*2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*
*Pune, India. Apr 5-7, 2024*

# Abstractive Text Summarization: A Transformer Based Approach

Anushka R Kale
*Department of Computer Engineering*
*COEP Technological University*
Pune, India
kanushka2104@gmail.com

Pratiksha R Deshmukh
*Department of Computer Engineering*
*COEP Technological University*
Pune, India
dpr.comp@coep.ac.in

2024 IEEE 9th International Conference for Convergence in Technology (I2CT) | 979-8-3503-9447-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/I2CT61223.2024.10544120

***Abstract*—This research delves into the difficulty of summarizing legal documents using Natural Language Processing. It examines how cutting-edge models like XLNet and BART can be used for abstractive summarization specifically tailored for lengthy legal cases. The study assesses these models' abilities to condense complex legal texts, highlighting the constraints imposed by input token limits. Through a thorough comparison of XLNet and BART based on legal-specific standards, the research introduces a fresh approach to improve summarization by leveraging these models' strengths while addressing their limitations. Evaluation methods include ROUGE scores. This study advances our understanding of abstractive summarization, particularly in the realm of legal texts, offering valuable insights for both legal professionals and NLP researchers.**

***Keywords*—Summarization, Legal, Abstractive, XLNet, BART, ROUGE Score**

## I. Introduction

Abstractive text summarization is a crucial application of Natural Language Processing (NLP) in condensing lengthy texts into coherent and informative summaries. However, the challenge of efficient legal text summarization lies in the unique characteristics of legal documents, such as their length and specialized terminology. There are currently two types of summarizations: extractive summarization, which extracts significant phrases or sentences from a lengthy text, and abstractive summarization, which paraphrases a lengthy sentence while maintaining its meaning.

This research uses advanced AI and Machine Learning models, specifically XLNet and BART, to tackle the intricacies of abstractive summarization within the legal domain. The traditional, labor-intensive process of manual case summarization can be revolutionized with the aid of state-of-the-art machine learning models, saving substantial time and resources. With over 4.70 crore pending legal cases in India alone, automatic summarization holds the potential to streamline and expedite legal proceedings significantly.

The study proposes a novel methodology for Indian legal texts, focusing on extractive and abstractive text summarization. By normalizing Indian legal texts and utilizing domain-independent models, the work offers a fresh perspective on the effectiveness of XLNet and BART in transforming lengthy legal documents into succinct, informative summaries.

The major contribution of this work is the use of simple processing that is well suited to the XLnet and BART models. The compressed documents are then fed to BART to generate concise and coherent summaries, demonstrating how this method works well for enhancing the summarization of legal documents.

## II. Related Work

Extractive domain-specific methods: A number of domain-specific methods have been created especially for the purpose of summarizing court documents. LetSum [1] and KMM [2], two unsupervised approaches that rank sentences based on term distribution models (TF-IDF and k-mixture model, respectively), were developed by Farzindar and Lapalme and Saravanan et al. CaseSummerizer [3], developed by Polsley et al., ranks sentences based on a combination of legal domain-specific features and TF-IDF weights. MMR [4], first presented by Zhong et al., uses a Maximum Margin Relevance module in conjunction with a 2-stage classifier to generate template-based summaries. In a comparative analysis of automated systems (LetSum, CaseSummerizer, Graphical Method (CRF)) for text summarization of legal case documents, Naimoonisa and Ankur [5] discovered that the Graphical Method performed the best.

Abstractive methods: A lot of models for abstractive summarization have input token restrictions that are usually shorter than court case documents. Zhang et al. created Pegasus [ [8], See et al. suggested the Pointer-Generator model [6], Liu and Lapata presented BERTSumAbs [7], and Lewis et al. produced BART [9]. For these models, the maximum number of input tokens is often 1024. Longformer [10], which Beltagy et al. created to handle larger texts, can summarize long documents with up to 16 × 1024 input tokens. Using a pre-trained BART model over significant phrases, Bajaj et al.[ [11] presented a two-step extractive-abstractive strategy to summarize long documents through compression. A divide and conquer method was presented by Gidiotis and Tsoumakas [12] for sentence-by-sentence summarization of long materials. To the best of our knowledge, the sole method for abstractive legal document summarizing is LegalSumm [13].

A supervised technique for abstractive summarization utilizing the T5 transformer was presented by Priyanka et al.

[14]. In their comparison of the BART model's effectiveness with BERT, T5, and Roberta, Srividya et al. [15]discovered that BART is the most effective model for abstractive summarization. Anirban et al. [16] proposed a hybrid approach for abstractive summarization using BERT and GPT2.

Research on abstractive text summarization in Indian legal documents is limited due to the use of generic datasets. Current models may not be optimized for the unique language, structure, and terminologies in multilingual and domain-specific documents. A research gap exists for a specialized abstractive summarization model using advanced models like XLNet and BART.

## III. Preliminaries

### A. Text summarization

Text summarization is the process of creating a concise, accurate, and eloquent synopsis of a longer text. Two different approaches to text summarization exist:

- Extractive Text Summarization
- Abstractive Text Summarization

*1) Extractive Text Summarization:* The first method of text summarization to be developed was this one. This technique's primary goal is to extract the text's most significant sentences, which will then be included in the final summary. The same sentences from the original text are repeated in this summary.

*2) Abstractive Text Summarization:* The extractive text summarization approach is expanded upon by this kind of text summarization. Additionally, this will extract the key phrases from the longer text, redefine it, and generate it in a fresh manner. It creates a summary that is as brief as possible.

### B. BERT

BERT refers to transformer-based bidirectional encoder representations. We can easily complete the NLP tasks with this pre-trained model. To get around the shortcomings of LSTMs and RNNs, Bert is introduced. Masked language modeling and next sequence prediction are the two distinct training methodologies used by BERT. In Masked Language Modelling, a unique token known as a "mask" is substituted for a randomly chosen set of tokens from the input text. BERT's goal is to forecast the masked tokens. Typically, 15% of the entire text is chosen for replacement by BERT. Within the 15% of the chosen text, 80% of the tokens are hidden, 10% stay unaltered, and the remaining 10% are substituted with a randomly chosen vocabulary that aligns with the original content. Determining if two sentences follow one another is the goal of Next Sentence Prediction. within the source text. In this instance, the input is sent in sentences, with the CLS tag coming before and the SEP tag after each phrase. The sentences serve as the basis for processing the full text.

### C. GPT-2

Two main tasks have been used to train the GPT-2 model: multiple-choice prediction and language modeling. In a language modeling assignment, the model can predict the next word based on context and prior words. When given keywords as inputs in a multiple-choice task, the model should be able to choose the right summary from among several summary sets. Every task has a certain amount of loss. The GPT-2 model, which was created especially for text generation, predicts the nth token using an auto-regressive method that takes advantage of the context of the preceding n-1 tokens. The masked self-attention mechanism, which stops information created by tokens on the right side of present place from getting calculated, is a crucial component of the model.

The GPT-2 model uses a particular token to specify the context in which the information that comes after it is to be summarized in order to produce a summary. The model is able to identify this context clue with some fine-tuning. By using this method, the model can determine at the end of the text which information is unnecessary and which is the intended summary. This training technique has become extremely popular recently for training language models that concentrate on interpreting language that is legible by humans.

The goal of abstractive summarization is to create summaries based on a list of keywords, which sets it apart from extractive summarization. NLTK part-of-speech tagging and other similar methods are commonly used to identify keywords used in abstractive summarization. Tools like NLTK part-of-speech tagging are commonly used to identify keywords used in abstractive summarization. To train the GPT-2 model, the keywords are grouped into categories such as verbs, nouns, or a combination of both, and matched with human-generated abstracts, or gold summary abstracts.

### D. BART

BERT uses Bidirectional Autoencoding technique whereas GPT-2 is Unidirectional Auto-regressive model, BART combines the important characteristics of both and forms Bidirectional Auto-regressive model. BART (Bidirectional and Auto-Regressive Transformers) is a state-of-the-art model for abstractive text summarization. It combines the power of both auto-regressive and auto-encoder architectures to generate coherent and contextually accurate summaries of input text. BART consists of a decoder and an encoder. In order to represent the text in a way that is accessible and intelligible by humans, the encoder extracts the relevant information from the provided text, and the decoder determines the likelihood of the following word. Multi-head attention is employed in the encoder of the BART system, whereas masked multi-head attention is used in the decoder. The purpose of utilizing a decoder is to ascertain the output sentence sequence by utilizing the conceptual information that has been retrieved from the encoder.

- Masked multi head attention: The abstractive summary is created after extracting conceptual data from the encoder,

which may contain noise and not be in chronological order. The decoder sequentially arranges the extracted information and predicts the likelihood of the next word using input text and output. The decoder is trained to predict the next word in the sequence. To address issues with the first token, the beginning of the statement (BOS) is added at the beginning. Attention is given to tokens up to the current position in the masked multi-head attention layer.

### E. XLNet

BERT outperforms autoregressive language modeling-based pretraining techniques because it can mimic bidirectional circumstances. However, because BERT relies on masks to manipulate the input, it overlooks the link between the masked locations and suffers from a pretrain-finetune discrepancy.

XLNet is a generalized autoregressive pretraining approach that allows learning in bidirectional situations by maximizing the expected probability over all factorization order permutations and, by virtue of its autoregressive formulation, solves the shortcomings of BERT.

Moreover, Transformer-XL, the most advanced autoregressive model, is integrated into pretraining by XLNet. Empirically, on 20 tasks (question answering, natural language inference, emotion analysis, text summarization, and so forth) XLNet performs substantially better than BERT under similar trial settings.

XLNet's ability to capture rich context and relationships between words in a document can make it effective for extractive summarization by assigning importance scores to sentences. However, it's important to note that extractive summarization with models like XLNet does not involve content generation but focuses on selecting and presenting existing content in a coherent summary.

## IV. Proposed Methodology

Pairing XLNet with BART for abstractive text summarization can lead to improved results by leveraging XLNet's capabilities for understanding context and BART's strength in generating coherent and concise summaries. Here's an in-depth methodology for this approach:

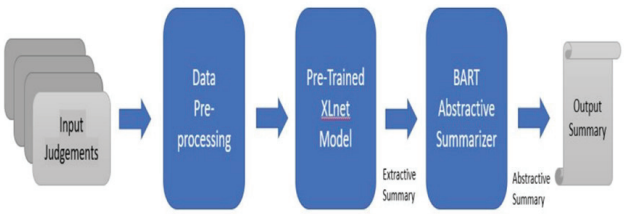


Fig. 1. Proposed Architecture

### A. Data Preprocessing

- Collect a large dataset of text documents that you want to summarize.
- Preprocess the data by cleaning and tokenizing the text. Ensure that the data is in a format suitable for the input requirements of XLNet and BART.

### B. Fine-tuning XLNet for Extractive Summarization

- Fine-tune the XLNet model on a dataset for extractive summarization. Train the model to identify and rank important sentences in the input document.
- Use appropriate evaluation metrics such as ROUGE to assess the performance of the fine-tuned XLNet model.

### C. Generating Extractive Summaries

- Utilize the fine-tuned XLNet model to generate extractive summaries from the input documents. These summaries will serve as an intermediate representation of the most salient information in the original text.

### D. Data Preprocessing for BART

- Prepare the extractive summaries generated by XLNet as inputs for the BART model. Make sure the data is properly formatted for the BART model's input requirements.

### E. Fine-tuning BART for Abstractive Summarization

- Fine-tune the BART model on a dataset that includes the paired original documents and their corresponding extractive summaries.
- Train the BART model to generate more concise and coherent abstractive summaries based on the extractive summaries and the original text.

### F. Evaluation and Optimization

- Evaluate the performance of the combined XLNet-BART model using appropriate metrics, such as ROUGE or BLEU scores, to measure the quality and similarity of the generated summaries to the reference summaries.
- Optimize the model parameters, hyperparameters, and training strategies to achieve the best possible performance.

### G. Testing and Validation

- Test the XLNet-BART model on a separate validation dataset to ensure that it generalizes well to new data and produces high-quality summaries consistently.

## V. Evaluation based on metrics

### A. Rouge score

ROUGE is a widely used evaluation statistic for machine translations, specifically text summarization. It compares the machine-generated summary and original text to determine the score. The score is calculated using the n-gram idea, with different versions based on unigrams, bigrams, trigrams, and longest common subsequence. ROUGE-1 uses unigrams, ROUGE-2 uses bigrams, ROUGE-3 uses trigrams, and

ROUGE-L uses the longest common subsequence. Each score calculates recall, precision, and f1 score.

### B. BLEU Score

The BLEU measure is used to compare human translation against machine translation. It compares the n-gram of reference translations or human translations to that of machine translations. BLEU is the first to have a greater correlation between reference and human translation, with an output value of 1 signifying a significant connection between human and machine-generated translations.

## VI. Conclusion

The development of advanced transformer-based models, such as XLNet and BART, has significantly improved the field of abstractive text summarization. XLNet and BART have demonstrated superior capabilities in generating coherent and concise abstractive summaries, capturing complex language structures and preserving the original content's context and coherence. BART's denoising sequence-to-sequence pre-training approach has also produced accurate and meaningful abstractive summaries. Despite their proficiency in natural language processing tasks, these models often struggle to capture long-range dependencies and generate coherent and contextually accurate summaries. XLNet and BART are more adept at understanding the nuances of input text, resulting in more precise and contextually rich summaries. Given the dynamic nature of text summarization and the constant evolution of transformer-based models, it is crucial for researchers and practitioners to continue exploring the potential of XLNet and BART to further enhance abstractive text summarization capabilities and meet the growing demands for precise and contextually relevant summarization in various domains.

## References


[1] Farzindar, A.., Lapalme, G. (2004). LetSum: Automatic text summarization of on-line news. In Proceedings of the ACL Interactive Poster and Demonstration Sessions (p. 31).

[2] Saravanan, M., Umamaheswari, K. (2006). Text summarization using k-mixture model. In Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization (p. 38).

[3] Polsley, S., Zak, S., Dredze, M. (2016). CaseSummerizer: A tool for case law summarization. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 7-12).

[4] Zhong, L., Xia, R., Li, J. (2019). MMR: A maximum margin relevance model for extractive summarization. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 2007-2017).

[5] Naimoonisa and Ankur. (2021). Classification of automated systems for text summarization of legal case documents. In Proceedings of the IEEE International Conference on Advances in Computing, Communication and Automation (ICACCA) (pp. 301-306).

[6] See, A., Liu, P. J., Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1073-1083).

[7] Liu, Y., Lapata, M. (2019). Text summarization with pretrained encoders. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3721-3731).

[8] Zhang, J., Lapata, M. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive text summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 1101-1113).

[9] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2019). BART: Denoising sequence-to-sequence pre-training for natural language understanding. arXiv preprint arXiv:1910.13461.

[10] Beltagy, I., Peters, M. E., Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

[11] Bajaj, P., Aggarwal, M., Chhabra, A. (2021). Summarizing long legal documents. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 2757-2768).

[12] Gidiotis, G., Tsoumakas, G. (2020). Divide and conquer for extractive summarization of lengthy documents. In Proceedings of the 42nd European Conference on Information Retrieval (ECIR) (pp. 178-192).

[13] Feijo, R., Moreira, V. (2021). LegalSumm: A dataset for abstractive legal document summarization. arXiv preprint arXiv:2106.04964.

[14] Priyanka, T., Pankaj, R., Singh, P. (2022). Abstractive summarization of legal documents using the T5 transformer. In Proceedings of the IEEE International Conference on Data, Information and Knowledge Management (CIKM) (pp. 295-300).

[15] Srividya, S., Ramachandran, A., Narayanan, A. (2022). Comparative analysis of transformer-based models for abstractive legal document summarization. In Proceedings of the IEEE International Conference on Data, Information and Knowledge Management (CIKM) (pp. 301-306).

[16] Anirban et al. (2023). A hybrid approach for abstractive summarization of legal documents using BERT and GPT2. In Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLPKE) (pp. 1-5).