CJPR: Court Judgment Prediction and Recommendation

A project submitted in partial fulfillment of the
Requirements for the award of the degree of
Bachelor of Technology
in
Computer Science and Engineering

Submitted By:
Sahil Bharodiya
Roll No.: 12011008

Supervised By:
Dr. Bhupesh Bhatti

Indian Institute of Information Technology
Sonepat

Acknowledgment:

In the present world of competition, there is a race of existence in which those with the will to succeed. A project is like a bridge between theoretical and practical working. With this willingness, I joined this particular project. First of all, I would like to thank the supreme power of the Almighty God who is obviously the one who has always guided me to work on the right path in life. Without his grace, this project could not become a reality. Next to him are my parents, to whom I am greatly indebted for me brought up with love and encouragement to this stage. I am feeling obliged to take the opportunity to sincerely thank Dr. M. N. Doja (Director of IIIT Sonepat) and special thanks to my worthy teacher of Computer Science Dr. Bhupesh Bhatti. Moreover, I am highly obliged in taking the opportunity to sincerely thank all the staff members of the CSE department for their generous attitude and friendly behavior. Last, but not least I am thankful to all my teachers and friends who have been always helping and encouraging throughout the year. I have no valuable words to express my thanks, but my heart is still full of favors received from every person.

Sahil Bharodiya (12011008)

Self Declaration:

I hereby that work contained in the project titled "**CJPR: Court Judgment Prediction & Recommendation**" is original. I have followed the standards of projects ethics to the best of my abilities. I have acknowledged all sources of information which I have used in the project.

GitHub Link:

Name: Sahil Bharodiya
Roll No.: 12011008

Department of Computer Science and Engineering
Indian Institute of Information Technology,
Sonepat – 131201, Haryana, India

Certificate:

This is to certify that Mr. Sahil Bharodiya has worked on the project entitled "CJPR: Court Judgment Prediction and Recommendation" under my supervision and guidance. The contents of the project, being submitted to Department of Computer Science and Engineering, IIIT, Sonepat, for the award of the degree of B. Tech in Computer Science and Engineering, are original and have been carried out by the candidate himself. This project has not been submitted in full or part for the award of any other degree or diploma to this or any other university.

Dr. Bhupesh Bhatti
Supervisor

Department of Computer Science & Engineering
Indian Institute of Information Technology
Sonepat-131201, Haryana, India

Table of Contents:

List of Abbreviation:

CJPR – Court Judgment Prediction and Recommendation
ILDC – Indian Legal Document Corpus
ML – Machine Learning
DL – Deep Learning
TF-IDF – Term Frequency Inverse Term Frequency
LR – Logistic Regression
SVM – Support Vector Machine
KNN – K Nearest Neighbors
RF – Random Forest
BiGRU – Bidirectional Gated Recurrent Unit
GCP – Google Cloud Platform
AWS – Amazon Web Service

Abstract:

There is a large dependency of cases in our country. The time taken runs into years more as a norm rather than an exception. Majority of the cases would be a repeated nature, with previous decisions available for guidance. Creation of database of all judgments in various cases and creation of a smart legal aid using AI would help in assessing the condition of each petition and thus may enable reduction in timelines. Any other technology based solution to reduce dependency timelines may also be identified. Technology could also be available to advise common citizens so as to enable them to understand where they stand before they file their petitions. It should also be able to predict the likely timelines for a case to see its final judgment, when petitioned in a court of law.

1. Introduction

The Indian legal system is plagued by an enormous backlog of cases, with the time taken to reach a verdict often running into years. This delay can lead to significant consequences for litigants, with cases sometimes losing their relevance over time or causing undue hardship to the parties involved. To address this issue, our project aims to develop a smart legal aid system using AI to help assess the condition of each petition and reduce the time taken to reach a final judgment.

One of the main components of our project will be the creation of a database of all judgments in various cases. This will help identify patterns and provide guidance for future cases, potentially reducing the time taken to reach a verdict. Additionally, we will leverage machine learning algorithms to predict likely timelines for a case to see its final judgment when petitioned in a court of law. This information will help litigants make informed decisions about their legal matters and enable them to plan accordingly.

Our project also aims to develop a user-friendly web application that can provide quick and accurate predictions and recommendations. The application will be designed to cater to the needs of individuals who may not have a legal background and may be unfamiliar with the legal system. By providing common citizens with a tool to understand where they stand before they file their petitions, we hope to empower them to make informed decisions about their legal matters.

In summary, our project aims to leverage technology-based solutions to reduce pendency timelines and make the legal system more accessible and transparent for all citizens.

2. Literature Overview

The problem of the backlog of cases in the Indian legal system has been a topic of discussion for several years. A number of studies have attempted to identify the root causes of the problem and suggest possible solutions.

One of the primary reasons for the backlog is the lack of resources and infrastructure, including judges and courtrooms. A study by the National Judicial Data Grid (NJDG) showed that there were over 30 million pending cases across various courts in India as of December 2020. The NJDG report also highlighted the need for technology-based solutions to help reduce pendency timelines.

Several previous studies have explored the use of technology in the legal system to address this issue. For instance, machine learning algorithms have been used to predict case outcomes based on previous judgments. A study by the Indian Institute of Technology (IIT) Kharagpur explored the use of natural language processing techniques to analyze judgments and provide insights into judicial decision-making.

Other studies have also focused on developing AI-powered tools to assist lawyers and judges in legal research and case analysis. For instance, the Indian government's e-Courts project aims to provide a common platform for all courts in India to automate their processes and make them more efficient.

In summary, while the problem of the backlog of cases in the Indian legal system is a complex one, there have been several efforts to address it using technology-based solutions. Our project aims to contribute to this body of research by developing a smart legal aid system using AI to assess petitions and predict case outcomes, as well as providing recommendations to litigants. By leveraging technology to make the legal system more efficient and accessible, we hope to make a positive impact on the lives of citizens across India.

3. Techniques and Methods

Data Collection: The first step is to collect data on previous court cases and their judgments. This can be done by accessing online databases, contacting court registries, and gathering information from legal experts. The data should be organized in a structured format to make it easy to analyze. A very few dataset is available for this type of task.

Data Cleaning: Once the data is collected, it needs to be cleaned and standardized. This involves removing irrelevant information, correcting errors, and ensuring consistency in the data format. Fortunately, this data is already well prepared, so no need to clean the data.

Data Analysis: The cleaned data is then analyzed to identify patterns and trends in previous judgments. This can be done using machine learning algorithms and statistical techniques to develop predictive models for future cases. Data has total 35,000 corpus of supreme court cases, each corpus is very long petition more then lakh characters. Data is splited into three parts train, valid, dev. Train for training purpose, valid for testing purpose and dev for cross validation.

Development of Judgment Prediction: A ML-DL based system will be developed for prediction of judgment whether it should be accepted or rejected. Various ML-DL models will be used.

Testing and Validation: The legal aid system should be tested and validated using a variety of scenarios to ensure accuracy and reliability. This can involve testing the system on a set of historical cases and comparing the system's predictions to the actual outcomes.

Development if Recommendation System: If the petition is accepted then similar petition filed in history will be recommended using recommended system model. One can modify how many similar petition should be recommended. This will help judges to quickly take decision based on historical petitions.

User Interface Design: A user interface for the legal aid system should be designed that is user-friendly and easy to understand. This involves identifying the key features and functionalities that are most relevant to the users and designing an interface that is intuitive and visually appealing.

Deployment and Maintenance: Once the CJPR system is developed and tested, it can be deployed for use by citizens and legal professionals. Ongoing maintenance and updates should be performed to ensure the system remains accurate and up-to-date with changes in the legal landscape.

In summary, the techniques and methods for this project involve collecting and cleaning data, analyzing previous cases to identify patterns, developing a legal aid system using ML-DL, testing and validating the system, designing a user-friendly interface, and deploying and maintaining the system for ongoing use.

4. Dataset

For this type of problem one need to use data that has real court cases datasets including petition and its label whether it is accepted or rejected. This type of data also need to well prepared as it involves NLP task. Looking into internet similar data is available in chinese language titles Chinese AI and Law Challenge Dataset (CAIL 2018) [1] for judgment prediction which has 2.68 million court cases in chinese language published by China's Supreme Court. There is also an English language dataset published which contains 11,478 cases of European Court of Human Rights (ECHR) [2]. But any of these data is not useful for this project because this targets Indian audience. So there is need of data which only contains Indian court cases data.

Hence there is a professor in IIT Kanpur named Dr. Ashutosh Modi. His team has prepared a dataset named ILDC: Indian Legal Document Corpus of 35,000 well prepared supreme court cases [3]. One have to send a valid request to him on his mail ashutoshm.iitk@gmail.com, if reason is genuine and valid then he will provide access to dataset after accepting term that will not publish the data online.

The data contains csv file that has 35,000 supreme court cases data column names 'text', label as 'accepted' or 'rejected' and split as 'train', 'test', 'dev' for which category particular belongs to. The data is already well prepared for CJPR task. So one can directly jump into training part.

5. Training and Evaluation

Since the data is already well prepared the model training can be done using following methods:

1. Classical models
2. Sequential models
3. Transformers
4. Hierarchical models

1) Classical Models:

This type of models include traditional machine learning algorithms for supervised classification tasks like Logistic Regression, SVM (Support Vector Machine), KNN (K Nearest Neighbors).

Before starting training data should be converted into text format to numerical format. There are many methods available for text encoding. Text encoding method are also of two types. i) traditional and ii) ml based. Traditional method used mathematical formula for text encoding like TF-IDF and ml based text encoding method includes ml algorithm like Sent2Vec [4], Doc2Vec[5]. Here is the results of classical model training:

Model	F1_Score
Doc2Vec + LR	61
Sent2Vec + LR	58
TF-IDF + LR	55

2) Sequential Models:

This type of model includes DL models like LSTM, RNN, BiLSTM, BiRNN, GRU, BiGRU. Which are specially designed for sequential data like text and time series data. To use this type of models, text encoding is done based on tokenization method. This models will have better accuracy then classical models. This models performs better and faster when hardware acceleration like GPU or TPU is provided. Here is the results of sequential models training:

Model	**F1_Score**
Sent2Vec + BiGRU	59
Doc2Vec + BiGRU	56
TF Tokenizer + LSTM	55

3) Transformers:

This models are advanced in terms of accuracy and performance as compared to previous models. This models are trained on large datasets and high performance systems with optimization. So definitly this models have high accuracy on ILDC data. For text encoding this models use own text encoding methods like BERT has BERTTokenizer, RoBERTa has RoBERTaTokenizer etc. This models performs better and faster when hardware acceleration like GPU or TPU is provided. Here is the results of transformers models:

Model	**F1_Score**
BERT	69
RoBERTa	73
XLNet	70

4) Hierarchical models:

In this types of model, transformers and sequential models are stacked. This models use text encoding methods from transformers. This models are the most accuracy models. But takes hours to complete. Hardware acceleration is necessary else memory limit will exceed. In this project this models are not tried, will be used in next version.

6. Recommendation System

In the previous section model is trained and best model is picked for further process. Second goal of the project CJPR is recommendation of similar court cases petition filed in the history if model predicted that petition is accepted. This will help judges to take decision quickly since most of the cases are in repetitive manner. Finding similar cases manually is very tough since India has millions of court cases completed. Also developing traditional algorithms using search methods are also far away from reality. So a ML based recommendation system could be a better approach.

For developing recommendation, text is need to be in numerical format. Means text should be encoded using any method. In project CJPR, TF-IDF [6] text encoding method is used. TF-IDF stands for Term Frequency Inverse Term Frequency. Now, petition that is predicted to be accepted by the model, will be also encoded using same TF-TDF method. There is method named *Cosine Similarity* [7], used to find similarity between two vectors. Cosine Similarity will be provided two arguments one is encoded petition and second in which similarity have to be find. Then cosine value will be given by the Cosine Similarity method, higher the cosine value implies lower the distance between two vectors implies higher the similarity. Then similarity between petition and each historical cases will be sorted and top petition will be selected.

Similar petition will be saved in a particular folder in .txt format. Judge can read it and can find relevant information from that.

7. Dockerizing

Once the model for judgment prediction and recommended system for recommending top similar court cases is prepared, deployment is necessary so user can use in CJPR task. Multiple methods are available for deployment. Most simple method is to create a flask website. Deployment on cloud services like GCP, AWS or Heroku can also be done. But here Docker deployment is preferred.

During the saving of RoBERTa model (best one), some files will be generated. Those files will be used in docker deployment for loading model in this phase. For deployment on docker some files will required like Dockerfile, python code to run script, requirement.txt. After building the image, container can be run in interactive mode user need to specify the path of the petition file in which is in .txt format. If the model predicted that petition is accepted then similar 10 petitions will be saved in recommended_petitions folder in same directory. And if model predicted the petition is not accepted then no need to show recommended petitions.

8. Ethical Concerns

The use of AI in the judiciary raises several ethical concerns. Here are some of them:

Bias and Discrimination: AI systems are only as unbiased as the data they are trained on. If the data used to train an AI system is biased or contains discriminatory elements, the AI system can replicate and amplify these biases. This can lead to unfair and unjust decisions. Therefore, it is essential to ensure that the data used to train AI systems in the judiciary is unbiased and that the algorithms used are transparent.

Lack of Accountability: Unlike human judges, AI systems cannot be held accountable for their decisions. This lack of accountability raises concerns about transparency and due process. There needs to be a mechanism in place to ensure that the AI system's decision-making process is transparent, and its decisions can be reviewed and challenged if necessary.

Privacy: The use of AI systems in the judiciary requires access to vast amounts of personal data, including sensitive information such as criminal records and medical history. There is a risk that this data could be misused or mishandled, leading to violations of individuals' privacy.

Limited Human Input: AI systems are only as good as the data they are trained on. Without proper human oversight and input, AI systems can make incorrect or biased decisions. Therefore, it is crucial to ensure that human

judges and legal experts are involved in the development and deployment of AI systems in the judiciary.

Fairness and Equity: The use of AI systems in the judiciary raises concerns about fairness and equity. For example, an AI system that relies on data from past convictions may perpetuate existing inequalities in the justice system. Therefore, it is essential to ensure that AI systems are designed to promote fairness and equity in the justice system.

Contrast with section 96 code of civil procedure: The right to appeal is an essential part of the Indian judicial system. It is a fundamental right that allows individuals to challenge a court's decision if they believe it is unjust or incorrect. The right to appeal is enshrined in the Indian Constitution and is available to all citizens, regardless of their social or economic status. In India, the right to appeal is provided for under various laws, including the Code of Civil Procedure, the Code of Criminal Procedure, and the Constitution of India. The right to appeal is available at different levels of the judicial system, depending on the nature of the case. For civil cases, the right to appeal is available to the High Court and Supreme Court. In criminal cases, the right to appeal is available to the District Court, Sessions Court, High Court, and Supreme Court. The appellate court has the power to review the lower court's decision and either confirm, modify, or reverse it. In short it is not accepted that any petitions is examined by AI system and it is accepted or rejected. There must be a human interrogator to examine. That's why *the CJPR is a helping system not a replacement system in judiciary*.

In conclusion, the ethical concerns surrounding AI in the judiciary are complex and multifaceted. It is important to address these concerns proactively to ensure that the use of AI in the judiciary promotes fairness, transparency, and accountability.

9. Future works

First here is the technical issues of CJPR:

Less Accuracy: Till now accuracy up to 72% is achieved. It should be high in this type of legal work. So there is need to explore more models, more specific, more reliable. Next version may include better accuracy models.

Task Specific Models: Models that are being used in CJPR are designed for language task like question-answering, language translation, text classification. These models are not designed for legal documents task. As legal documents are more sophisticated documents. There is need to be identify which clauses are important for judgment. So next version may include explanation or identification of those clauses which are used for judgment. This will drastically increase performance and also will help judges to identify which clauses are used for petition. Many organizations are working on that. Hugging Face is also working on a transformers models specially designed for legal documents [9].

Problem with Text Encoding Methods: In classical and sequential models the text encoding methods like TF-IDF has many problems like:

1) Large Vocabulary: Legal documents often use complex language and terminology, which can result in a large vocabulary of words that need to be considered when using TF-IDF. This can lead to high computational costs and require a significant amount of memory and processing power.

2) Limited Context: In legal documents, the context in which words appear is critical to understanding their meaning. However, TF-IDF only considers the frequency of a word in a document and does not account for its context. This can result in incorrect or misleading interpretations of the document's meaning.

3) Bias and Variability: Legal documents can be subject to bias and variability, which can affect the effectiveness of TF-IDF. For example, the same legal concept may be expressed using different terminology or phrasing, making it challenging to identify and compare across documents.

4) Handling Synonyms: Legal documents often use synonyms to refer to the same concept, which can result in TF-IDF assigning different weights to words that have the same or similar meanings. This can lead to incorrect interpretations of the document's meaning and make it difficult to identify related concepts across documents.

5) Complex Sentences: Legal documents often use complex sentences with multiple clauses and qualifiers, which can make it challenging to identify the most important words and phrases using TF-IDF.

In conclusion, while TF-IDF is a useful technique for many applications, it has some potential problems when used on legal documents. Legal professionals need to be aware of these issues and use a combination of techniques, including human review and analysis, to ensure accurate and meaningful interpretations of legal documents.

10. Conclusion

In conclusion, the CJPR project is designed to assist individuals, especially judges, in predicting whether a petition filed in court will be accepted or rejected. By utilizing machine learning techniques, CJPR can analyze historical data to provide insights on the likelihood of a successful petition. Additionally, if the petition is accepted, CJPR can recommend similar court cases from history to help expedite the judgment process. Overall, the CJPR project has the potential to significantly improve the efficiency and accuracy of the judicial system, saving time and resources while ensuring fair and just outcomes.

11. List of References

[1] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478.

[2] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4317–4323, Florence, Italy. Association for Computational Linguistics.

[3] Malik, Vijit & Sanjay, Rishabh & Kumar Nigam, Shubham & Ghosh, Kripa & Guha, Shouvik & Bhattacharya, Arnab & Modi, Ashutosh. (2021). ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation.

[4] M. N. Moghadasi and Y. Zhuang, "Sent2Vec: A New Sentence Embedding Representation With Sentimental Semantic," 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 2020, pp. 4672-4680, doi: 10.1109/BigData50022.2020.9378337.

[5] Lau, J. & Baldwin, T.An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, , 78--86. arXiv:1607.05368 [cs.CL]

[6] Qaiser, Shahzad & Ali, Ramsha. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications. 181. 10.5120/ijca2018917395.

[7] sklearn.metrics.pairwise.cosine_similarity — scikit-learn 1.2.2 documentation..

[8] lias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.