10.1016/j.jksuci.2019.11.015.
3. A. Shukla, P. Bhattacharya, S. Poddar, R. Mukherjee, K. Ghosh, P. Goyal, and S. Ghosh, "Legal Case Document Summarization: Extractive and Abstractive Methods and their Evaluation." in Proceedings of the conference of the Asia-Pacific chapter of the Association for Computational Linguistics and the international joint conference on natural language processing, Oct 2022, Available: https://arxiv.org/abs/2210.07544.
4. Adarsh, "Text Summarization with Attention based Networks" https://medium.com/@iit2018056/text-summarizationwith-attention-based-networks-8492e9277c0 (accessed Apr, 29, 2024).
5. Z. Wang, "An Automatic Abstractive Text Summarization Model based on Hybrid Attention Mechanism," Journal of Physics: Conference Series, vol. 1848, no. 1, 012057, April 2021, doi: 10.1088/1742-6596/1848/1/012057.

6. N, Alipour and S. Aydin, “Abstractive summarization using multilingual text-to-text transfer transformer for the Turkish text”, International Journal of Artificial Intelligence, vol. 14, no. 2, April 2025, pp. 1587-1596.
7. J. K. Adeniyi, S. A. Ajagbe, A. E. Adeniyi, H. O. Aworinde, P. B. Falola, and M. O. Adigun, “EASESUM: an online abstractive and extractive text summarizer using deep learning technique”, International Journal of Artificial Intelligence, vol. 13, no. 2, June 2024, pp. 1888-1899.
8. S. Dutta, A. K. Das, S. Ghosh, and D. Samanta, "Graph-based clustering technique for microblog clustering," in Data Analytics for Social Microblogging Platforms. Academic Press, 2023, pp. 165-192. Accessed: Apr, 29, 2024. [Online]. Available: https://books.google.com.my/.
9. N. A. Ranggianto, D. Purwitasari, C. Fatichah, R. W. Sholikah, Abstractive and Extractive Approaches for Summarizing Multi-document Travel Reviews. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi). 2023 Dec 30;7(6), pp. 1464-75.
10. A. M. Zakariae, B. Frikh, and B. Ouhbi. "EXABSUM: a new text summarization approach for generating extractive and abstractive summaries." Journal of Big Data 10, no. 1, 2023, pp. 163.
11. N. Giarelis, C. Mastrokostas, and N. Karacapilidis, "Abstractive vs. Extractive Summarization: An Experimental Review," Applied Sciences, vol. 13, no. 13, pp. 7620, 2023, doi: 10.3390/app13137620.
12. M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871-7880 Available: https://arxiv.org/abs/1910.13461.
13. G. Erkan, and D. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," Journal of Artificial Intelligence Research, vol. 22, pp. 457-479, 2004, doi: 10.1613/jair.1523.
14. D. R. Radev, and Z. Zhang, (2001, October 5). "Experiments in Single and MultiDocument Summarization Using MEAD. ResearchGate," in First document understanding conference, 2001, pp. 1-7.
15. K. Sharma, K. Singh, K. Sharma, and J. Gupta, "Question Summation and Sentence Similarity using BERT for Key Information Extraction." International Journal for Research in Applied Science and Engineering Technology, vol. 11, no. 4, pp. 1636–1639, 2023 doi: 10.22214/ijraset.2023.50087.
16. P. Watanangura, S. Vanichrudee, O. Minteer, T. Sringamdee, N. Thanngam, and T. Siriborvornratanakul. "A comparative survey of text summarization techniques." SN Computer Science, vol. 5, no. 1, 2023, pp 47.
17. R.D. Lins, H. Oliveira, and S.J. Simske, “Assessing the Reliability and Validity of the Measures for Automatic Text Summarization.” In Proceedings of the ACM Symposium on Document Engineering 2024, pp. 1-4.
18. B. Zhao, and Y.M. Lui, “Towards A Reliable Text Summarization Evaluation Metric Using Predictive Models”. International Journal of Pattern Recognition and Artificial Intelligence, vol. 36, no. 10, pp. 2251011, 2022.
19. A.M. Ibrahim, A. Marco and M. Aref. "A Systematic Review On Text Summarization Of Medical Research Articles." International Journal of Intelligent Computing and Information Sciences vol. 23, no. 2, pp. 50-61, 2023.
20. H. Özbolat, "Text Summarization: How to Calculate BertScore," https://haticeozbolat17.medium.com/textsummarization-how-to-calculate-bertscore-771a51022964 (accessed Apr, 29, 2024).
21. A. Mulkar, "Explainable AI (xAI) in Natural Language Processing (NLP)," https://ankushmulkar.medium.com/explainable-ai-xai-innatural-language-processing-nlp-d75d5be216e3 (accessed Apr, 29, 2024).
22. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in Advances in Neural Information Processing Systems, vol. 30, 2017 Available: https://arxiv.org/abs/1706.03762
23. N. H. A. M. Norkute, "Explainable AI for Text Summarization of Legal Documents," https://hyperight.com/explainable-ai-for-textsummarization-of-legal-documents/ (accessed Apr, 29, 2024).

24. "What are the pros and cons of using attention mechanisms in text summarization with RNNs?" www.linkedin.com. https://www.linkedin.com/advice/3/what-pros-cons-using-attentionmechanisms-textsummarization (accessed Apr, 28, 2024).
25. J. Vig, "Visualizing Attention in Transformer-Based Language Representation Models," 2019, Available: https://arxiv.org/abs/1904.02679.
26. D. Garreau, and U. Luxburg, "Explaining the Explainer: A First Theoretical Analysis of LIME," in Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 26–28 August 2020, vol 108, pp. 1287–1296 https://proceedings.mlr.press/v108/garreau20a.html
27. A. Shukla, P. Bhattacharya, S. Poddar, R. Mukherjee, K. Ghosh, P. Goyal, and S. Ghosh, "Legal Case Document Summarization: Extractive and Abstractive Methods and their Evaluation," in Proceedings of the conference of the Asia-Pacific chapter of the Association for Computational Linguistics and the international joint conference on natural language processing, 2022. doi: 10.5281/zenodo.7152317
28. C. Burnell, J. Wood, M. Babin, S. Pesznecker, and N. Rosevear, "Writing Summaries," Pressbooks. https://openoregon.pressbooks.pub/wrd/chapter/writing-summaries/ (accessed May, 12, 2024).
29. M. H. Asif, and A. U. Yaseen, "Comparative Evaluation of Text Similarity Matrices for Enhanced Abstractive Summarization on CNN/Dailymail Corpus." Journal of Computing & Biomedical Informatics, vol. 6, no. 01, 2023, pp. 208-215.
30. G. Sharma, and D. Sharma, "Automatic text summarization methods: A comprehensive review." SN Computer Science, vol. 4, no. 1, 33, 2022.
31. T. Chellatamilan, S. K. Narayanasamy, L. G. K. Srinivasan, and S. MN. Islam. "Ensemble Text Summarization Model for COVID-19-Associated Datasets." International Journal of Intelligent Systems 2023, no. 1, 2023, pp. 3106631.