Milestone 2-
COL-764 Project
Precedent Retreival of Legal Cases

Pratik, Prawar, Rohan Debbarma,
Indian Institute of Techonology, Delhi
cs5180415@iitd.ac.in, cs5180417@iitd.ac.in

Abstract

This report presents a detailed description of the task of "Precedent Retrieval of Legal Cases". It also tries to compile together some of the approaches that will be employed to approach the above stated problem along with some brief description of the same. It also present a detailed description of the datasets collected to evaluate the system.

1 Introduction

While framing judgements, the judges rely upon previous decisions which involve interpretation of a question of law on similar cases. Therefore, such previous cases or instances which can be taken as a rule while pronouncing judgements are known as **precedents or priors** in a legal system.

In this project, our goal is to identify the precedent cases that have been cited in the judgement of a current case, provided we have the current case in the form of a query. Since this involves ranking the collection of prior cases in order of their relevance to the current case (query), it can be adjudged as - **Ranking task** on which various Information Retrieval techniques for ranking documents can be applied.

1.1 What is a Precedent ?

In legal field, precedent signifies guidance or authority of past cases on current/future cases. Various legal experts/philosophers have put forward their definitions of precedent. According to Gray, ‘precedent covers everything said or done, which furnishes a rule for subsequent practice.’ Salmond opined that, ‘in a loose sense, it includes merely reported case law which may be cited and followed by courts.’

In a judgement, these precedents are reported and often cited by the judges, thus are effectively followed by the courts. In other words, when there is some settled rule of law established through precedents, it becomes the duty of the judge who pronounces future judgments to take guidance from it and follow it.

1.2 Importance of Precedents in Indian Context

The principle of binding precedents directly follow from the doctrine of ’Stare Decisis’. A stare decisis signifies to ’stand by the things decided’. This ensures consistency in the application of law. Thus, existing binding precdents from past cases are applied to present cases by analogy.

The Indian Legal System is heavily derived from British Common Law as a direct consequence of the 300 year colonial period in India during the British Raj. The doctrine of 'stare decisis' is fundamental to thee British Legal System and as a result, India also follows it with some modifications.

India has a three-tier hierarchy of courts with Supreme Court at the centre, High Courts in the states supervised by the Supreme Court, Set of sub-ordinate courts under each high court. In the case of India, each court is bound by the decisions/judgements of the higher court above it. Decisions of one high court is not binding on any other high court but have persuasive value. The decisions of the Supreme Court is binding on all lower courts including the High Courts. However, the Supreme Court is not bound by its own decisions.

In view of the above points, it is apparent that identifying precedents and studying them is critical for both judges as well as the lawyers who argue cases on behalf of their clients. Therefore, Information Retrieval related to Precedent Retrieval of a current/future case has a far-reaching significance since it will assist judges and lawyers in studying cases/instances in a structured and timely manner.

2 Problem Statement

Given a set of queries which are cases judged in the Supreme Court of India, the task is to identify the set of relevant prior or precedent cases of that case from a collection (directory) of prior cases already provided.

In other words, this task involves ranking the collection of prior cases for a given query case.

There are two variants of the problem-

1. With Citation Markers - In the FIRE-2017 dataset, the citation markers which indicate the region of citation for that query is provided without the actual reference to the prior case.

   This information regarding citations can be utilised to formulate better queries out of query documents.
2. Without Citation Markers - In the FIRE-2019 dataset, there is no such citation marker. We are only provided description of a case. This makes retrieval of the cases more challenging since we only have a description of the case.

The objective is to find suitable approaches for both the variants.

3 Description of Datasets

The datasets collected for the task are - 1) FIRE-2017 Information Retrieval from Legal Documents (IRLeD) 2) FIRE-2019 Artificial Intelligence for Legal Assistance (AILA 2019)

The datasets could be accessed through the following drive link- Link to the dataset

3.1 FIRE-2017 IRLeD Dataset

This dataset consists of judgements of the Supreme Court of India parsed from HTML files from the website LIIofIndia.

Here is a detailed description of the dataset. The dataset contains the following directories and files:

1. Current_ Cases - This directory contains 200 query case documents (judgments given after year 2000) for which the prior cases are to be retrieved.
2. Prior_Cases - This directory contains a set of 2000 prior cases (judgments given before year 2000). The list of priors for each prior case has to be evaluated from this list.
3. irled-qrel - It contains relevance judgments for each of the query cases for evaluation of performance.

Note, the actual citations in each query document has been removed and replaced with a marker $[?CITATION?]$. This marker is provided to locate the actual position of the citations in a document.

3.2 FIRE-2019 AILA Dataset

This dataset consists of the following directories and files:

1. Object_casedocs - This directory contains 2914 Prior Cases which are relevant to the given queries.
2. Query_doc.txt- This file contains 50 queries which describe a particular case/situation. The format of the line in this fiile - QueryId || <QueryText>.
3. relevance_judgments_priorcases.txt - This file contains the relevant prior cases for each query. Its format is Format : <query-id> Q0 <document-id> <relevance>. Here, the relevance is 0 (not relevant) or 1 (relevant) based upon the prior case for a particular query.

Note- The difference between the FIRE-2017 and FIRE-2019 datasets is the existence of citation markers in FIRE-2017 datasets. Since no such citation markers are present in FIRE-2019 track, employing a citation specific area search in the document is not possible whereas such a method could be utilised in case of the first dataset

4 Approaches Proposed

Some of the methods that may be used to approach this task are as follows-

4.1 With Citation Marker

4.1.1 Citation Context + BM25

In this method, we would try to experiment with extracting text from the paragraphs and regions of text associated with a citation marker, selecting different chunks as the query and then processing it.

Corresponding to each citation in a particular query, we would obtain the text query to work with. This text query would be then fed to various variants proposed below-

1. Citation Context + BM-25- The text query obtained could be direclty fed to a BM-25 model. We can also experiment with different variants of BM-25. We will then obtain the top 1000 documents for each citation with some score. Taking the union of these documents for all the citations, we could then get the top-1000 documents for the query
2. Citation Context + IDF + BM-25 - In contrast to the previous item, instead of selecting the whole text query, we can select top50$\%$ words sorted by IDF score as the query and then apply BM-25. We can also vary the $\%$ of words to select the optimum.
3. Citation Context + Language Modeling + BM-25 - Instead of using IDF scores to select the query, we can experiment with Langauge models to select words for the query and then feed it to BM-25 model.

4.1.2 Dirichlet Prior Smoothing

In this method, the current cases could be treated as queries and the prior cases as documents.

Stopword Removal + POS Tagging + Stemming could be experimented with to obtain the query words from a query case.

This could then be ranked with tuning of Dirichlet Prior Smoothing based Language model discussed in class.

4.1.3 Vector Space Model using TF-IDF

In this method, stemming + POS tagging could be experimented with to obtain the text for a query case. TF-IDF scores could then be generated for each prior case and query case.

Metrics like Cosine Similarity could then be used to re-rank the documents.

4.1.4 Latent Dirichlet Allocation + Doc2Vec

In this variant, we can utilise a combination or weighted average of LDA + TF-IDF. LDA is applied to obtain matching topic words for prior cases and query cases. The simlarity metric is based upon the total number of matching words between a prior and a query. Doc2Vec algorithm could also be used to generate similarity scores based on cosine similarity. A weighted average of both the methods could then be used to rank the documents.

4.1.5 Wordnet + Sysnet

Using the paragraphs around citation marker as the query, Wordnet Lemmatizer could be used to generate the query as well as the prior cases.

These tokens after lemmatization could then be converted into sysnets i.e. cluster of cognitive synonyms. Thus, for each query and prior case, we have a sysnet representation,

We could then experiment with various similarity measures between them like Wu-Palmer Similarity metric, Leacock-Chodorow metric, Jiang-Conrath similarity etc.

4.2 Without Citation Marker

4.2.1 IDF + BM25

As there are no citation markers in this case, we can create an index corresponding to the documents in the dataset and then extract keywords using IDF scores. BM25 can then be used to obtain search scores.

To check for further improvements, we can also perform a weighted re-ordering of the results obtained in the above case to see if the performance improved or not.

4.2.2 TF-IDF+Word2Vec

In this technique, we can use extract the top term and case documents based on the TF-IDF scores for a query. Now, we can use word2vec to represent the vectors obtained as a result. Now, for ranking the documents, we can use the general euclidian distance function between the vectors.

4.2.3 Language Model + textRank

In this method, we can use textRank to extract keywords from queries, followed by Language Model to retrieve the keywords.

Another variant of this can be to use TF-IDF in place of textRank and BM25 or vector space retrieval in place of Language model which might improve the performance.

5 References

1. Mandal, A., Kripabandhu Ghosh, Arnab Bhattacharya, Arindam Pal and Saptarshi Ghosh. "Overview of the FIRE 2017 IRLeD Track: Information Retrieval from Legal Documents." FIRE (2017).
2. Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. FIRE 2019 AILA Track: Artificial

Intelligence for Legal Assistance. In Proceedings of the 11th Forum for Information Retrieval Evaluation (FIRE '19). Association for Computing Machinery, New York, NY, USA, 4–6.

3. Padigi, Sai Vishwas and Mayank, Mohit and Subramanyam, Natarajan. (2019). Precedent Case Retrieval using Wordnet and Deep Recurrent Neural Networks