**Enhancing Legal Document Accessibility with LegalEase: An AI-Powered Summarization, Translation, and Text-to-Speech Solution**

Mylie Mudaliyar1, Muskan Sah2,Diksha Adhikari3 and Akila Vector4

1 Vellore Institute of Technology, Vellore, Tamil Nadu, India

**Abstract.** Legal documents are often difficult for people without expertise in the field because they use complex language. People's ability to access legal information is hampered by this barrier, which could lead to misunderstandings or misinterpretations that could eventually impair their ability to assert their legal rights and make informed decisions. In this paper, a model called LegalEase, an AI-powered tool that helps laypeople read legal documents is presented. It seeks to improve accessibility and optimize legal documents. LegalEase offers tools for text-to-speech, translation, and summarization by utilizing cutting-edge natural language processing techniques. The summarization employs the Pegasus model, fine-tuned on legal cases. The paper outlines LegalEase's methodology and emphasizes how it improves the accessibility of legal documents. The study addresses the Pegasus model's advantages and disadvantages in the legal field and assesses it using the ROUGE measure. LegalEase demystifies legal papers through audio output, translation, and summarizing, enabling people to navigate the legal system with greater clarity.

**Keywords:** Artificial Intelligence, Natural Language Processing, Legal Document Processing, Abrastaction, Summarisation, Translation, Speech Conversion

# 1 Introduction

It can be difficult for non-experts to fully understand legal papers due to the complicated language and terminology used in the legal sphere. It is ultimately difficult for people to exercise their rights and make educated decisions because of this barrier, which also makes it difficult to obtain legal information and may cause misconceptions or misinterpretations. Using artificial intelligence (AI) and natural language processing (NLP) tools, LegalEase seeks to close this gap by making legal papers easier to comprehend and accessible to a wider audience.

The abstractive text summarization component uses the Pegasus model, a state-of-the-art model that was refined on a dataset of Indian court cases. We investigated the applicability of several pre-trained summarizing models, such as LED (Lightweight Extractive Document summarization) and BART (Bidirectional and Autoregressive Transformer for Pre-training). We finally selected the Pegasus transformer model since it was the most successful in producing factually correct, succinct, and grammatically good summaries (state your precise criterion here), even if both BART and LED performed moderately. A dataset of Indian court cases was used to refine Pegasus, a cutting-edge model for abstractive text summarizing so that it could better comprehend legal jargon and provide better summaries in this area. The ROUGE metric is used to assess the created summaries, as it gauges the degree of overlap between the generated and reference summaries. Users can obtain legal information in their preferred language by integrating the Google Translator API into the translation component. To improve accessibility for users who prefer or require audio formats, the text-to-speech conversion component makes use of the pyttsx3 package to provide audio output of the summarized documents.

It guarantees that legal information is communicated in an understandable and straightforward way. By tackling these issues, LegalEase hopes to democratise access to legal information and enable people to more confidently and clearly traverse the judicial system.

# 2 Related Work

Numerous research efforts have focused on improving the readability and accessibility of legal texts. These include developing domain-specific natural language processing (NLP) models tailored for the legal domain [1], aiming to capture the nuances and complexities of legal language. However, such models often require large amounts of annotated data and specialized training procedures, which can require significant time and resources. Fine-tuning pre-trained language models on legal corpora [2] has also been explored, leveraging the knowledge and language understanding capabilities of these models. However, this approach may struggle with domain-specific terminology and context, requiring additional fine-tuning or domain adaptation techniques. Extractive and abstractive summarization techniques have been applied to legal texts [3], aiming to generate concise and coherent summaries. While extractive methods select and combine relevant sentences from the original text, abstractive approaches generate new text that captures the essence of the document. However, these techniques may struggle with the complexity of legal language and the need to preserve critical details and nuances. Research has also focused on multilingual legal document processing [4], addressing the challenge of making legal information accessible to speakers of different languages. Approaches include creating language-independent models and methods for cross-lingual transfer learning. However, these methods may require extensive parallel data or suffer from performance degradation when applied to low-resource languages. User interface design for document processing tasks has been explored [5], aiming to create clear and intuitive interfaces that enhance user experience and accessibility. However, these efforts often focus on specific use cases or domains, and may not directly address the unique challenges and requirements of legal document processing. To assess the quality of the summaries produced, specific evaluation criteria have been proposed in legal frameworks such as ROUGE [6]. While these metrics provide a quantitative measure of performance, they may not fully capture the nuances and requirements of legal document summarization, such as preserving critical details and ensuring legal accuracy.

LegalEase tackles these issues by combining a number of different elements into a complete solution. To tackle the complexity of legal language, the Pegasus transformer model [7] was used, which is an advanced tool for abstractive text summarization that has been fine-tuned on a dataset of court cases. Multilingual access to legal information is made possible by the integration of the Google Translator API [8], and text-to-speech conversion is provided for improved accessibility through the use of the pyttsx3 package [9].

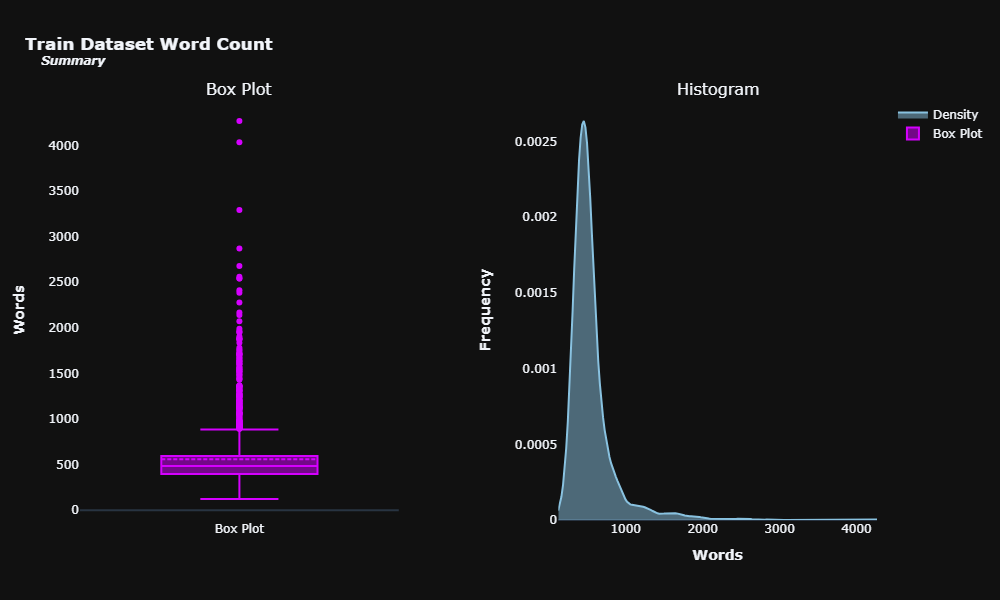
By integrating these elements, LegalEase intends to offer a user-friendly platform that ensures legal accuracy, preserves important information, and streamlines legal paperwork. ROUGE metrics are used to assess the platform's performance, giving an objective assessment of its summarizing abilities and pointing out areas that still require work. Furthermore, LegalEase addresses the challenge of potential biases or limitations in NLP models [10] by fine-tuning the Pegasus model on a diverse dataset of legal cases, aiming to mitigate biases and improve its performance across various legal domains. The platform also incorporates privacy and security measures [11] to protect sensitive information and ensure compliance with legal regulations. LegalEase also extends previous research that investigates using large language models and transfer learning techniques for processing legal documents [14-18]. These studies have investigated fine-tuning large pre-trained models such as BERT, RoBERTa, and GPT using legal datasets. They excelled at tasks like legal question answering, contract analysis, and case law retrieval. Multimodal approaches that combine text, images, and other modalities have also been studied for legal document processing [19-22]. These techniques seek to improve comprehension and analysis by making use of the wealth of information included in legal texts, including tables, figures, and other visual components. Various techniques such as homomorphic encryption [27], safe multi-party computation [26], and differential privacy [23–25] have been used to address privacy and security problems in the processing of legal documents. These methods seek to facilitate safe data sharing and cooperation while safeguarding sensitive information. In order to promote accountability and transparency in decision-making processes—two essential elements in the legal domain—explainable AI and interpretability in legal natural language processing (NLP) systems have also been investigated [28–30]. In general, LegalEase expands on this wide range of research by including cutting-edge methods for text-to-speech conversion, summarization, translation, and multimodality. It also tackles important issues related to security, privacy, and interpretability in the processing of legal documents.

# **3 Methodology**

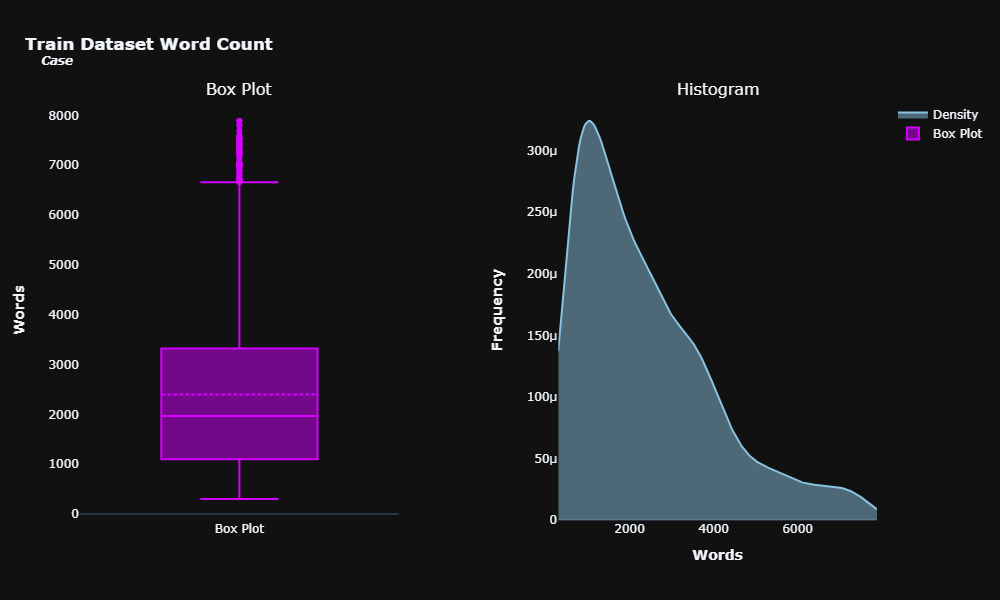
This paper investigates the Pegasus, BART, and LED text summarising models. The variety of summarising techniques is facilitated by the distinct capabilities and structures that each model offers. Although all three models demonstrate competence in producing abstractive summaries, differences in their methodologies and architectures affect how well they perform in particular scenarios.

## 3.1 Dataset description

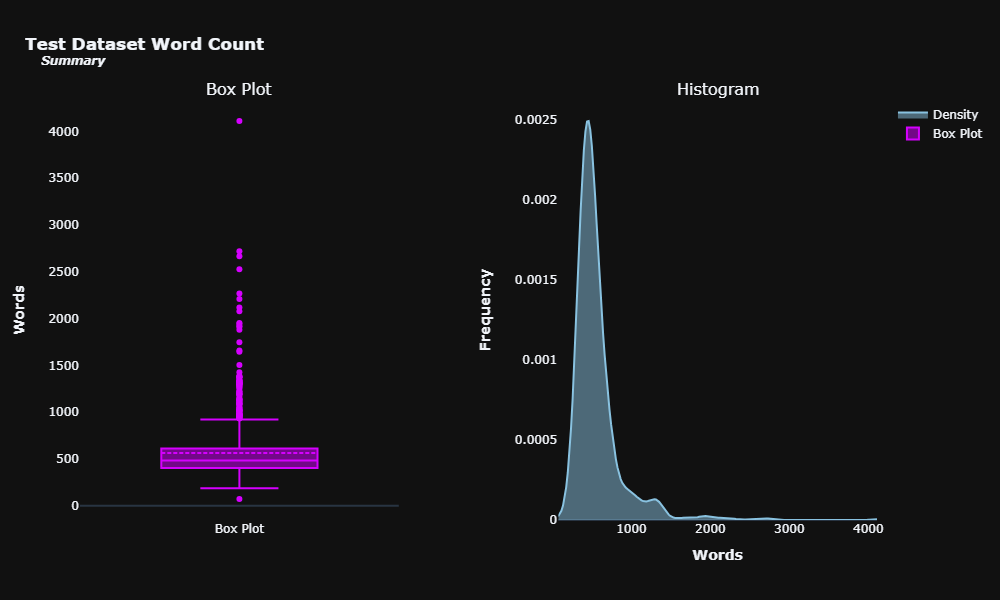
LegalEase is developed and assessed using the Indian Legal Case (ILC) dataset as a basis[7]. The dataset provides a rich source of diversified legal content, with a substantial collection of 3,073 court cases that are all supported by comprehensive descriptions and summaries. This extensive dataset was painstakingly assembled by carefully gathering information from multiple web sources, guaranteeing its legitimacy and relevancy. The dataset was intentionally split into 2,058 training cases and 1,015 test cases, facilitating effective model training and evaluation. The train and test cases consist of three features: Title, Summary and Case. This paper utilises Summary and Case columns for model implementation. Figure 1(a) and Figure 1(b) represent the total number of words present in the columns of the training dataset which helps us to analyse the average length of each column. Similarly, Figure 2(a) and Figure 2(b) represent the total number of words present in the columns of the test dataset. The box plot and histogram diagrams provide substantial evidence of balanced partitioning in the dataset, LegalEase can thoroughly validate its performance in a variety of legal contexts, which adds to its dependability and efficiency in streamlining legal papers. However, the dataset portrays some outliers with very extensive text going over 6,400 words per Case and over 850 words per Summary. Hence, we dropped all the cases with token lengths less than 300 and more than 6400 with their corresponding summaries as they can affect the performance of the trained model. We also dropped cases where the corresponding summaries token length was less than 140.



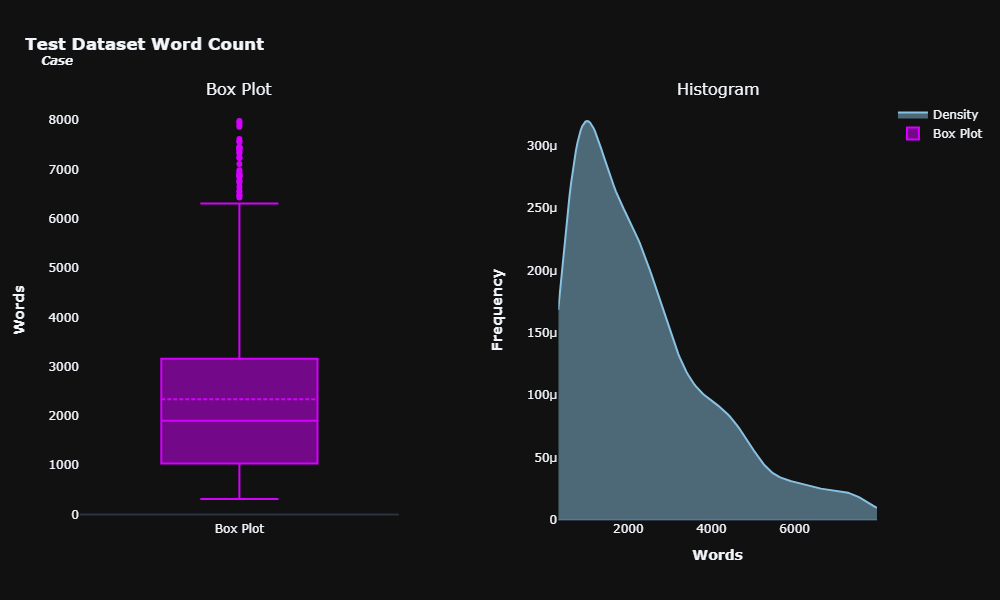
**Fig. 1(a).** Visual Representation of Summary column from the training dataset



**Fig. 1(b).** Visual Representation of Case column from the training dataset



**Fig. 2(a).** Visual Representation of Summary column from the test dataset



**Fig. 2(b).** Visual Representation of Case column from the test dataset

## 3.2 Model Exploration

We examined the efficacy of three different pre-trained summarization models from the hugging face platform for legal text processing during the early stages of development[8]. The models are fine-tuned on the ILC dataset for achieving optimal performance. The models are described as follows:

**3.2.1. BART Transformer Model**

This study looks into the BART transformer model for text summarization in addition to Pegasus. BART provides a strong foundation for creating abstractive summaries and is well-known for its bidirectional and auto-regressive features. This work assesses BART's suitability for summarising tasks in legal contexts by fine-tuning it on legal corpora and comparing its results with Pegasus. While BART is an effective language model, it may not be properly designed for the complex language and domain-specific terminology found in legal documents due to its pre-training on a large dataset of general text and code. This could result in summaries that are inaccurate in terms of law or exclude important information that is necessary to comprehend the legal substance.

**3.2.2 LED**

The LED model is an alternative method of text summarization that is recall-oriented and intended to extract pertinent information from input material. In contrast to Pegasus and BART, LED places more emphasis on recall than precision, to guarantee that the summaries produced capture all relevant information from the input documents. Although this method can be effective, it might not be the best choice for legal writings that require complicated information to be condensed and presented succinctly while maintaining essential ideas. The LED summary may be unfluent and exclude important information that isn't expressed clearly in individual phrases.

**3.2.3 Pegasus Transformer Model**

When it comes to abstractive text summarising tasks, the Pegasus model excels. Pegasus uses self-supervised learning objectives in its encoder-decoder architecture to provide logical and succinct summaries. Pegasus has been refined using a collection of Indian court cases to better understand and generate language in legal contexts. For abstractive summarising, this model makes use of pre-training with extracted gap sentences, which helps to preserve important information while guaranteeing the coherence and fluency of the summaries that are produced.

**Motives behind Selecting Pegasus:**

* In summarising legal papers, Pegasus does reasonably well even with marginally lower ROUGE scores than BART.
* Pegasus is more suited to the legal realm than BART and LED because it has been specifically optimised on legal corpora.
* Pegasus's design and training goals closely match the needs of summarising legal documents, which makes it a good fit for this purpose.

**3.3 Model Architecture**

Pre-training the Pegasus model on a huge corpus of text data is achieved by employing Gap Sentence Generation (GSG), a self-supervised objective function. In GSG, the model learns to produce cohesive, fluid writing while maintaining the crucial information from the input by being trained to anticipate the masked-out text spans from the input text.

The Transformer design, which comprises several layers of feed-forward networks and self-attention, is used by the Pegasus model. The feed-forward networks help with non-linear transformations of the input representations, while the self-attention mechanism allows the model to grasp long-range relationships within the input sequence.

Initially, the dataset is preprocessed i.e. cleaned to remove HTML tags, empty lines or spaces and get rid of duplicated words. To better align its language production and interpretation capabilities with the legal domain, the Pegasus model was refined using the ILC dataset. Before fine-tuning the pre-trained model on the ILC dataset, the data is converted into a format that the model understands which involves tokenizing the data. Tokenization and preprocessing were followed by determining how these tokenized data points are collated into batches for training. Using Hugging Face’s TrainingArguments class and Trainer class eased the training process.

The Pegasus model generation process first uses an encoder to initially encode the input legal document and the decoder to sequentially generate the summary token by token. The encoder's output representations, along with the previously generated tokens, are forwarded to the decoder. The decoder then produces a probability distribution across the vocabulary at each time step. The subsequent item in the summary is selected based on which token has the highest likelihood. The modules involved in the model’s implementation are illustrated through an architecture diagram using Figure 3 and they are explained as follows:

# 

**Fig. 3.** Architecture Diagram for LegalEase

**3.3.1 Summarization Model Methodology**

1. **Pre-training using Extracted Gap Sentences:** Using a sizable corpus of text data, the Pegasus model is pre-trained to predict masked-out text spans from the input text. Pre-training using extracted gap sentences is a self-supervised learning technique that helps the model learn to produce coherent and fluid text while retaining important information from the input.
2. **Encoder-Decoder Architecture:** The encoder and the decoder are made up of two primary parts. The encoder interprets the token sequence that is fed in and converts it into a series of continuous representations while preserving the input's context and semantics. The decoder produces the output sequence token by token, relying on the encoder's representations and the tokens generated previously
3. **Legal Corpora Fine-tuning:** Following pre-training, a collection of Indian court cases is used to refine the Pegasus model. By fine-tuning its language understanding and generation capabilities, the model becomes capable of producing summaries that are both pertinent and coherent in legal contexts.
4. **Generation Process:** During the generation process, the encoder component of the model is employed to initially encode the input legal document. One token at a time, the decoder produces the summary after receiving the encoded representations. The encoder's representations and the previously generated tokens are sent into the decoder at each time step, causing it to produce a probability distribution throughout the vocabulary. The next token in the summary is chosen based on whatever token has the highest likelihood.
5. **Evaluation using ROUGE Metric**: The ROUGE metric is used to evaluate the generated summaries' quality.

## 3.3.2 Language Translation

LegalEase utilizes the Google Translator API for language translation to offer multilingual access to legal information. The Google Translator API is a cloud-based machine translation service that translates text between different languages using cutting-edge neural machine translation models.

Similar to the Pegasus model used for summarization, the Google Translator API employs an encoder-decoder design based on the Transformer paradigm. The decoder generates the output word sequence in the target language based on these representations, while the encoder maps the input word sequence in the source language to a series of continuous representations.

The translation process involves several steps as follows:

**Text Preprocessing:** Tokenization and preprocessing are used to handle punctuation, special characters, and other formatting components in the incoming text.

**Encoding:** A learnt embedding layer is used to encode the preprocessed text into a series of numerical representations**.**

**Transformer Encoder**: Multiple layers of feed-forward and self-attentional networks make up the Transformer encoder, which processes the encoded sequence. The self-attention mechanism enables the model to extract contextual information from the input sequence and capture long-range dependencies.

**Transformer Decoder**: The Transformer decoder receives the encoder's output representations and uses them to create the output sequence in the target language one token at a time. Additionally, the decoder has an attention mechanism built in, which enables it to generate each output token by selecting and focusing on pertinent segments of the input sequence.

**Decoding and Postprocessing:** To handle formatting and special characters, the resulting output sequence is decoded back into text and any necessary postprocessing is carried out.

LegalEase can translate legal documents into many languages thanks to the extensive language support of the Google Translator API.

## 3.3.3 Text-to-Speech Conversion

LegalEase uses the pyttsx3 package to convert text to speech and produces an audio version of the condensed legal papers. Users who prefer or need audio formats will find greater accessibility thanks to this capability.

Text can be converted into spoken audio using the cross-platform text-to-speech engine provided by the Python module pyttsx3. The built-in Microsoft Speech API (SAPI) on Windows, NSSpeechSynthesizer on macOS, and the espeak and festival engines on Linux are just a few of the speech synthesis engines that it supports.

The following procedures are involved in the text-to-speech conversion process in PyTTSX3:

**Initialization:** The preferred speech synthesis engine is chosen and the pyttsx3 engine is initialised.

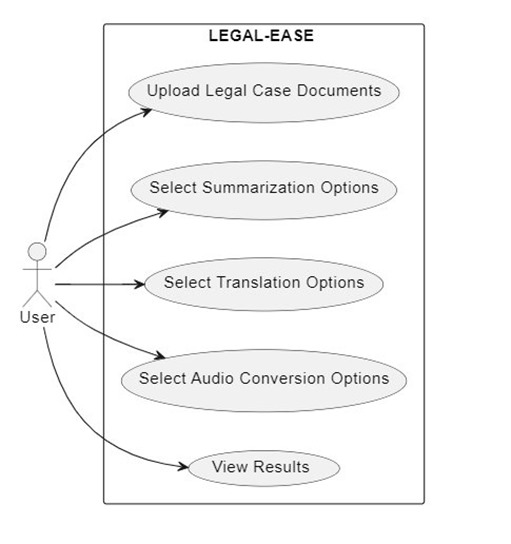
**Text Preprocessing:** To handle formatting, punctuation, and special characters, the input text is preprocessed.

**Speech Synthesis:** The speech synthesis engine receives the preprocessed text and turns it into audio data.

**Audio Output**: The system uses a standard audio output device to play back the generated audio data.

Pitch, volume, and speech rate are just a few of the configuration options available in the pyttsx3 library to personalise the voice output. Additionally, it enables LegalEase to offer text-to-speech translation in a variety of languages and accents by supporting numerous voices and languages.

By first using the Pegasus model to generate the summarised text and then providing it to the pyttsx3 library for conversion to audio format, LegalEase combines the text-to-speech feature. The user has the option to hear the generated audio output again or store it as an audio file for later use.



**Fig. 4.** UseCase Diagram for LegalEase

The use case diagram shown in Figure 4, depicts the interactions between the user and the system, including document upload, processing, and output generation.

## 3.4 Model’s Evaluation Metrics

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric calculates the overlap between the generated summary and the reference summary using different n-gram statistics.

ROUGE scores are computed using precision, recall, and F1-score metrics, which involve counting the overlapping n-grams (unigrams, bigrams, or longer sequences) between the generated summary and the reference summary. Mathematically, precision, recall, and F1-score are calculated as follows:

**Precision = (Number of overlapping n-grams) / (Total number of n-grams in the generated summary)** (1)

**Recall = (Number of overlapping n-grams) / (Total number of n-grams in the reference summary)** (2)

**F1-score = 2 \* (Precision \* Recall) / (Precision + Recall)** (3)

LegalEase specifically uses the following variants of ROUGE:

**ROUGE-N:** Evaluates the overlap of n-grams (sequences of n words) between the generated summary and the reference summary. ROUGE-1 and ROUGE-2, commonly used metrics, evaluate the overlap of unigrams (single words) and bigrams (word pairs), respectively.

**ROUGE-N = ∑gramn∈RefSummaryCount(gramn)/ ∑gramn∈RefSummaryCountmatch(gramn)**  (4)

where:

**Countmatch(gramn)** represents the maximum number of n-grams co-occurring in both the generated summary and the reference summary.

**Count(gramn)** denotes the total number of n-grams in the reference summary.

**ROUGE-L:** Measures the longest common subsequence (LCS) between the generated summary and the reference summary. It computes the length of the longest subsequence of words that is present in both the generated summary and the reference summary.

**ROUGE-L= LCS(RefSummary,GenSummary) / len(RefSummary)** (5)

where:

**LCS(RefSummary, GenSummary)** represents the length of the longest common subsequence between the reference summary and the generated summary.

**len(RefSummary)** represents the length of the reference summary.

These ROUGE variants offer insights into the model's capability to capture essential information, preserve context, and generate coherent summaries.

# 4 Results

As mentioned above, the performance of LegalEase's summarization component is evaluated using the ROUGE metric, which measures the overlap between the generated summaries and the reference summaries. Specifically, the ROUGE-1, ROUGE-2, and ROUGE-L variants are used to assess different aspects of the summarization quality.

Table 1 presents the ROUGE scores obtained by the Pegasus model on the ILC dataset for both the training and test data.

**Table 1**. ROUGE scores for the Pegasus model on the ILC dataset

| **Metric** | **Train Data** | **Test Data** |
| --- | --- | --- |
| ROUGE-1 | 0.10386 | 0.08314 |
| ROUGE-2 | 0.02533 | 0.01299 |
| ROUGE-L | 0.09435 | 0.07417 |

The ROUGE-1 scores indicate that the Pegasus model is reasonably effective in capturing individual words from the reference summaries, with a higher score on the training data compared to the test data. This suggests that the model has learned to identify and include relevant keywords and terms from the legal documents in the generated summaries. However, the lower ROUGE-2 scores, particularly on the test data, suggest that the model struggles with capturing longer sequences and preserving context. The ROUGE-2 metric evaluates the overlap of bigrams (word pairs) between the generated and reference summaries, which is essential for maintaining coherence and capturing the logical flow of information. Similarly, the ROUGE-L scores, which measure the longest common subsequence between the generated and reference summaries, highlight room for improvement in generating summaries that accurately capture the essential details and key points of the legal documents. The discrepancy between the training and test data scores may indicate potential overfitting, where the model performs better on the data it was trained on but struggles to generalize to unseen data. This may be due to the complexity and diversity of the legal language, but also due to the limited size and scope of training data.

# 5 Discussion

While the Pegasus model outperformed other models in the experiments, the ROUGE scores suggest that further fine-tuning and optimization are required to enhance the summarization quality for legal documents. Potential areas for improvement include:

**Dataset Expansion:** Incorporating additional legal datasets from diverse sources and domains could help improve the model's generalization capabilities and expose it to a wider range of legal terminology and language patterns.

**Domain Adaptation:** To further improve model performance on legal documents, it may be useful to investigate techniques such as domain-specific pretraining, multi-task learning, or domain-adaptive fine-tuning could be beneficial. These approaches can help the model better capture the variations and complexities inherent in the legal domain.

**Post-processing Techniques:** Investigating post-processing methods, such as re-ranking, filtering, or refinement of the generated summaries, could potentially improve their coherence, accuracy, and overall quality.

**Evaluation Metric Refinement:** While ROUGE is a commonly used metric for summarization evaluation, it may not fully capture the nuances and requirements of legal document summarization. Exploring alternative or specialized evaluation metrics tailored to the legal domain could offer a more comprehensive evaluation of the model's performance.

Despite the limitations identified, the Pegasus model's performance on the ILC dataset demonstrates its potential for simplifying legal documents through summarization. With further research and development, LegalEase can leverage advanced techniques to enhance the quality and accuracy of the generated summaries, ultimately improving the accessibility and understanding of legal information for a broader audience.

# 6 Conclusion

LegalEase presents a promising solution to simplify legal documents and improve accessibility to legal information through AI-powered summarization, language translation, and text-to-speech conversion. By leveraging the Pegasus transformer model and integrating translation and audio output capabilities, LegalEase aims to bridge the gap between complex legal language and layman's understanding. The evaluation results demonstrate the potential of the Pegasus model in summarizing legal documents, as evidenced by the ROUGE scores obtained on the ILC dataset. However, the results also highlight areas for further improvement, particularly in capturing longer sequences, preserving context, and generating summaries that accurately capture the essential details of legal documents. Future work should focus on exploring advanced fine-tuning techniques, incorporating additional legal datasets, and investigating post-processing methods to enhance the coherence and accuracy of the generated summaries. Potential avenues for improvement include dataset expansion, domain adaptation, post-processing techniques, and refinement of evaluation metrics tailored to the legal domain. Furthermore, LegalEase can be extended to support a wider range of languages for translation and text-to-speech conversion, increasing its global reach and accessibility. Integrating advanced techniques such as few-shot learning, multi-task learning, and language model fine-tuning could further enhance the summarization and translation capabilities of the platform. LegalEase has the potential to empower individuals by simplifying legal documents, enabling them to make informed decisions and navigate the legal landscape with increased confidence. By democratizing access to legal information, LegalEase aligns with the principles of transparency, fairness, and equal access to justice. As legal technology continues to evolve, LegalEase can incorporate cutting-edge developments in areas such as multimodal document processing, privacy and security enhancements, and explainable AI for legal applications. Continuously exploring and integrating these advancements will ensure that LegalEase remains at the forefront of legal document simplification and accessibility, further empowering individuals and organizations to navigate the legal landscape with clarity and confidence.

# 7 Future Scope

LegalEase holds promise in simplifying legal documents and enhancing accessibility, but several avenues for future improvements exist. These include expanding language support, integrating advanced NLP techniques, incorporating legal databases and knowledge graphs, implementing personalization features, accommodating multimodal documents, enhancing privacy and security measures, and optimizing deployment and scaling. By continually embracing cutting-edge technologies, LegalEase can maintain its position at the forefront of legal document simplification, empowering users to navigate the legal landscape confidently and clearly.

# References

1. Garimella, A., Sancheti, A., Aggarwal, V., Ganesh, A., Chhaya, N., & Kambhatla, N. (2022, December). Text Simplification for Legal Domain:{I} insights and Challenges. In *Proceedings of the Natural Legal Language Processing Workshop 2022* (pp. 296-304).
2. Anusha, S., & Praveen, K. S. ADVANCE E-LAWYER APPLICATION PORTAL
3. Alves, A., Miranda, P., Mello, R., & Nascimento, A. (2023). Automatic Simplification of Legal Texts in Portuguese Using Machine Learning. In *Legal Knowledge and Information Systems* (pp. 281-286). IOS Press.
4. Τζωρτζίδου, Ε. (2020). Exploring the current state of the legal services market: the rise of legal tech and the effects of digital transformation.
5. Dragoni, M., Villata, S., Rizzi, W., & Governatori, G. (2016, December). Combining NLP approaches for rule extraction from legal documents. In *1st Workshop on MIning and REasoning with Legal texts (MIREL 2016)*.
6. Khandelwal, A., Agrawal, H., Jain, R., Srivastava, A., & Jain, P. (2021, June). Automated Legal Text Summarization: Experiments with Summary Structuring. In Proceedings of the Natural Legal Language Processing Workshop 2021 (pp. 78-87). Association for Computational Linguistics.
7. Trivedi, P., Jain, D., Gite, S., Kotecha, K., Bhatt, A., & Naik, N. (2023). Indian Legal Corpus (ILC): A Dataset for A Dataset Summarizing Indian Legal Proceeding Using Natural Language. Engineered Science, 27, 1022.
8. https://huggingface.co/transformers/v3.2.0/main\_classes/model.html
9. Leitner, E., Rehm, G., Moreno-Schneider, J. (2019). Fine-grained Named Entity Recognition in Legal Documents. In Proceedings of the Natural Legal Language Processing Workshop 2019. Association for Computational Linguistics.
10. Chalkidis, I., Androutsopoulos, I., & Aletras, N. (2018). Legal document anonymization with recurrent neural networks. Artificial Intelligence and Law, 26(4), 335-354.
11. Hadiwinoto, C., Ng, H. T., & Gan, W. B. (2018). Extracting conditions and exceptions from contracts using SVM and heuristic rules. In International Conference on Artificial Intelligence and Law (pp. 53-62). Springer, Cham.
12. Lippi, M., Palka, P., Contissa, G., Lagioia, F., Micklitz, H. W., Panagis, Y., & Sartor, G. (2017, June). Automatic Detection of Unfair Clauses in Online Consumer Contracts. In Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law (pp. 145-154). ACM.
13. Pandya, V. (2019). Automatic text summarization of legal cases: A hybrid approach. arXiv preprint arXiv:1908.09119.
14. Parikh, V., Mathur, V., Mehta, P., Mittal, N., & Majumder, P. (2021). Lawsum: A weakly-supervised approach for Indian legal document summarization. arXiv preprint arXiv:2110.01188.
15. Anand, D., & Wagh, R. (2022). Effective deep learning approaches for summarization of legal texts. Journal of King Saud University-Computer and Information Sciences, 34(5), 2141-2150.
16. CORE Homepage, <http://www.springer.com/lncs>
17. Shukla, S., Bhattacharyya, P., & Hazarika, D. (2021). Legal Statutes Simplification: Datasets and Baselines. arXiv preprint arXiv:2106.12624.
18. Britz, K., & Lippoldt, J. (2022). Towards Automating Due Diligence: Extracting Representations and Warranties from M&A Contracts with Transfer Learning. In Proceedings of the Natural Legal Language Processing Workshop 2022 (pp. 163-173).
19. Kaminaka, M. S. (2022). Automatic Summarization of Court Decisions: A Benchmarking Study. arXiv preprint arXiv:2204.09186.
20. Saha, S., Pahuja, V., Khapra, M. M., Sankaranarayanan, K., & Chandar, S. (2018). Complex word identification using reinforcement learning. arXiv preprint arXiv:1811.07465.
21. Zhang, L., Wu, Y., & Li, X. (2014, May). Pruning and parallel practice of dependency parsing for legal text. In International Conference on Knowledge Science, Engineering and Management (pp. 263-269). Springer, Cham.
22. Zhang, L., Li, X., & Rahman, A. (2013). Towards a unified approach for linking textual definitions and illustrations in digitized law texts. Artificial Intelligence and Law, 21(1), 1-29.
23. Wu, J., Shang, Z., Li, H., Chen, X., & Dong, W. (2022). Legal Pre-trained Language Models for Legal Document Understanding. arXiv preprint arXiv:2211.02596.
24. Araci, D. (2019). Towards automated computational fact-checking. Proceedings of the Natural Legal Language Processing Workshop 2019, 143-148.
25. Goldstein, A., Shaham, R., & Geva, S. (2021). Towards Automatic Fact-Checking for Legal Decision Documents. arXiv preprint arXiv:2112.00399.
26. Tuggener, D., & Aizawa, A. (2020). Automatic Extraction of Factual Information from Legal Documents. arXiv preprint arXiv:2001.04448.
27. Mahesan, N., Hettiarachchi, S., Abeysinghe, E., & Sandaruwan, D. (2022). Application of Deep Learning Techniques for Legal Document Summarization. arXiv preprint arXiv:2211.12847.
28. Shah, P. A., Kolli, A., Gupta, M., & Verma, V. (2022). Legal Query Understanding and Evidence Retrieval Using Transformer Models. arXiv preprint arXiv:2209.03998.
29. Arora, A., Uppal, S., Ahmad, N., & Kaur, M. (2022). CL-Norm: A Norm Extraction and Classification Dataset for Indian Legal Documents. arXiv preprint arXiv:2201.08761.