RAG InLAvS: RAG based Indian Legal Advisory System

Mudit Gupta 2020315
Devansh Arora 2020053
Srishti Jain 2020543
Aditya Choudhary 2020489
Prasun Pratik 2020101
Shrugal Tayal 2020408

Abstract

This project aims to develop a smart legal advisory system that can provide relevant and accurate information to users based on their queries and feedback. The system will use natural language processing and information retrieval techniques to understand the user's intent, context, and preferences and retrieve the most suitable information from a large corpus of legal documents. Our system will particularly cater to the Indian audience and utilize formal and informal datasets to learn about India's legal system. The system will handle complex and diverse legal scenarios and interact with users more naturally by utilizing large language models. (Find code here: Github)

1 Introduction

1.1 Problem Statement

Traditional legal research and advice provision methods often suffer from time constraints, information overload, and limited user interaction and improvement avenues.

To address these challenges, we aim to develop an innovative chatbot system that combines advanced natural language processing (NLP) techniques with an efficient information retrieval system to deliver intelligent legal advice tailored to an Indian user's needs.

1.2 Importance of Problem Statement

In India, the need for a legal advice chatbot stems from challenges like limited judicial capacity and difficulties accessing legal aid. With only 0.5% of Indians accessing legal aid despite an 80% entitlement rate, there's a crucial gap that demands swift and scalable solutions. We provide a tool capable of providing legal information at the user's fingertips.

Most importantly, very few legal chatbots are trained on Indian laws and the problems that Indian users face.

2 Related Work

2.1 LAWBO: A Smart lawyer Chatbot

LAWBO is a chatbot aiding lawyers and legal researchers in judicial system information retrieval. It employs heuristics, dynamic memory networks (DMN), and GloVe word representation for NLP, facilitating legal advice provision and case comparison. Capable of understanding user intents, transitive reasoning, and relevant information retrieval, LAWBO iteratively improves performance through user feedback. This innovative application merges legal research, feedback integration, and efficient information retrieval [1]

2.2 Legal Document Summarizer

Shukla et al. (2022) analyze both extractive and abstractive summarization methods for legal texts[2], from unsupervised LexRank and DSDR to supervised Legal-Pegasus. The study promotes using targeted extractive techniques and sophisticated abstractive models like BART[3] and Longformer[4], emphasizing the importance of finetuning on legal datasets and advocating for domain-specific evaluation to align with expert opinion.

2.3 From Neural Re-Ranking to Neural Ranking

The paper explores extending neural re-ranking methods to optimize the ranking process directly, utilizing neural networks to create a sparse representation for efficient inverted indexing. Zamani et al. (2018)[5] evaluate their models in ad-hoc retrieval, demonstrating comparable performance to state-of-the-art models and highlighting the effectiveness of pseudo-relevance feedback in the learned latent space, outperforming baselines.

2.4 Exploiting Simulated User Feedback for Conversational Search

Addressing challenges in conversational search, Owoicho et al. (2023)[6] introduce ConvSim, a user simulator utilizing LLMs for multi-turn inter-

actions. Their findings reveal that incorporating feedback consistently enhances retrieval, demonstrating a +16% improvement in nDCG@3 after a single turn and a +35% boost after three rounds. This highlights promThe article [8] serves as a guide to explore the integration of human feedback into chatbots, emphasizing rewarding mechanisms. It addresses methods for soliciting and utilizing feedback to enhance chatbot performance. The study offered valuable insights for effectively implementing human feedback systems in chatbot development and highlights strategies for integrating rewarding mechanisms into the feedback systems effectively.ising prospects for advancements in Conversational Search, urging further exploration of feedback processing methods

2.5 Improving Access to Justice With Legal Chatbots

The paper presents the problem of access to justice and how legal chatbots can help provide legal information to people who cannot afford legal services or representation. The paper describes the tools and data they used to develop their chatbots. They explain how they collected and annotated legal documents and questions and used the MLflow framework to track and reproduce their experiments. It also mentions the challenges and difficulties faced in this process. One example being LegalGPT which is a variant of the GPT model specifically tailored for legal text generation and understanding, its models are trained on data from specific jurisdictions and may not generalize well to other legal systems, limiting their applicability in global or diverse legal contexts.

2.6 Rewarding Chatbots for Real-World Human Feedback

The article serves as a guide to explore the integration of human feedback into chatbots, emphasizing rewarding mechanisms. It addresses methods for soliciting and utilizing feedback to enhance chatbot performance. The study offered valuable insights for effectively implementing human feedback systems in chatbot development and highlights strategies for integrating rewarding mechanisms into the feedback systems effectively.

3 Novelty

Our project stands out by utilizing and training itself on Indian laws and context while incorporating a robust feedback loop and reward mechanism. We focus on dealing with situations that Indian citizen users commonly encounter and using an up-to-date law database to help Indian users keep up with the ever-changing rules in India. We also try to improve user engagement through a friendly feedback mechanism, promptly addressing issues and refining the chatbot's responses. This approach actively engages users, contributing to India's more inclusive legal landscape.

4 Dataset

4.1 About the dataset

We have utilized two datasets to solve our problem statement. The first one is a novel dataset that we are calling 'LegalActsDB.' This dataset consists of formal information, including laws, acts, and articles introduced in the Indian constitution between 2009-2019. This dataset is used to gather context for the user query. The second dataset is a publicly available dataset derived from r/LegalAdviceIndia, an Indian subreddit (social media platform). This is somewhat of an informal dataset used to fine-tune our large language model, which is used for response generation.

4.2 Data Collection

The key step for our data collection was web scraping. Both LegalActsDB and our informal datasets are publicly available on the internet. To collect the LegalActsDB, we went to the official website of the government of India and manually went through the bare acts that were available. We decided to only take acts between the period 2009-2019 because they seemed the most relevant. We also had to keep the dataset size in check since we only have limited computational capacity.

4.3 Data Analysis

We perform all of our data analysis on our novel dataset, "LegalActsDB" since that is the more important dataset, as it is being used for context recognition. We begin by breaking our files into a large number of chunks. This is done to bring individual data samples down to a size that is easily manageable by our machines. All analysis hereon is done chunk-wise.

Figure 1 displays the distribution of mean word lengths across different text chunks or segments. This boxplot suggests that the text being analyzed consists primarily of shorter and simpler words.

The distribution of the number of words per

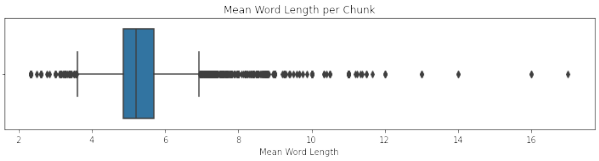

Figure 1: Average Length of Tokens in a Chunk

chunk sis highly skewed with a significant number of chunks containing a large number of words (up to around 1200). The typical range, in most chunks is between 20 and 80 words. The top-occurring words unigrams are "service," "act," "authority," "government," etc but they don't tell us anything. Thus, we also plot the bi-gram frequency chart. Figure 2 gives us better results. We can see that in the last five years, most laws and acts introduced were on the topics of "goods and services," "mental health," "corporate debtor," and "service tax."

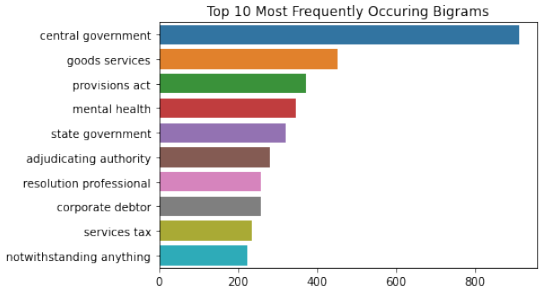

Figure 2: Bi-Gram Frequency chart

4.4 Data Processing

From the collected dataset, that is the list of acts from 2009 to 2019, we used Google Gemini to generate 15 questions per act and asked to generate answers for the same and stored all the questions and answers in a CSV file, by the name real.csv, so that we can train our model on it. We did the same thing using a weak LLM and stored the questions and answers in a separate CSV file and named it fake CSV. The purpose of this was to train the classifier so that it can identify real data.

5 Methodology

5.1 Data Handling and Query Processing

We extract text from a pdf, split it into titles and paragraphs, combine paragraph text into single rows, and save the data as a CSV using PyPDF2, csv, and Pandas libraries. The code defines functions for pdf text extraction, DataFrame processing, and csv saving.

5.2 Data Retrieval Techniques and Implementation

The project efficiently extracts legal information from texts using IR techniques, streamlining retrieval of relevant legal documents. It tackles the time-consuming manual search process for legal professionals, exploring key approaches like BM25 with N-grams and other indexing techniques.

To improve retrieval from a large dataset, the project enhances speed and accuracy by using techniques like the Inverted Index for quick document lookup. TF-IDF scores each document's relevance based on term frequency and rarity, aiding in efficient retrieval.

N-gram indexing, including Bi-gram and Tri-gram, aids information retrieval. A Bi-gram approach constructs a system capable of handling keyword and Boolean queries. The project uses pandas DataFrame to implement the BM25 algorithm for relevance scoring. Additionally, Tri-gram indexing combined with BM25 offers a comparative analysis.

Optimizations such as SpaCy preprocessing and parallel processing improve efficiency, with the chosen approach depending on factors like dataset size and the desired balance between accuracy and efficiency in legal text retrieval.

5.3 Custom Data Generation and Classification

We generated a custom dataset by manually going through our LegalActsDB and making some custom queries. We then passed these queries through Gemini (Google's LLM) to attain samples belonging to class 'Good' and the responses from vanilla 'Mistral' LLM belonged to class 'Bad'. Then, we trained a BERT classifier on this data to get results for our Custom Classification Task. In our recent advancements within the Reinforcement Learning from Human Feedback (RLHF) pipeline, we have successfully implemented the trained classifier as a value head. This integration plays a crucial role in enhancing model alignment by effectively discerning and adhering to patterns of real-world accuracy and contextual relevance. The classifier's ability to distinguish between more and less accurate responses allows it to guide the model towards generating outputs that are not only contextually appropriate but also accurately reflect real-world data and scenarios. This step marks a significant enhancement in our approach to developing

AI systems that can interact more naturally and effectively with human users.

5.4 Prompt Generation and Response Handling

In the initial phase of the process, relevant contextual information is systematically extracted from the document corpus. Following retrieval, this data is subjected to a preprocessing stage to optimize the efficiency of the language model's generation capabilities. This stage involves carefully constructing prompts designed with precision to evoke meaningful responses from the Mistral AI-7B model. Subsequently, these meticulously crafted prompts are fed into the model to facilitate the generation, employing the context provided. Various generative parameters are fine-tuned to enhance the generated output's quality.

Model	Hallucination Score	Context Adherence Score	Generation Score
SOTA	0.11	0.89	0.95
Benchmark	0.15	0.85	0.9
{RagInLavs}	0.25	0.76	0.7
RagInLavs(Midsem)	0.59	0.47	0.3

Table 1: Evaluation Results

Epoch	Accuracy	Precision	Recall
1	78.78%	85.29%	69.88%
2	81.24%	71.43%	96.39%

Table 2: Custom Classification Task Results

6 Evaluation

The evaluation methodology employed in this analysis involved assessing the quality of responses based on three key parameters: Hallucination index, Context Adherence, and Generation Quality. The hallucination index measured the degree to which the responses introduced information not directly supported by the given context, with a score of 0 indicating no hallucination and 1 indicating a fully hallucinated response. Context adherence was scored on a scale of 0 to 1, with 1 representing complete adherence to the provided context. Generation quality was evaluated on a scale of 0 to 1, considering factors such as coherence, relevance, and the ability to directly address the given question. For each prompt-answer pair, scores were assigned for these three parameters. The individual scores were then averaged to calculate the overall scores for each CSV file, providing a comprehensive assessment of the responses' quality in terms of hallucination, context adherence, and generation. This methodology allowed for a systematic and quantitative evaluation of the generated responses, enabling a comparative analysis between different CSV files.

7 Conclusion

Our methodology of combining multiple novel datasets, processing using various NLP techniques, including multiple IR techniques, and introducing machine learning with custom classification tasks exceeded expectations in terms of performance. We can reach benchmark (chatgpt 3.5) results with substantially less data. With further inclusion of human feedback, the model is only expected to keep performing better.

8 Contribution

Equal contribution of all members-

- Mudit Gupta, Devansh Arora: Implemented ML and DL models for IR functionality, developed human feedback technique, and Integrated chatbot with IR for a user-friendly interface.
- Srishti Jain, Aditya Choudhary: Processed data, developed chatbot UI, and worked on indexing techniques.
- Shrugal Tayal, Prasun Pratik: Developed NLP query processing methods and worked on filtering techniques and optimizations.

9 References

1. G, S., N, U., & G, K. (2018, January 11). *LAWBO*. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. https://doi.org/10.1145/3152494.3167988.

2. Shukla, Abhay, et al. "Legal case docu-

ment summarization: Extractive and abstractive methods and their evaluation." arXiv preprint arXiv:2210.07544 (2022).

3. Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461 (2019).

4. Beltagy, Iz, Matthew E. Peters, and Arman Cohan. "Longformer: The long-document transformer." arXiv preprint arXiv:2004.05150 (2020).

5. Zamani, Hamed, et al. "From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing." Proceedings of the 27th ACM international conference on information and knowledge management. 2018.

6. Owoicho, Paul, et al. "Exploiting simulated user feedback for conversational search: Ranking, rewriting, and beyond." Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023.