Legal Document Retrieval

Juhi Pandey
juhi18393@iiitd.ac.in
Indraprastha Institute of Information Technology
New Delhi, Delhi, India

Yashraj
yashraj18422@iiitd.ac.in
Indraprastha Institute of Information Technology
New Delhi, Delhi, India

Punit Singh
punits@iiitd.ac.in
Indraprastha Institute of Information Technology
New Delhi, Delhi, India

Ankit Rana
ankit18381@iiitd.ac.in
Indraprastha Institute of Information Technology
New Delhi, Delhi, India

Prince Yadav
prince18037@iiitd.ac.in
Indraprastha Institute of Information Technology
New Delhi, Delhi, India

Ritik Malik
ritik18406@iiitd.ac.in
Indraprastha Institute of Information Technology
New Delhi, Delhi, India

KEYWORDS

Information Retrieval, Law, Legal Documents Retrieval System, POS tagging, Keyword Detection

ACM Reference Format:
Juhi Pandey, Yashraj, Punit Singh, Ankit Rana, Prince Yadav, and Ritik Malik. 2022. Legal Document Retrieval. In *Proceedings of Feb 26, 2022 (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 ABSTRACT

With the increase in crime rate, it becomes essential to have a fast justice delivery system. To reach a final verdict a legal practitioner has to go through a lot of related previous judgments for research and which is a time-consuming process. To reduce the search time for finding the right document we have proposed a solution that will find similar/relevant law cases that were registered earlier with their adjudication in the common law system based on the input query document of the case. We have achieved this by performing Precedence Extraction to extract relevant or cite-able documents from our dataset of prior cases and find the relevant case documents as per the input query using TF-IDF and Natural Language Processing based models.

2 INTRODUCTION

Precedents or previously closed cases are of importance to lawyers to maintain consistency in their judgements and ascertain that similar cases aren't treated differently due to biases.
For every new case, the concerned lawyer has to go through prior relevant cases and examine how the ongoing case was dealt with in the past. Courts and law firms have a colossal repository of such documents and going through each document in detail and extracting the useful information can be really a tedious and time-consuming task. In today's world, this problem can be automated. We have a dataset of lakhs of prior cases and their final verdict. Based on the input query i.e. ongoing case we can design a retrieval system that aids lawyers in their work by providing them with a list of cases that have been closed previously (a.k.a precedents), that are relevant to their current case.
We aim to achieve this using models and similarity scores like TF-IDF, Doc2Vec, Cosine, Jaccard, and ML/NLP models.

3 MOTIVATION

In recent times, one of the hard lines of work to automate have been social sciences. Law in particular is affected by this considering the legal liabilities. However, due to the wealth of textual data available in the form of transcripts and records, we can attempt to put Natural Language Processing, Data Mining, Machine Learning and Information Retrieval to good use. The dataset we are using is the FIRE 2017 IRLeD Dataset that consists of Indian Supreme Court decisions.

4 LITERATURE REVIEW

(1) **Mohamed H Haggag. 2013**
Keyword extraction using semantic analysis International Journal of Computer Applications 61, 1 (2013).
Keyword extraction is a very important task while retrieving information. The extracted words should correctly represent the content of the text. This paper proposes an extraction model based upon semantic similarities using word-to-word and word-to-whole semantic relatedness through term frequency analysis. The algorithm runs recursively till a saturation state which provides enhanced results compared to traditional methods. [1]

(2) **Tran, Vu. (2018)**
Automatic Catchphrase Extraction from Legal Case Documents via Scoring using Deep Neural Networks
This paper proposes an approach to generate important keywords particularly from legal case texts. The methodology involves predefined boundaries to select the phrases and subsequently, the phrase are scored and ranked using

Conference acronym 'XX, New Delhi, India,

ACM ISBN 978-1-4503-XXXX-X/18/06…$00.00
https://doi.org/XXXXXXX.XXXXXXX

deep learning models. The phrase with top scores are finally selected.[2]

(3) **Yang Liu and Mirella Lapata 2019**
Institute for Language, Cognition and Computation School of Informatics, University of Edinburgh USA.
This article proposes an interesting method to summarise a document using various encodings of BERT. The model uses Deep learning methodology and the methods of Trigram blocking and MMR are used to reduce the redundancy in the selected texts.The bert based model gives better ROGUE results than the other extractive and abstractive methods.[3]

(4) **Chin-Yew Lin 2004**
Rouge: A package for automatic evaluation of summaries
Proceedings of the ACL-04 workshop, Vol. 8. Barcelona, Spain.
To get the context of the test, better summaries are required. This paper proposes methods to evaluate the quality of the summaries when compared to (ideal) human written summaries of the texts. The methods give high correlation while evaluating single document summaries.[4]

(5) **Jose Aguilar, Camilo Salazar, Henry Velasco, Julian Monsalve-Pulido and Edwin Montoya 2**
Comparison and Evaluation of Different Methods for the Feature Extraction from Educational Contents
This article talks about various techniques like BM25, LSA, Doc2Vec and LDA for the feature extraction from the text documents. The quality of the data features determines the results of the retrieval model. The article proposes that LSA and BM25 performs better for the recommendation models. [5]

(6) **Omid Shahmirzadi, Adam Lugowski, Kenneth Younge. 2018, Patent Research Foundation, Seattle, USA**
Text Similarity in Vector Space Models: A Comparative Study
In this paper, we evaluated the performance of text vectorization methods for the real-world application of automatic measurement of patent-to-patent similarity. For the dataset used in the research, simpleTFIDF gives best performance given the cost, than more complex methods which require extensive tuning.[6]

(7) **Lisna Zahrotun. 2016 Department of Informatics Engineering, Universitas Ahmad Dahlan**
Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method
This paper compares various approaches to compare the similarity between two texts. It uses the method of document clustering for classification. The paper concludes with observations that cosine similarity gives best results even over the combination of cosine and jaccard similarity techniques.[7]

5 METHODOLOGY

We followed many approaches to find the best model.

(1) We performed pre-processing on the data to remove the irrelevant words i.e. stop words, remove punctuations, numbers and special characters, etc. from the text and perform tokenization and lemmatization on the dataset. To get a better idea of data, we found the frequency count of unique relevant words. For this we can set a threshold frequency and if a word count passes that threshold it is classified as one of the keywords of interest
(2) We pre-processed our query database similarly.
(3) Now we found and ranked the prior cases in our database that are relevant to the input set of documents.
(4) For this we used TF-IDF, Doc2Vec, Parsimionious Language Model, and BM25 models.
   (a) TF-IDF calculates the score based on the product of term frequency and inverse document frequency. TF is taken as the frequency of the term as opposed to the binary, raw count or normalised frequency.
   (b) Doc2Vec computes a feature vector for every document in the corpus.
   (c) BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document.
   (d) Parsimonious language models parse the documents using a greedy algorithm, and these models are good at picking up the terms that distinguish a text document from other documents in a collection. This was used similar to TF.
   (e) The remaining models are mixture models of 2 models to avoid bias and produce reproducible results.
(5) We calculate relevance based on various factors such as:
   - The frequency of the word in document. (TF)
   - IDF score of the word which signifies the importance of word in the document and corpus.
   - Parsimonious language model smoothing score
   - Doc2Vec similarity scores
(6) Based on these scores, we can rank the prior cases and test our models.

6 EVALUATION

6.1 TF-IDF Model Results

As mentioned in the following table, we can observe the recall to be increasing due to more documents being included as the value of k increases. Similar with Mean Reciprocal Rank value but that plateaus around k=5.
Precision however decreases as we increase k since we only have 5 relevant prior cases for every current case and as we add more and more irrelevant documents the score decreases.
Discounted Cumulative Gain can't be calculated since the relevant prior cases (true labels) are not in order of relevance. MAP score comes out to be 0.3597 for k = 5.

k	Recall @k	Precision @k	Mean Reciprocal Rank @k
1	0.092	0.460	0.460
2	0.157	0.392	0.520
3	0.211	0.352	0.545
4	0.251	0.314	0.562
5	0.281	0.281	0.568
6	0.307	0.256	0.572
7	0.330	0.236	0.575
8	0.347	0.217	0.576
9	0.359	0.199	0.577
10	0.370	0.185	0.578
11	0.379	0.172	0.579
12	0.396	0.165	0.5793
13	0.405	0.156	0.5797
14	0.420	0.150	0.5807
15	0.423	0.141	0.5807

6.2 Comparison with other approaches

Model	Recall @5	Recall @10
TF-IDF	0.281	0.370
Doc2Vec	0.168	0.256
Doc2Vec + TF-IDF	0.266	0.363
ParsimoniousLM	0.189	0.286
ParsimoniousLM + TF-IDF	0.209	0.308
BM25	0.236	0.309

Model	Precision @5	Precision @10
TF-IDF	0.281	0.185
Doc2Vec	0.168	0.128
Doc2Vec + TF-IDF	0.266	0.1815
ParsimoniousLM	0.189	0.143
ParsimoniousLM + TF-IDF	0.209	0.154
BM25	0.236	0.1545

Model	MRR @5	MRR @10
TF-IDF	0.568	0.578
Doc2Vec	0.381	0.395
Doc2Vec + TF-IDF	0.557	0.566
ParsimoniousLM	0.388	0.404
ParsimoniousLM + TF-IDF	0.415	0.430
BM25	0.472	0.487

6.3 How does the system perform on new data

Our model analyses prior cases to suggest as precedents to the current cases given as input. In case of new prior case data, the model needs to be retrained on the new dataset before being used. TF-IDF will still be the preferred model since it gives substantially better results so it can be inferred that it'll either outperform or match the other models' metrics with additional data especially since more data will only tend to make our suggestions better.
In case of querying cases with a file that doesn't match any prior cases, its score is zero and is not reflected or shown in the search results of our website.

6.4 Analysis of results

Pre-processing steps as mentioned in methodology proved to be the best combination of the individual components. TF-IDF performed the best and a combination of Doc2Vec and TF-IDF was a close second. The combination might be useful for different datasets since one model can be biased and useful for certain datasets but combination models offer to reduce this bias and have reproducible results. Since court documents are succinct and therefore do not have lots of repeating words or very inconsistent document sizes, TF-IDF with TF as the term frequency (and not binary, raw count or normalised counts) works best. Albeit a simple approach, it is a useful one for documents of similar size and non redundant words. Adding a combination model like Doc2Vec to TF-IDF only enhances the model further.

7 CONCLUSION

In our work, we compared multiple models for document retrieval and found TF-IDF and TF-IDF + Doc2Vec models to retrieve the most relevant documents from prior cases. Our final model of choice would still be the TF-IDF model. When comparing these results with the FIRE 2017 IRLeD Submissions, our model stands at around the fifth position.

8 FUTURE WORK

We would like to extend our model on the context citations, and smoothing and feature selection methods, and considering k-grams for better accuracy with catchphrases to improve accuracy in the future. We would also like to improve our tool by adding text matching and suggestions.

9 LIMITATIONS

- Only unigrams are considered in TF-IDF
- These models can only be specialised for a single judicial system as the judgements might differ for different laws pertaining to a country/state.

10 CITATIONS AND BIBLIOGRAPHIES

REFERENCES

[1] Mohamed H Haggag. (2013) *LaTeX: Keyword extraction using semantic analysis*, International Journal of Computer Applications.
[2] Tran, Vu.. (2018) *LaTeX: Automatic Catchphrase Extraction from Legal Case Documents via Scoring using Deep Neural Network.*
[3] Yang Liu and Mirella Lapata (2019) *LaTeX: Institute for Language, Cognition and Computation*, Institute for Language, Cognition and Computation.
[4] Chin-Yew Lin (2004) *LaTeX: Rouge: A package for automatic evaluation of summaries*, Proceedings of the ACL-04 workshop, Vol. 8. Barcelona, Spain.
[5] Jose Aguilar, Camilo Salazar, Henry Velasco, Julian Monsalve-Pulido and Edwin Montoya *LaTeX: Comparison and Evaluation of Different Methods for the Feature Extraction from Educational Contents.*
[6] Omid Shahmirzadi, Adam Lugowski, Kenneth Younge (2018) *LaTeX: Text Similarity in Vector Space Models: A Comparative Study*, Patent Research Foundation, Seattle, USA.
[7] Lisna Zahrotun (2016) *LaTeX: Comparison Jaccard similarity, Cosine Similarity and Combined Both of the Data Clustering With Shared Nearest Neighbor Method*, Department of Informatics Engi- neering, Universitas Ahmad Dahlan.