Efficient Data-Driven Methods for Legal Document Summarization, Similarity, and Role Labelling

Project-II (EC47004) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Electronics and Electrical Communication Engineering

by

Shruti Shreyasi

(19EC10086)

Under the supervision of

Professor Partha P. Chakrabarti

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2022-23

April 28, 2023

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: April 28, 2023 (Shruti Shreyasi)
Place: Kharagpur (19EC10086)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Efficient Data-Driven Methods for Legal Document Summarization, Similarity, and Role Labelling**" submitted by **Shruti Shreyasi** (Roll No. 19EC10086) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Electronics and Electrical Communication Engineering is a record of bona fide work carried out by her under my supervision and guidance during Spring Semester, 2022-23.

Date: April 28, 2023

Place: Kharagpur

Professor Partha P. Chakrabarti
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Abstract

Name of the student: **Shruti Shreyasi** Roll No: **19EC10086**

Degree for which submitted: **Bachelor of Technology**

Department: **Electronics and Electrical Communication Engineering**

Thesis title: **Efficient Data-Driven Methods for Legal Document Summarization, Similarity, and Role Labelling**

Thesis supervisor: **Professor Partha P. Chakrabarti**

Month and year of thesis submission: **April 28, 2023**

Summarizing and retrieving similar legal documents is a challenging task due to their large size and number. We demonstrate that improving the accuracy of sentence role labeling can enhance the quality of summaries. We have introduced a new summarization algorithm, the Knapsack summarizer, which outperforms existing extractive methods. We also introduce the Top-K Analysis metric to compare summarizer performance using similarity scores. Our next objective is to find similar documents in a legal corpus, for which existing methods use supervised approaches, but we explore semi-supervised and unsupervised methods. We show that similarity scores computed on summaries of legal documents are faster and more reliable than those computed on entire documents. Furthermore, we investigate the effect of replacing rhetorical roles with cluster labels of sentences obtained via clustering. In total, we have tried to establish triad relationships between summarization, similarity, and role labeling in the legal domain.

Contents

List of Figures

List of Tables

Abbreviations

FAC	**Fac**t
PRE	**Pre**cedent
STA	**Sta**tute
RPC	**R**uling by **P**resent **C**ourt
RLC	**R**uling by **L**ower **C**ourt
RC	**R**uling by **C**ourt
ARG	**Arg**ument
Ratio	**Ratio** of decision
AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
NLP	**N**atural **L**anguage **P**rocessing
LDA	**L**atent **D**irichlet **A**llocation

Chapter 1

Introduction

1.1 Background

The legal judiciary system of most countries, including the USA, UK, and especially India, has a lot of administrative work done in courts. Owing to the large number of judgements, it is really difficult to look for cases similar to a particular case or perform exploratory analysis on the documents.

This quite often leads to a lot of judicial time being wasted for work which can be handled much faster by the use of state-of-the-art technology in the domain of AI/ML. Legal experts as well as both of the concerned parties might find it difficult to find cases relevant to a particular case. Owing to the large size of legal documents, it can be very difficult to summarize them and find relevant points. This, in turn, can cause inefficiencies in the court hearings, a substantial reduction in the pendency of court cases, an equitable distribution of case hearings of all types of cases, and a lot of time being saved by the parties and legal representatives involved in them. Our research focuses on the legal and judicial systems of India. The main focus is on the triad of legal rhetorical role labelling, summarization, and similarity ranking of legal documents.

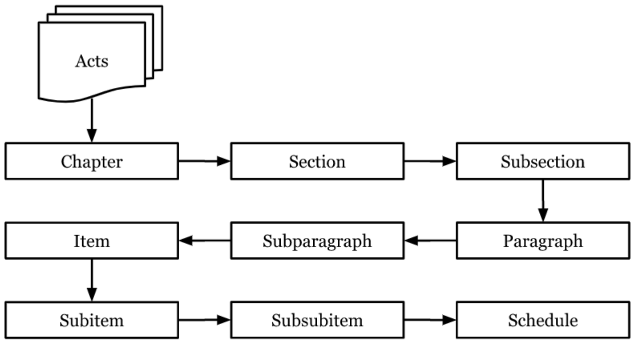


FIGURE 1.1: Hierarchy of acts

Indian Legal System

Indian law refers to the system of law which operates in India. It is largely based on the practices of English common law. Various Acts introduced by the British are still in effect in modified form today. Much of contemporary Indian law shows substantial European and American influence.

1.2 Objective of my project

The project's ultimate objective is to build a legal domain-specific search system capable of retrieving similar legal documents based on a query. The other important task is to predict and analyze case-related data for the prediction of case status, smart scheduling, and efficiency analytics. This project focuses on the first objective.

The target system should be able to summarize legal documents in multiple ways and prepare a basic, labelled, and structured summary. It should role label the documents as well as the summary. Similar documents are to be returned based on TF-IDF and cosine similarity as well as based on role-labelled legal documents and clustering-based and rule-based methods. Custom search for similar cases based on the headnote itself as well as the judgment and the summary prepared.

1.3 Summary of work done

The four main research questions addressed in this project are as follows:

Q1. Are there ways to use rule-based techniques to perform the task of role labelling?

A1. We explored a rule-based approach and a cluster-based labelling technique for performing role labelling of legal domain documents.

Q2. How can the system be made scalable?

A2. Legal language has a well-defined structure, and we employed rule-based methods for role labelling the documents, which can be easily scaled.

Q3. What is the importance of the triad of similarity, summarization, and role labelling?

A3. We concluded that better summaries lead to better similarity scores, and role labelling improves the similarity and summaries of legal documents. Additionally, role labelling can help in forming better summaries.

Q4. How can role labelling be employed to improve the quality of summaries?

A4. We have introduced a new summarization algorithm called the Knapsack summarizer, which incorporates cluster-based summaries and performs better. We also introduced a metric, Top-K analysis, to measure the quality of summaries.

1.4 Future scope

The future scope of research is as follows:

1) To explore the visual summary data structure in more detail and add more features.

2) To implement a dynamic similarity algorithm while forming a similar document retrieval list and perform an ablation study on the same.

3) To implement a structured case display based on a concept graph and retrieve the similarity of documents based on different clusters/roles.

4) To establish entailment relationships in the document flow concerning role labels.

5) To perform a quantitative ablation study on clustering and its effect on the Knapsack summarizer.

1.5 Layout of the report

The Introduction (Chapter 1) of the report outlines the Background for the project, the Objectives to be attained, a Summary of the work done and the Future scope of this project.

The Literature Review (Chapter 2) contains a summary of all the literature cited in the project for similarity, summarization and role labelling.

The Overview of the System (Chapter 3) dives deep into the system overview and explores the triad of Similarity, Summarization and Role labelling.

Initial Studies and research hypotheses (Chapter 4) explain the research goals and the initial studies performed on TF-IDF and Doc2Vec, as well as the functionality test on similarity on summaries of legal documents.

Ablation study of summary and similarity (Chapter 5) explains the method of comparison followed to compare various summarizers. It is then followed by the results obtained for various summarizers when used for similarity ranking.

Role labelling and unsupervised approaches (Chapter 6) introduce a clustering-based unsupervised approach for cluster-based sentence labelling. It qualitatively derives the optimum number of clusters.

Unsupervised Legal Summary (Chapter 7) introduces a new summarization algorithm: The Knapsack or Union summarizer.

Intrinsic evaluation of Knapsack summarizer (Chapter 8 evaluates the Knapsack summarizer on ROUGE score and Summarization - Similarity analysis.

Top - K Analysis Metric (Chapter 9) introduces the Top - K Analysis as a metric of evaluation of summaries.

Extrinsic Evaluation of Knapsack Summarizer (Chapter 10) evaluates the Knapsack summarizer on Top - K Analysis Metric and establishes coherence of the metric based on correlation and scaling.

Ablation study of extractive summarizers (Chapter 11) performs a comparative study of Knapsack summarizer and other extractive summarizers based on the Top - K Analysis Metric.

Visual Summary (Chapter 12) focuses on the need of having a visual data structure for the representation of a legal document.

Fetching Similar Documents: An Algorithmic Approach (Chapter 13) introduces an algorithmic approach to retrieve similar documents for a query document without searching the entire corpus.

Triad relationships (Chapter 14) finally connect all the dots regarding the triad relationships and gives closure to this report.

Chapter 2

Literature Review

2.1 Summarizing legal documents

Wagh and Anand (2020) proposed the approach of concept-based similarity estimation among court judgments and used graph-based methods to identify the prominent concepts present in a judgment and to extract sentences representative of these concepts. The sentences and concepts extracted were then used to express/visualize the likeness among concepts between a pair of documents from different perspectives. The experimental results suggest that the proposed approach of concept-based similarity is effective in extracting relevant legal documents and performs better than other competing techniques.

They also proposed to aggregate the different levels of matching so obtained into one measure quantifying the level of similarity between a judgment pair and employed the ordered weighted average (OWA) family of aggregation operators for obtaining the similarity value. The proposed two-level abstraction of similarity enabled informative visualization for deeper insights into case relevance.

How this idea was incorporated:

A legal document can be considered a set of sentences

$$D = \{S_1, S_2, ...S_i, ..., S_K\}$$

and we have clustered each sentence into a total number of C clusters. The main concept covered by each cluster is called the topic of the cluster. We have tried to use these cluster labels for cluster role labelling (as corresponding to rhetorical role labelling) of the documents.

2.2 Similarity of legal documents

Pandey et al. (2021) proposed to increase the existing role labels to the concept of Case Analysis labels Legal Issues, Argument by Appellant, Argument by Respondent, Argument by Amicus Curiae, Relief Prayer, Observation Findings, Legal Principles, Fact, Rationale, Conclusion, Verdict, Interim Order and Compliance. They also applied rule-based approaches to label the same. Also, this work is centered around Air Pollution cases only, and it is a future task to examine how the rule-identification module performs across various categories of case types.

How the ideas were incorporated:

A legal document $\{S_1, S_2, ...S_i, ..., S_K\}$ have been clustered into a total number of C clusters representing cluster role labels. The value of C is varied to obtain a suitable cluster size. For the identification of rhetorical roles, rule-based techniques have been employed for standard role labels. They are suited to all types of cases in general. The idea of having case labels is extended to assigning multiple cluster labels or role labels to the sentences of the legal document. It is assumed that each sentence falls under the category 'Facts'. Then, the rest of the rules are applied.

Bhattacharya et al. (2020) proposed to utilize the complementary knowledge to text-based methods using document embedding and to apply graph embedding techniques Node2Vec and Metapath2Vec over Hier-SPCNet. They have used the cosine similarity between node embeddings for computing final similarity scores. They have concluded that Pearson correlation coefficient (ρ) on Network-based methods (Metapath2vec on Hier-SPCNet: 0.674) is lesser as opposed to text-based methods (Doc2Vec: 0.734)

How the ideas were incorporated:
We have incorporated the idea by using text-based methods as a baseline and considering both text-based data and cluster information while computing similarity scores of legal documents.

Mandal et al. (2021) shows that Doc2Vec performs markedly well with full documents, paragraphs, and summaries. It does not perform well enough with the other form of text representations. BERT and other large models cannot be used as it does not perform well on long sequences of texts and, we have very little data, which is not enough for fine-tuning. It was seen that traditional vectorizers (such as TF-IDF and LDA) perform well in case of document similarity. It was seen that Doc2Vec and TF-IDF performed very well. To explore this approach, a preliminary comparative study was done between Doc2Vec and TF-IDF approaches in our project.

2.3 Role labelling

Bhattacharya et al. (2019a) proposed to assign one of the seven role labels to each sentence in the legal document Facts (FAC), Ruling by Lower Court (RLC), Argument (ARG), Statute (STA), Precedent (PRE), Ratio of the decision (Ratio), Ruling by Present Court (RPC) and they have incorporated Hierarchical BiLSTM CRF Classifier with handcrafted features in the CRF layer to identify rhetorical role labels. The same role labels are used in our project.

Chapter 3

Overview of the system

3.1 Legal search system

The ultimate objective of the project is to build a legal domain-specific search system which is capable of retrieving similar legal documents based on a query.

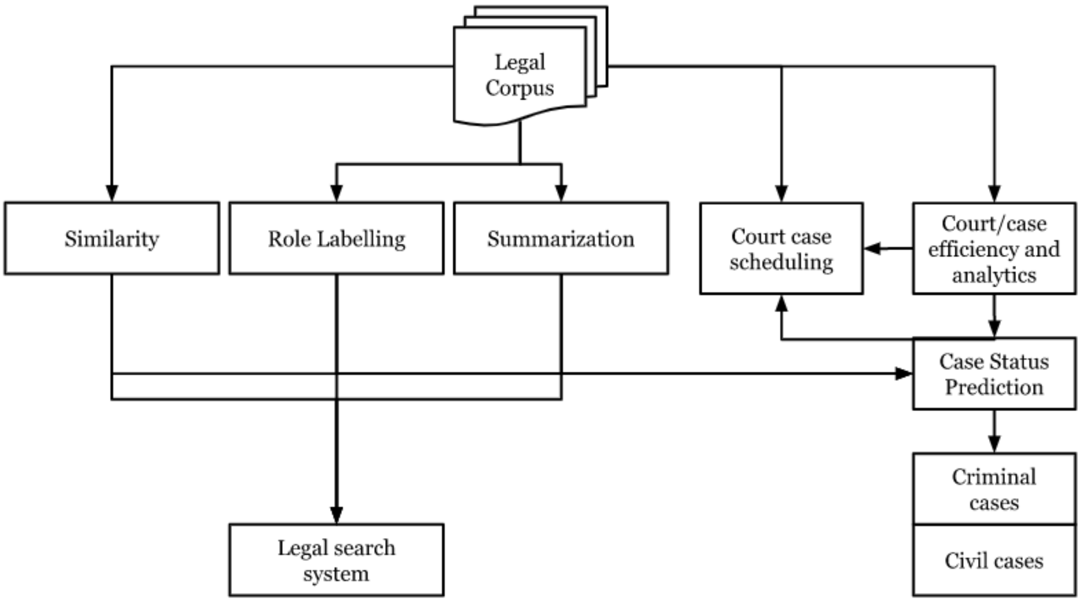


Figure 3.1: Overview of the system

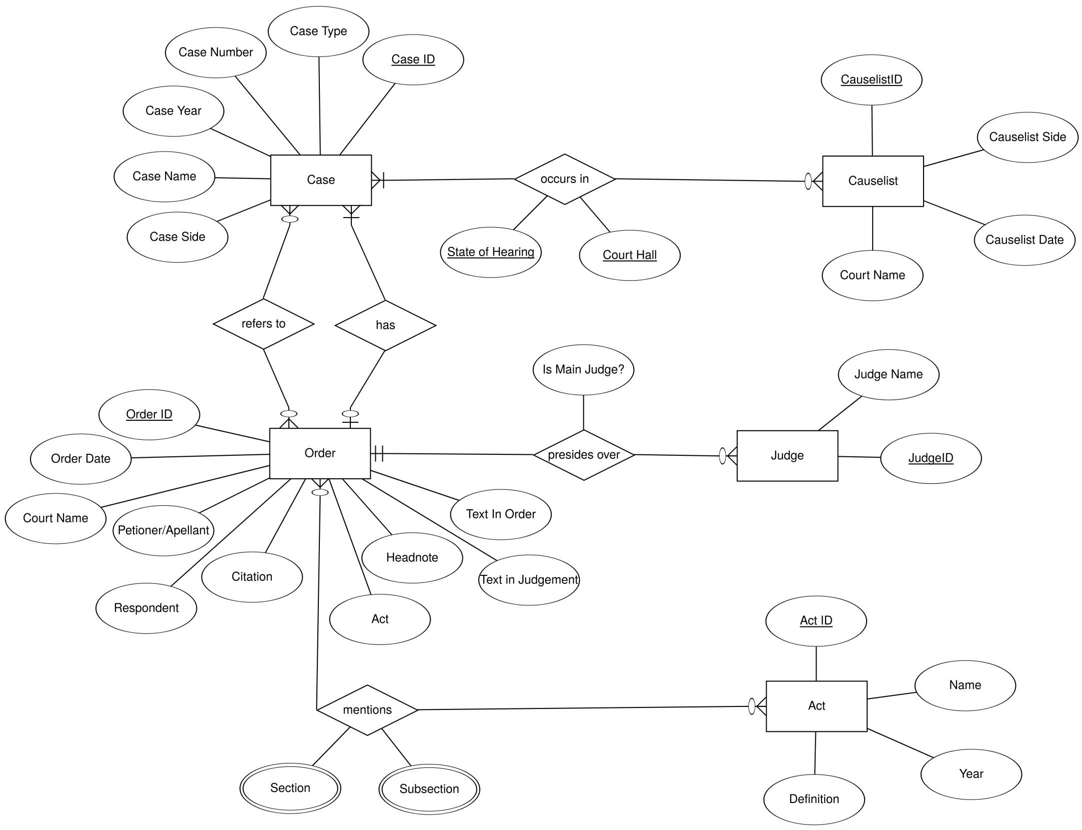


Figure 3.2: ER diagram

The other important task is to predict and analyse case-related data for prediction of case status, scheduling and efficiency analytics. This project focuses on the first objective. Figure 3.2 lists the data to be stored in the database for the legal documents.

3.2 Similarity - Role labelling - Summarization

In the legal domain, similarity, legal rhetorical role labelling and summarization form a very important triad.

The similarity of a legal document can be improved if we can produce better role labelling (Bhattacharya et al., 2021a). We hypothesise that better summaries also lead to better similarity scores that are more efficient to compute.

Role labelling can be improved if we can produce better similarity algorithms. In this project, a clustering-based algorithm has been explored. Some rule-based approaches have also been studied.

Summarization of legal domain documents with the aid of role labels has already been looked into (Bhattacharya et al., 2021a). Our project aims to summarize documents using similarity metrics found based on clustering and rule-based techniques. We have introduced a new summarization algorithm called the Knapsack summarizer that involves cluster-based role labels as well as a metric called the Top-K Analysis Metric to evaluate summarizers based on similarity.

3.3 System overview

The objective of the system is to provide various summarization methods for legal documents, generating both basic and structured summaries, along with role labelling for both the documents and summaries. The system should be capable of retrieving similar documents using techniques such as TF-IDF and cosine similarity, clustering, and rule-based methods on role-labelled legal documents. Additionally, the system should facilitate custom searches for similar cases using headnotes, judgements, and prepared summaries.

3.4 Research questions

The four main research questions are as follows:

1. Legal language is well-defined and we can make use of language/file structure. Are there ways to use rule-based techniques to perform the tasks of role labelling, similarity and summarization?

2. How can the system be made scalable? It is not possible to employ ML/DL-based models since they are difficult to scale and need large amounts of data to fine-tune

3. What is the importance of the triad of similarity, summarization and role labelling?

4. How can role labelling be employed to improve the quality of summaries?

Chapter 4

Initial studies and research hypothesis

4.1 Comparative study of TF-IDF and Doc2Vec

A comparative study was done between TF-IDF and Doc2Vec on a mini supreme court dataset of 15 documents.

Observations:

• TF-IDF performs better as compared to Doc2Vec on manual inspection of the documents

• TF-IDF was also performed on summaries of the documents and manual inspection, it was observed that the results were better

Hypothesis:

Similarity scores obtained on summaries are better and faster as compared to similarity scores computed on entire documents. This hypothesis can also be derived alternatively from the triad of similarity, summarization, and role labelling.

Table 4.1: Comparative study of Doc2Vec and TF-IDF

	TF-IDF	Doc2Vec modified	TF-IDF (summaries)
0	0 7 8 6 9 4 5 10 14 2	[7, 7, 7, 10, 7, 7, 4, 3, 7, 3]	0 8 14 9 2 7 12 11 6 13
1	1 14 4 11 10 12 5 7 9 3	[1, 4, 10, 13, 14, 4, 7, 13, 8, 3]	1 4 14 2 12 11 5 10 7 13
2	2 6 7 9 5 4 8 12 11 14	[7, 2, 3, 14, 7, 8, 3, 12, 7, 12]	2 1 9 7 4 14 11 0 6 8
3	3 9 5 7 11 12 4 6 1 14	[9, 7, 8, 5, 7, 3, 10, 10, 6, 7]	3 5 9 6 8 11 7 12 14 0
4	4 7 6 1 10 9 5 14 12 3	[9, 4, 2, 11, 4, 12, 15, 7, 7, 4]	4 1 10 2 7 14 12 9 11 6
5	5 7 6 9 4 3 12 14 2 1	[8, 10, 8, 7, 2, 6, 7, 7, 3, 10]	5 7 1 14 8 3 4 6 9 2
6	6 7 9 2 5 4 11 8 0 10	[10, 10, 7, 8, 7, 3, 1, 6, 8, 10]	6 9 8 11 3 2 10 4 5 7
7	7 6 9 5 4 2 0 8 14 10	[12, 3, 8, 8, 7, 7, 7, 9, 7, 8]	7 9 11 5 2 10 12 14 4 0
8	8 7 0 6 10 9 4 14 5 2	[3, 10, 7, 7, 7, 3, 10, 7, 7, 7]	8 0 9 10 6 5 14 2 3 11
9	9 6 7 5 2 3 4 8 12 14	[2, 5, 3, 8, 7, 3, 3, 4, 3, 4]	9 7 6 2 8 0 14 3 4 12
10	10 12 14 4 7 1 8 6 11 9	[7, 7, 9, 3, 10, 7, 3, 7, 3, 3]	10 14 4 11 7 1 8 12 6 2
11	11 14 1 7 6 3 10 4 5 9	[9, 7, 7, 10, 7, 10, 7, 3, 8, 3]	11 7 1 10 14 6 12 2 4 3
12	12 10 7 4 14 5 1 3 9 6	[9, 14, 10, 2, 4, 10, 15, 3, 3, 15]	12 1 14 7 4 11 13 10 0 9
13	13 14 5 7 9 12 10 4 1 11	[7, 7, 11, 12, 3, 3, 13, 3, 3, 7]	13 14 12 0 1 2 8 5 11 9
14	14 1 11 10 7 4 13 12 5 9	[10, 8, 10, 13, 8, 8, 3, 7, 10, 3]	14 1 10 13 0 12 2 11 5 7

Conclusion:

The similarity scores obtained using TF-IDF are better as opposed to modified Doc2Vec. The modified variant of Doc2Vec takes into account multiple sentences that may be similar to the query document. Thus, the document appearing more frequently as well as the document appearing first, both of them are relevant and the position in the rank list defines the relevance of the document.

4.2 Functionality test of similarity on summaries

Hypothesis:

Similarity scores obtained on summaries are better and faster as compared to similarity scores computed on entire documents.

A comparative study was done between similarity scores of entire documents and summaries of legal documents based on TF-IDF vectors on a supreme court dataset of 500 documents. The results are based on Precision at K = 5 for both cases.

Observations:

• On manual inspection of top 5 ranked similar documents for 6 documents, the average precision at 5 was labelled as 0.66/5 for entire documents whereas, it was 1.33/5 for summaries

• This confirms the results obtained on the 15-document mini-corpus

• Cosine similarity on TF-IDF vectors was also performed on summaries of the documents, and manual inspection, it was observed that the results were better

• Short summaries are misleading and need to be removed from the corpus

• This was the case where all the metadata was considered in the summary. In the case where only the main judgment was considered, summaries performed better. For instance, on the mini dataset taken above (15 docs of SC)

Conclusion:

Summaries do perform better for similarity as compared to complete documents. The mean precision at 5 was labelled as 0.66/5 for entire documents whereas, it was 1.33/5 for summaries.

$$Precision\,@\,5 = \frac{number\,of\,relevant\,documents}{5} \tag{4.1}$$

$$Mean\,Precision\,@\,5 = \frac{\Sigma Precision\,@\,5}{N} \tag{4.2}$$

TABLE 4.2: Top 5 Relevant documents for summaries & complete documents

TABLE 4.3: 1.txt

Result	Complete doc		Summary of doc	
	File	Relevance	File	Relevance
1	10619	0	10320	1
2	10676	0	10	1
3	10854	0	10619	0
4	10241	0	10748	1
5	10	1	10517	0

TABLE 4.4: 10.txt

Result	Complete doc		Summary of doc	
	File	Relevance	File	Relevance
1	10514	0	10320	0
2	10276	0	10357	0
3	10619	0	1	1
4	10424	0	10651	0
5	10625	0	10619	0

TABLE 4.5: 10004.txt

Result	Complete doc		Summary of doc	
	File	Relevance	File	Relevance
1	10427	0	10320	1
2	10323	0	10210	1
3	10527	0	10757	0
4	10321	1	10659	0
5	10325	0	10671	0

TABLE 4.6: 10108.txt

Result	Complete doc		Summary of doc	
	File	Relevance	File	Relevance
1	10285	1	10320	0
2	10836	0	10232	0
3	10607	0	10285	1
4	10803	0	10627	0
5	10438	0	10178	0

TABLE 4.7: 10114.txt

Result	Complete doc		Summary of doc	
	File	Relevance	File	Relevance
1	10827	0	10320	0
2	10341	0	10827	0
3	10183	0	10626	0
4	10505	1	10181	1
5	10626	0	10410	0

TABLE 4.8: 10154.txt

Result	Complete doc		Summary of doc	
	File	Relevance	File	Relevance
1	10437	0	10320	0
2	10335	0	10612	1
3	10582	0	10599	0
4	10323	0	10335	0
5	10321	0	10437	0

Chapter 5

Ablation study of summary and similarity

This part of the project concerns studying the relationships between summaries and similarity scores of a legal document. We aimed to test the earlier made hypothesis that summaries give better similarity results as compared to complete documents. To test that, a comparative study was performed with various summarizer algorithms in a series of experiments.

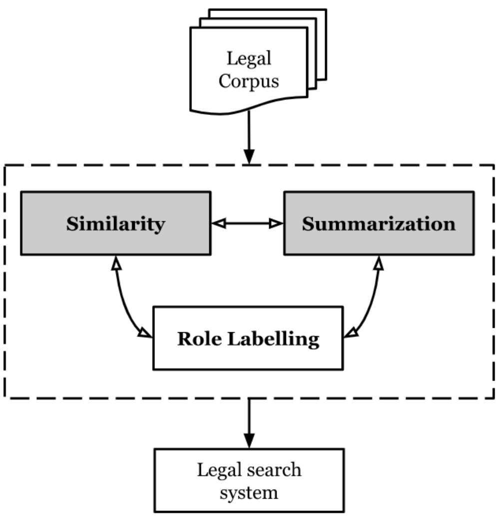


Figure 5.1: Exploring relationships between summary and similarity

The comparison was done between LSA and LEXRank summarizers and DelSumm from (Bhattacharya et al., 2021a). DelSumm is a summarizer based on rhetorical role labels. Weights are given to each role and that sets the ratio of that role in the final summary.

5.1 Method of comparison

A ranked list of retrieved similar documents is maintained for the complete document case as well as the summaries of the documents. In each case, top K documents are considered. Say the top K documents from the case of the complete documents are

$$L_d = \{d_{c1}, d_{c2}, ...d_{ci}, ..., d_{cK}\}$$

and for the summaries are

$$L_s = \{d_{s1}, d_{s2}, ...d_{sj}, ..., d_{sK}\}$$

respectively. Our goal is to find an appropriate K for the case of each summarizer so that the mean deviation between the lists L_d and L_s minimizes where we are considering deviation as the difference in ranks in the lists L_d and L_s for a document which is present in both the lists.

We are also trying to maximize the number of common documents in both the lists L_d and L_s.

5.2 Comparison of various summarizers

Corpus size:

500 documents of Supreme Court files

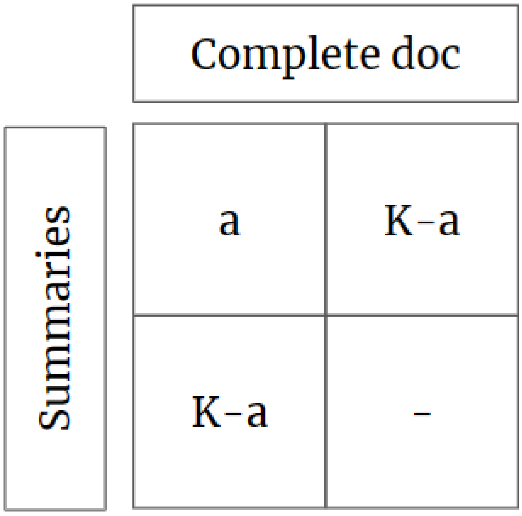


FIGURE 5.2: a/K documents are common in the ranking lists of L_d and L_s

- Summarizers used:

  1) `DELSumm`
  2) `LSA`
  3) `LexRank`

- Hypothesis: We can obtain similar documents from the summarized documents more efficiently (faster)

- Data:

1) For every document d in corpus D, we have an ordered list of similar documents for every document.

2) For every document d in corpus D, we have an ordered list of similar documents for every document summarized.

3) For every document, we have summary size as 10%, 15%, 20%, and 25% of the total length of the document.

4) Also, we have top k documents for comparison (K=10,20,50,100)

- Plots:

For every plot, the x-axis contains the k-value (top k docs) or proportion of the total documents of the corpus. y-axis contains the F1 scores or proportion of common documents in a full-doc similarity list as well as a summarized similarity list.

$$Precision = \frac{True\,Positive}{True\,Positive\,+\,False\,Positive} \tag{5.1}$$

$$Recall = \frac{True\,Positive}{True\,Positive\,+\,False\,Negative} \tag{5.2}$$

$$F1\,score = \frac{2\,*\,Precision\,*\,Recall}{Precision\,+\,Recall} \tag{5.3}$$

TABLE 5.1: Number of documents common in top-k documents [DELSumm]

	10%	15%	20%	25%
k=10	1	1	12	30
k=20	0	2	8	31
k=50	1	4	10	36
k=100	1	4	14	36

TABLE 5.2: Number of documents common in top-k documents [LSA]

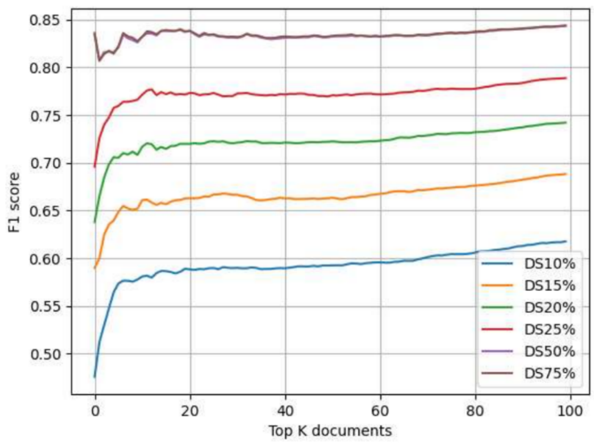
	10%	15%	20%	25%
k=10	1	6	18	44
k=20	1	5	17	48
k=50	3	6	19	58
k=100	3	8	24	62

TABLE 5.3: Number of documents common in top-k documents [LexRank]

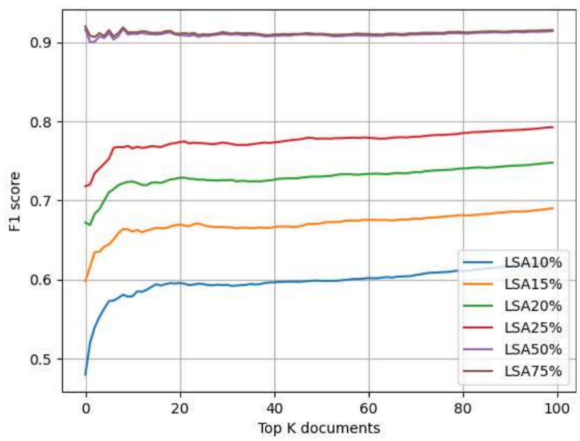
	10%	15%	20%	25%
k=10	2	4	18	52
k=20	2	9	23	54
k=50	5	11	31	59
k=100	7	14	35	70

- Observations:

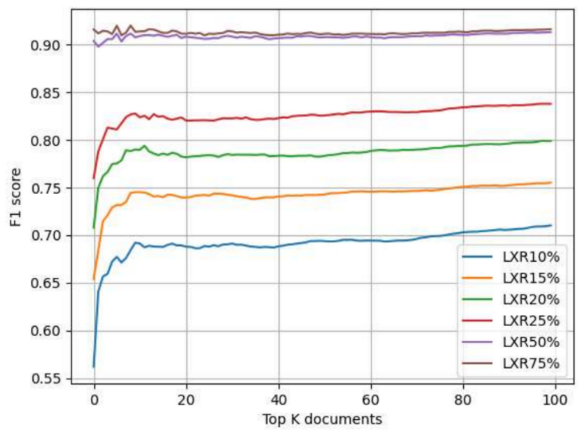
a) LexRank outperforms LSA and DELSumm in all scenarios.



(a) DELSumm

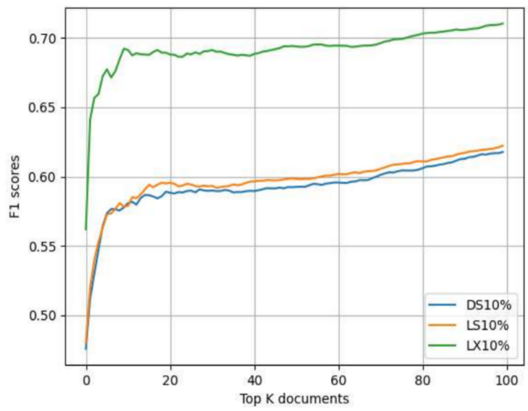


(b) LSA

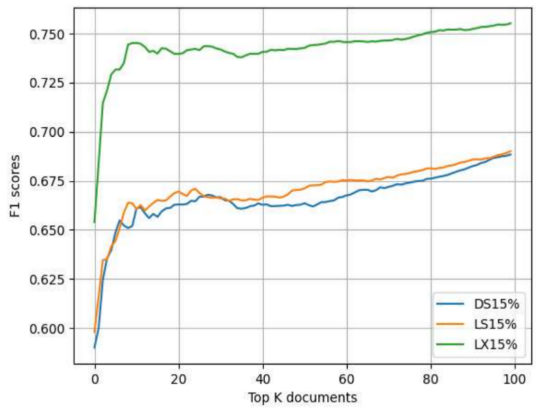


(c) LexRank

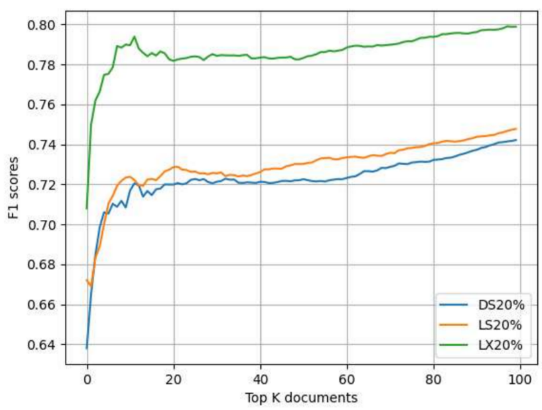
FIGURE 5.3: F1 score of summarized docs with variable % of summary



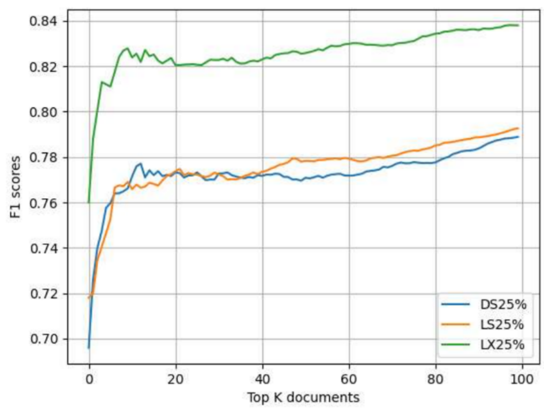
(a) 10%



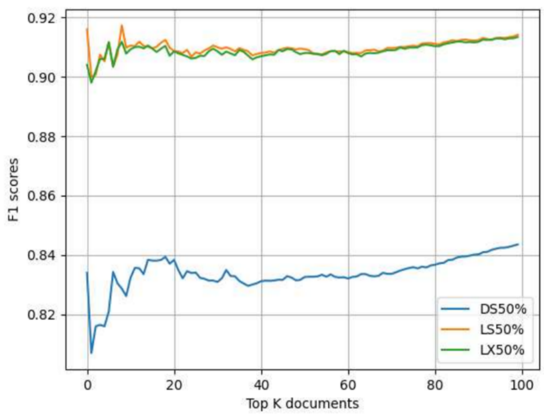
(b) 15%



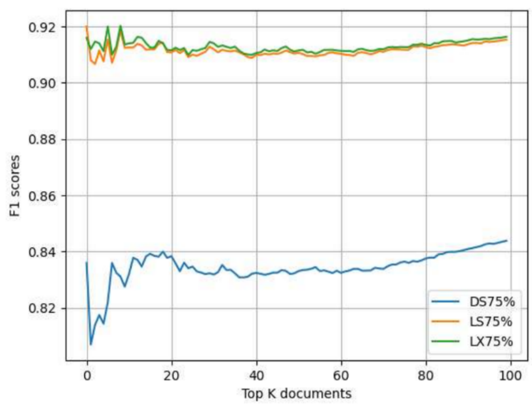
(c) 20%



(d) 25%



(e) 50%



(f) 75%

Figure 5.4: F1 score of summarized docs with variable % of summary

b) LSA and LexRank both reach their highest value and grow slowly whereas DELSumm plot rises steadily over k.

c) Delsumm and LSA have similar results. However DELSumm takes lesser number of words as compared to LSA in our case.

Chapter 6

Mixed approach of role labelling and unsupervised approaches

Role labelling legal documents is a very challenging task. We have explored two methods of sentence labelling of a legal document. First is a rule-based approach where we have tried to explore the syntactic aspect of law code. Next, we have experimented with a clustering-based unsupervised approach for cluster-based sentence labelling.

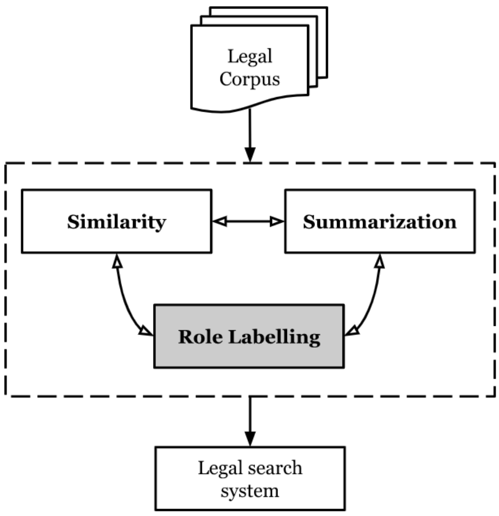


Figure 6.1: Exploring Role labels of a legal document

6.1 Rule Based Approach

We have considered the seven rhetorical roles: FAC (Facts), ARG (Arguments), PRE (Precedent), STA (Statute), Ratio (Ratio of decision), RLC (Ruling by lower court) and RPC (Ruling by present court) as used in (Bhattacharya et al., 2019b). We have clubbed the roles: RLC and RPC into one role: Ruling by the court (RC) and have extracted all the STA roles using a RegEX matching approach. The RegEx snippet for the same is as follows:

```
((sub(-||\s)section|section(s)*|sec\.*|\ss\.*|ss\.*)\s*[0-9()]+\s*([0-
9()\,\.a-z]*\s*(,|and|&)\s*)*(\(?[a-z]?\)?\(?[a-z]?\)?\s+)*\s*\-*\s*
(of)*\s*(the)*\s*(act\s|((?!act\s).)*\sact[^a-z])*\.*\,*\s*[0-9()]*)|
((art\.|article|articles)\s[0-9()a-z,]*\s*and\s*[0-9()a-z]*)|(((item|
items)\s[0-9(),\sa-z]*)*(\sof\s)*(sch\.\s*[()0-9a-z]*))
```

We have used the RegEx of the forms:

```
<section> of the <act>
<item> of the <schedule>
```

as the two basic constructs. Next, we explored the distribution of role labels to look for patterns. Some of the very frequently occurring patterns and observations were:

- FAC are mostly present at the beginning of the documents
- PRE -> Ratio/RC
- ARG -> Ratio -> ARG -> PRE -> Ratio
- Ratio -> RC
- FAC -> (RC -> FAC)* -> Ratio/ARG -> RC
- FAC -> (RC -> FAC)* -> (Ratio/ARG -> PRE)+ -> RC
- The role label present at the end of the legal document is mostly RC

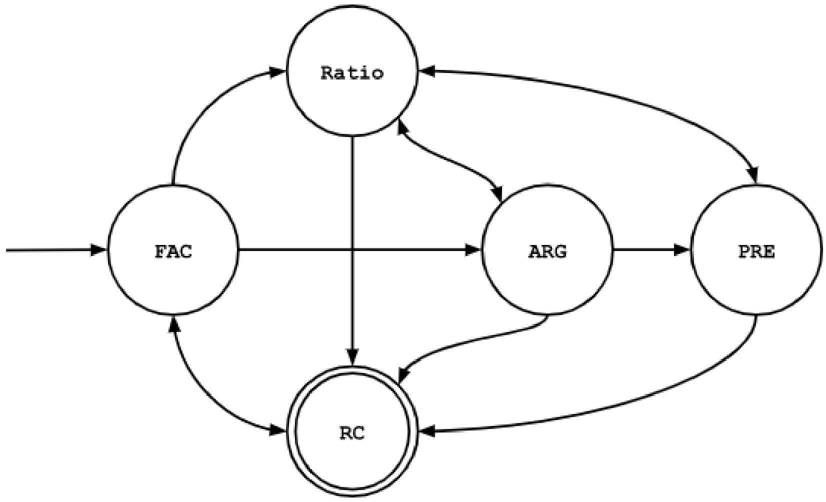


FIGURE 6.2: Exploring finite state automata of role labels

where

```
-> : followed by
/  : or
*  : 0 or more occurrences
+  : 1 or more occurrences
```

It was noticed that we can find the weights for the labels but that requires a lot of expert-annotated documents. Hence, this method is not scalable.

The PRE labelled sentences can also be extracted using a RegEx since they were found to have only certain fixed patterns (examples supplied):

`(____ / (____ vs/v. ____)), YEAR, PRECEDENT(AIR/ SCR/ IPC...), NUMBER`

- `Bhagwati Devi v. Chowdry Bholonath Thakur (1874-75) 2 I.A 256`
- `Shah Mathuradas case 1976 Indlaw SC 400 (supra)`

`section ____ of ____ (IPC, etc.)`

- `Section 409 I.P.C.`
- `Section 3(1) of TADA`

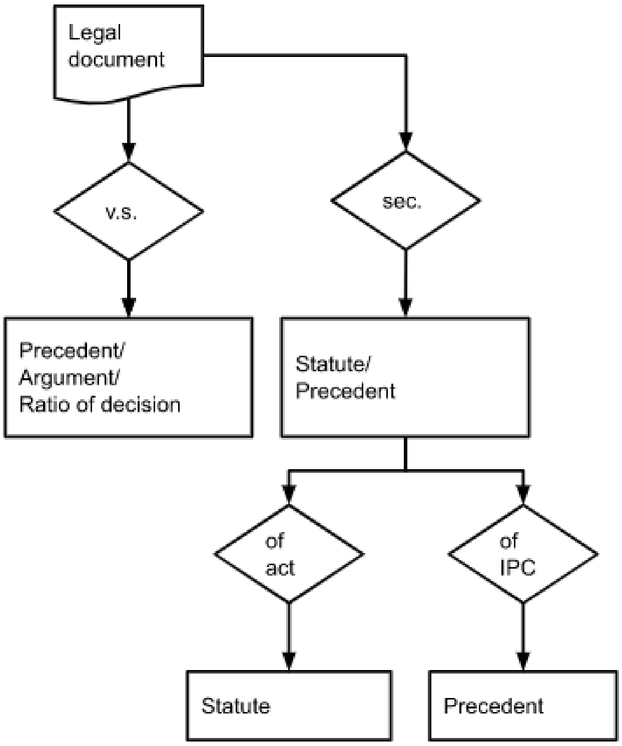


FIGURE 6.3: Exploring Role labels that can be extracted using rules

`Art. ____ of the constitution`

- `Article 226 of the Constitution`

This pattern also captures ARG as well Ratio. The approach in (Pandey et al., 2021) can be followed to use a verb-based method to identify the same. This is to be explored in the future.

Multiple role labels can be assigned to each sentence. Each sentence can be assumed to have the label FAC. All PRE and STA can be extracted using their respective RegEx and the ARG/Ratio can be further explored using verb-based approaches. Similarly, RC can be assessed using a similar verb-based approach.

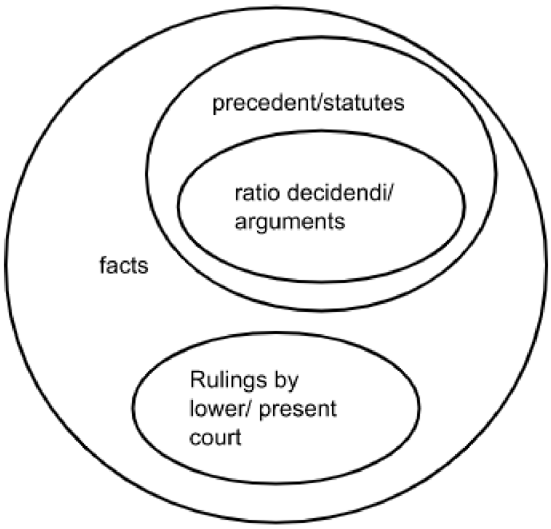


FIGURE 6.4: Exploring Role labels as a hierarchical representation

6.2 Clustering

For the second part of the unsupervised approach, the clustering technique was used. A visual estimation was in this part to estimate the optimum number of clusters. The optimum number of clusters was found to be seven as there was very minimal cluster overlap.

This result is also intuitive as the number of clusters matches the number of legal role labels. Hence, the topic classes for each cluster were studied qualitatively to understand the relationship between the two. It was seen that the clusters were mapped one-to-one to the classes. The basis of establishing a mapping between the clusters and the role labels was as follows: The top topic classes were observed for each cluster and the first topic with an in-topic distribution of more than 50% was considered and mapped logically to the rhetorical roles according to rules defined above, in the first part of the project:

- `STA, PRE by the topics Act, section, sub_section, vs, Article`
- `Ratio by evidence, appear, suit, order, file, claim, involve, say`
- `ARG by consider, question, Act, jurisdiction, require, give`

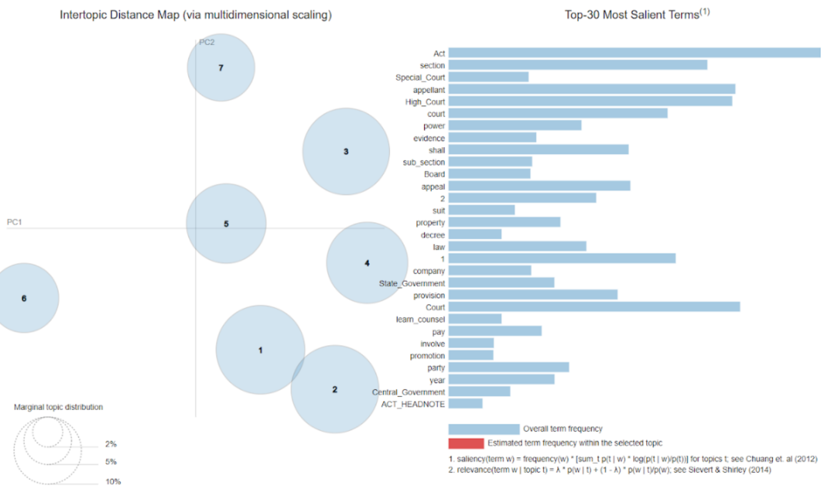
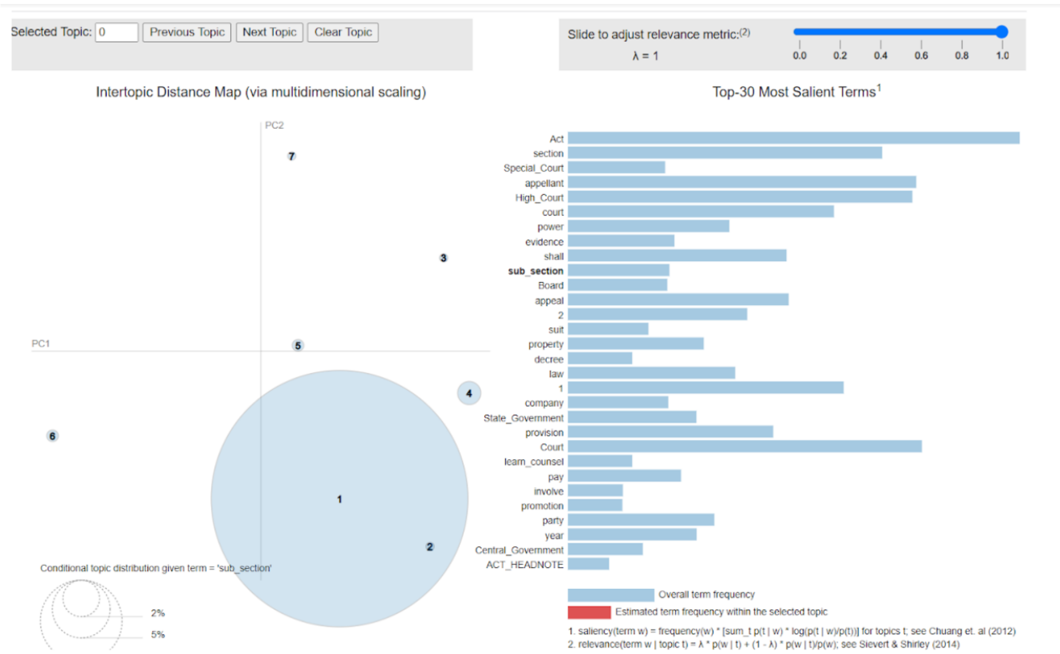
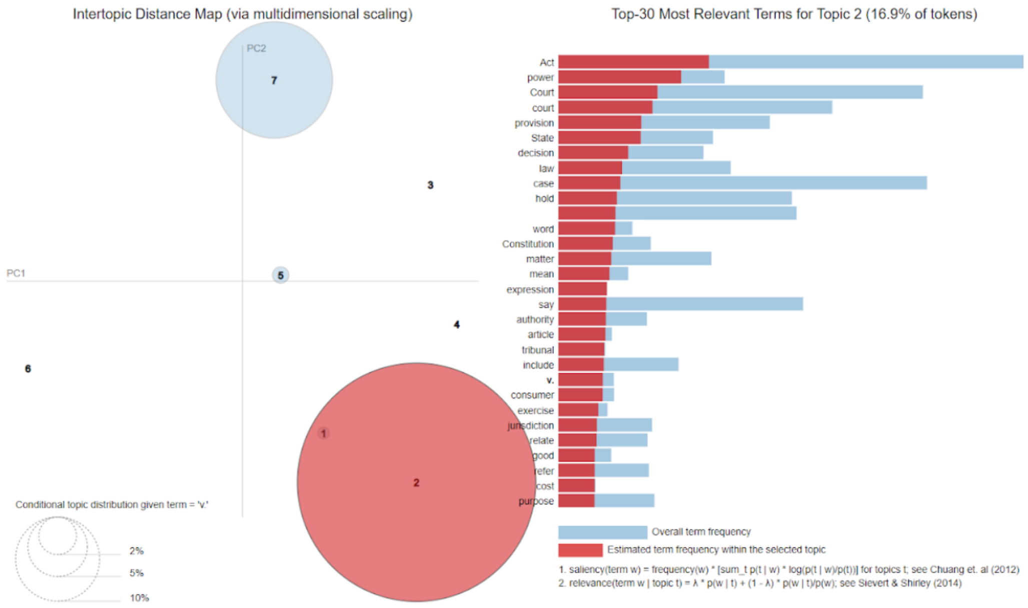


FIGURE 6.5: Exploring cluster labels of a legal document

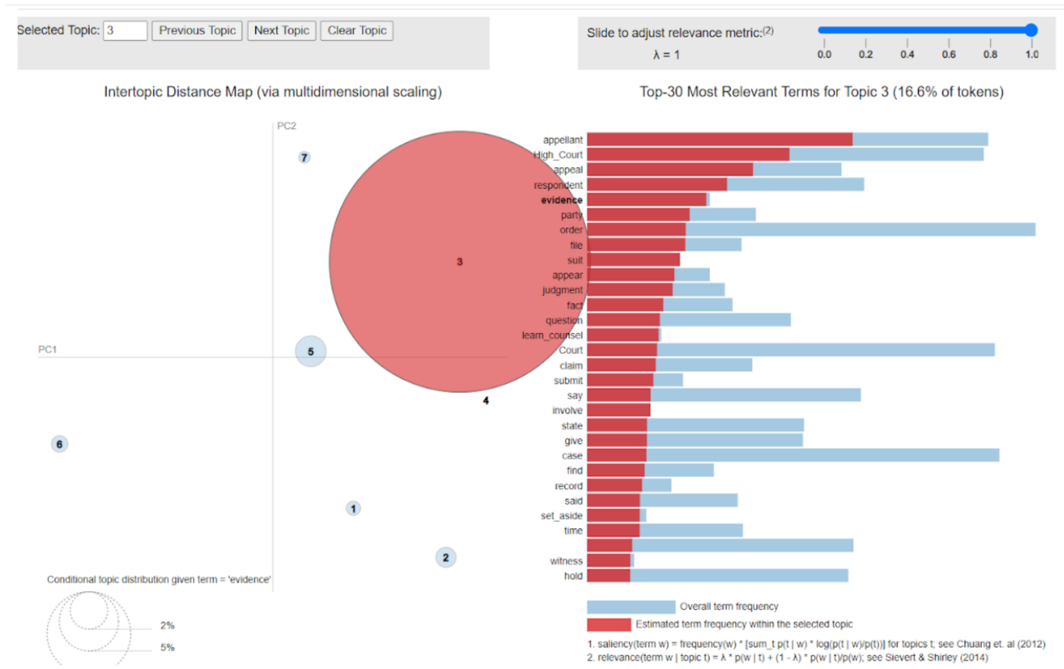


(a) Cluster 1

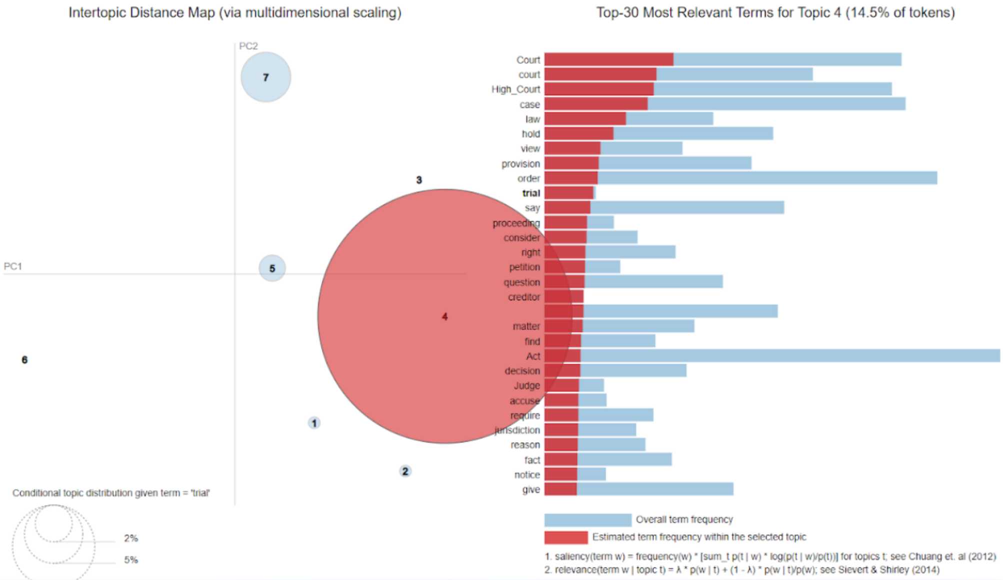
- `RC by find, promotion, give, shall, decree, service, granted`
- `Metadata by Act_Headnote, Judgement, Citation`
- `FAC for the rest because every sentence falls under that category`



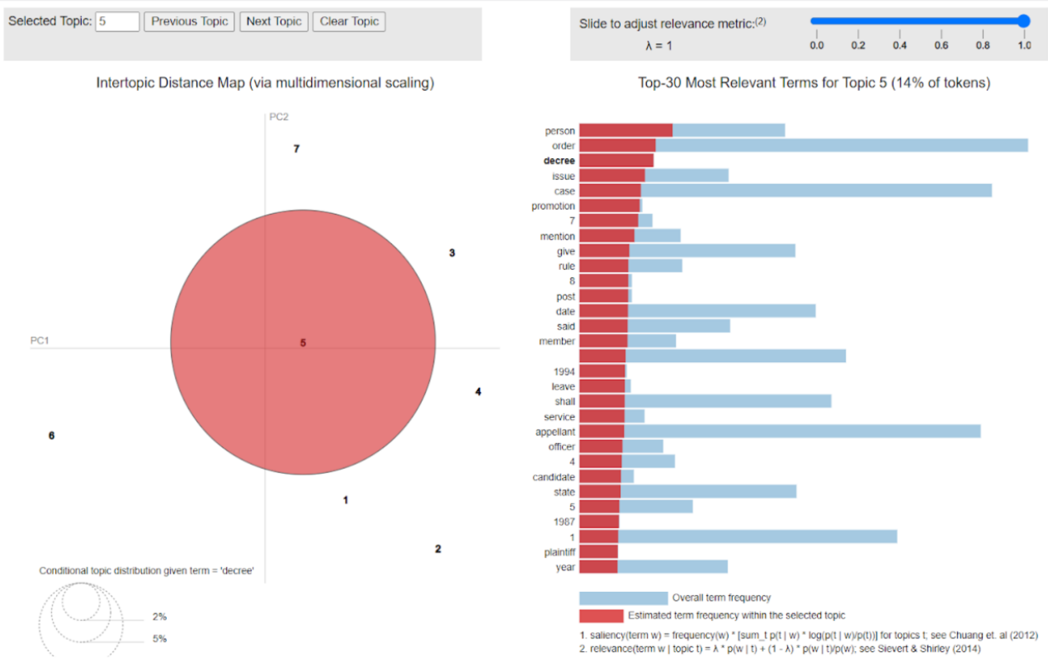
(b) Cluster 2



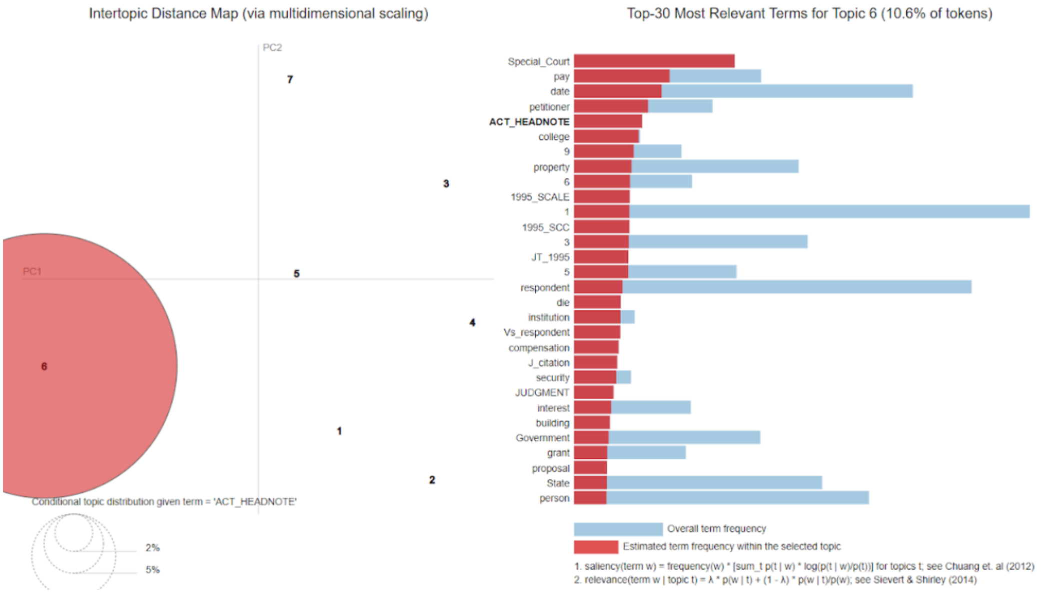
(c) Cluster 3



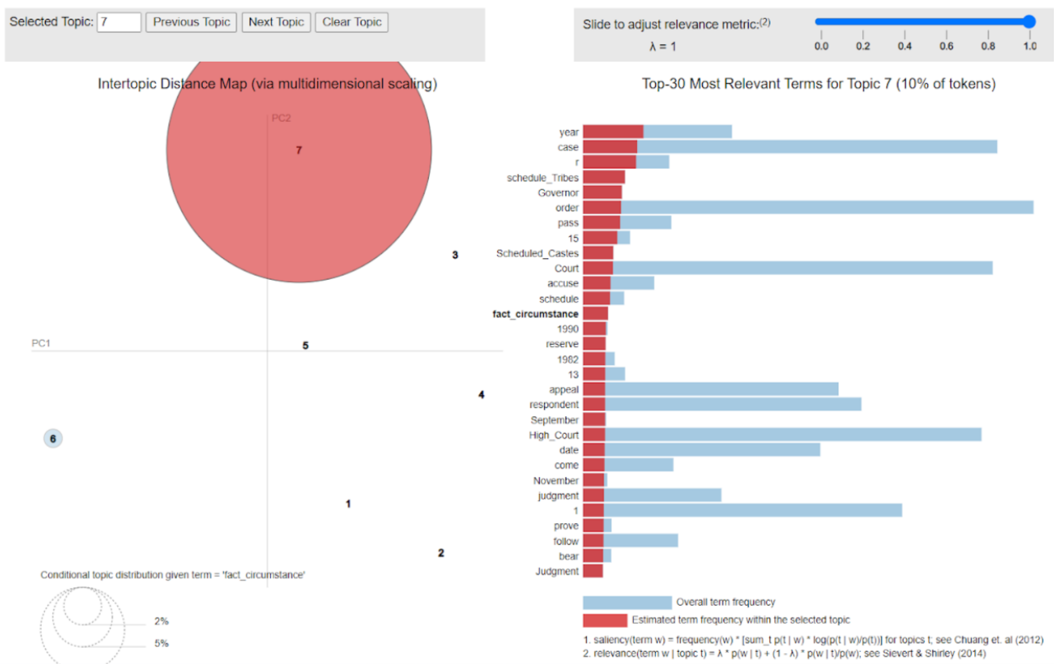
(d) Cluster 4



(e) Cluster 5



(f) Cluster 6



(g) Cluster 7

FIGURE 6.6: Clusters obtained for legal documents

`The following mappings were obtained:`

1. `Cluster 1 maps to STA`
2. `Cluster 2 maps to PRE`
3. `Cluster 3 maps to Ratio`
4. `Cluster 4 maps to ARG`
5. `Cluster 5 maps to RC`
6. `Cluster 6 maps to Metadata`
7. `Cluster 7 maps to FAC`

6.3 Conclusions

Cluster-based labelling is more feasible than supervised role labelling because it does not require labelled training data by experts which is difficult to obtain and it is hence more scalable. Rule-based approaches can also be used because legal language is very well-defined.

Chapter 7

Unsupervised Legal Summary

7.1 Need for unsupervised algorithm

In the legal domain, there is a lot of difficulty in obtaining annotated documents because:

1) The length of legal documents is very long.
2) The length of the legal corpus is huge. There is 1 Supreme Court, 25 High Courts, and 672 District Courts in India. The total number of judgments per year is very high and it is difficult to obtain expert summaries or annotations on the same.
3) The annotations are subjective. The role labels and other annotations might differ from one legal expert to another.
4) It is difficult to find legal experts for annotation.

Due to the issues mentioned above, it is difficult to use supervised algorithms for summarizing legal documents. Hence, we aim to use unsupervised algorithms to achieve summaries of reasonably good quality and which scales.

Automatic text summarization is an essential technique in natural language processing that condenses a large document into a concise summary while preserving

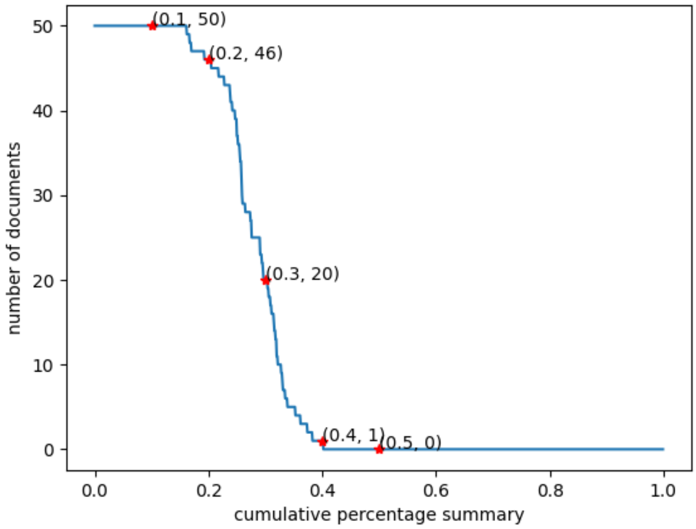


FIGURE 7.1: Determining optimum summary length based on expert summaries

the important information. Summarization can improve the efficiency of information retrieval, reduce the time spent reading, and enhance the comprehension of the original content.

7.2 Determining the optimum summary length

Based on the expert summarised corpus of 50 documents available, the length of expert summaries was determined.

It was observed that 49 out of 50 documents had a summary length $\leq 40\%$. Hence we have taken this as the baseline expert summary length. Next, the aim was to determine what factor of the expert summary length gives the best ROUGE score on the 50 documents gold standard corpus. We performed an analysis with LexRank summaries taking various factors of the Expert summary length as the summary length. For the analysis, the length of LexRank summaries was taken as 0.5E, 0.75E, E, 1.25E, and 1.5E where E was taken as the expert summary length. ROUGE scores against the expert summaries were computed.

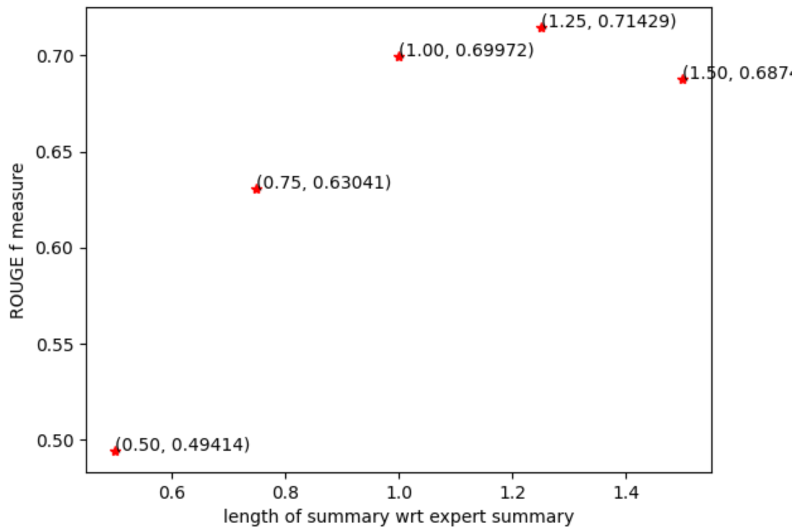


FIGURE 7.2: Ablation study on summary length as a factor of E

The conclusion obtained from this analysis was that 1.25E was the optimum summary length. Hence for all the further analysis, the summary length was taken as 1.25*40% = 50% of document length.

TABLE 7.1: Ablation study on summary length as a factor of E

Length of LexRank Summary	**ROUGE F1 Score**
0.5E	0.49414
0.75E	0.63041
E	0.69972
1.25E (benchmark result)	**0.71429**
1.5E	0.6874

7.3 PageRank-LexRank Union Summarization

We shall now propose a summarization scheme involving the cluster labels we have obtained thus far and show how it enhances the quality of summaries.

We are considering three different summarization schemes: Lexrank, Pagerank on clusters followed by dynamically selected relevant sentences, and Union of LexRank and cluster summaries followed by dynamically selected relevant sentences. We shall conclude how the union method gives the best result for the summarization of legal documents.

7.3.1 LexRank Summarizer

LexRank is designed to summarise a cluster of documents by proposing which sentences subsume the most information in that particular set of documents. In our case, the edge weights are the similarity of the sentences within the document. For some document d, the edge weight between nodes corresponding to sentence s_i and s_j will be sim(s_i, s_j). We are using cosine similarity.

LexRank is a graph-based algorithm that assigns a score to each sentence based on its similarity to other sentences. It considers the document as a graph and applies the principles of eigenvectors to compute the sentence's importance. The LexRank algorithm is formulated as follows:

$$S(i) = \Sigma_{j \in adj[i]} \frac{S(j)}{deg(j)}$$

Also, continuous LexRank is defined as follows:

$$S(i) = \frac{d}{N} + (1-d)\Sigma_{j \in adj[i]} \frac{sim(i,j)}{\Sigma_{k \in adj[j]} sim(j,k)} S(j)$$

where,

$S(i)$ = LexRank of sentence i

$adj(i)$ = set of sentences connected to sentence i

$deg(i)$ = number of sentences connected to sentence i

$sim(i,j)$ = cosine similarity between sentences i and j

The LexRank algorithm generates a summary of the document, where each sentence represents a set of sentences that are highly similar to other sentences.

7.3.2 Cluster Summarizer

We propose to cluster the documents (50 expert-summarised corpus) into 7 clusters using LSA clustering by giving a seed corpus of 550 documents (500 from the supreme court + 50 experts labelled). We perform the PageRank algorithm on each cluster to remove redundant sentences which are already captured by the selected nodes. We call these cluster summaries. We take a union of all the cluster summaries and dynamically select the sentences so that they satisfy a length constraint (we are setting 1.25 times the expert summary length which roughly corresponds to 45-50% of document length) and the similarity of the summary with the original document is maximized. This gives the desired summary.

The code takes a document as input and performs text summarization in several stages. Firstly, the document is clustered into seven clusters using the LDA algorithm. LDA is a generative statistical model that assigns topics to text documents based on the probability distribution of words. It represents each document as a mixture of topics and each topic as a distribution of words. The LDA algorithm is formulated as follows:

Formula:

$$LDA : p(w, z, \Theta, \Phi \mid \alpha, \beta) = \prod_{d} p(\Theta_d \mid \alpha) \prod_{n} p(w_{d,n} \mid z_{d,n}, \Phi) p(z_{d,n} \mid \Theta_d)$$

where,

w = word

z = topic

Θ = topic distribution

Φ = word distribution

α, β = hyperparameters

The LDA algorithm generates seven clusters of the document, where each cluster represents a set of sentences that share similar topics.

Secondly, the code generates a PageRank summary of each cluster. PageRank is a graph-based algorithm that assigns a score to each sentence in a document based on its connectivity with other sentences. It considers the document as a graph and applies the principles of random walks to compute the sentence's importance. The PageRank algorithm is formulated as follows:

$$PR(u) = \frac{(1-d)}{N} + d.\frac{\Sigma_{v \in B(u)} PR(v)}{\mid B(v) \mid}$$

where,
$PR(u)$ = PageRank of node u
d = damping factor
$B(v)$ = set of nodes pointing to node v
N = total number of sentences

The PageRank algorithm generates a summary for each cluster of the document, where each summary represents a set of sentences that are highly ranked based on their connectivity with other sentences.

7.3.3 Union or Knapsack Summarizer

We take a union of all the cluster summaries and the LexRank summary (corresponding to 1.25 times the expert length). We dynamically select the sentences so that they satisfy a length constraint (we are setting 1.25 times the expert summary length which roughly corresponds to 40% of document length) and the similarity of the summary with the original document is maximized. This gives the desired summary. This is alternatively called the Knapsack summarizer.

7.4 Pseudocode

Algorithm 1 Get Cluster Summaries

Require: A list of clusters and a list of sentences
Ensure: A list of summaries, one for each cluster

```
clusterSummaries ← [ ]
for sentence ∈ document do
    sentenceCluster[sentence] ← cluster assignment from clusters
end for
for c ∈ clusters do
    clusterSentences[c] ← all sentences in cluster c
    clusterText ← JOIN(clusterSentences)
    clusterSummaries ← clusterSummaries + pagerank(clusterText, ratio)
end for
return clusterSummaries
```

Algorithm 2 Create Summary

Require: All the cluster summaries and the max length

```
originalLength ← 0
for sentence in sentences do
    originalLength ← originalLength + LEN(sentence)
end for
maxSummaryLength ← maxLength ∗ originalLength
sentenceLengths ← [LEN(sentence) for sentence in sentences]
selectedSentences ← select sentences(sentences, maxLength)
summary ← JOIN(selectedSentences)
return summary
```

This pseudocode represents a function called *selectSentences* that takes a list of sentences doc and a maximum length *maxLength* as input and returns a selected list of sentences based on their similarity to the original document. It uses dynamic programming to compute the optimal subset of sentences that best represents the original document while also satisfying the maximum length constraint.

Algorithm 3 select sentences

Require: A list of sentences and maximum length

$similar \leftarrow$ cosine similarity of sentence vectors concerning the document
$n \leftarrow \text{LEN}(sentences)$
for $i \leftarrow 1$ to n **do**
    $weights[i] = LEN(sentences[i])$
end for
$maxWeight \leftarrow maxLength \times \text{LEN}(sentences)$
$dp \leftarrow$ array of zeros of size: $(n+1) \times (maxWeight + 1)$
for $i \leftarrow 1$ to n **do**
    for $w \leftarrow 0$ to $maxWeight$ **do**
        if $weights[i-1] > w$ **then**
            $dp[i][w] \leftarrow dp[i-1][w]$
        else
            $dp[i][w] \leftarrow \text{MAX}(dp[i-1][w], dp[i-1][w - weights[i-1]] + similar[i-1])$
        end if
    end for
end for
$selected \leftarrow []$
$i \leftarrow n$
$w \leftarrow maxWeight$
while $i > 0$ and $w > 0$ **do**
    if $dp[i][w] = dp[i-1][w]$ **then**
    else
        $selected.\text{APPEND}(sentences[i-1])$
        $w \leftarrow w - weights[i-1]$
    end if
    $i \leftarrow i - 1$
end while
$selected.\text{REVERSE}()$
return $selected$

Chapter 8

Intrinsic evaluation of Knapsack summarizer

8.1 ROUGE Scores

For our analysis, we are using the dataset introduced in Bhattacharya et al. (2021b). This is an expert summarized corpus of 50 Supreme Court documents.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of evaluation metrics that are commonly used to measure the quality of automatic summarization and machine translation outputs. The primary purpose of ROUGE is to assess how well a system-generated summary or translation captures the essential information from the reference summary or translation. The ROUGE metric is based on a comparison of the n-gram overlap between the system-generated output and the reference summary.

ROUGE scores are typically computed as precision, recall, and F1-scores for various n-gram lengths (unigrams, bigrams, trigrams, etc.) and for different types of matching (exact, partial, stemmed, etc.). The formulas for precision, recall, and F1 scores are as follows:

$$Precision = \frac{Number\ of\ overlapping\ Ngrams}{Number\ of\ Ngrams\ in\ generated\ summary}$$

$$Recall = \frac{Number\ of\ overlapping\ Ngrams}{Number\ of\ Ngrams\ in\ reference\ summary}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The overall ROUGE score is calculated by averaging the individual F1 scores across all the n-gram lengths and matching types. Higher ROUGE scores indicate better performance of the system in generating summaries or translations that are similar to the reference summaries or translations.

Table 8.1: Intrinsic evaluation of Cluster-based summary for document 1

Length of Summary	ROUGE p	ROUGE r	ROUGE F1
No dynamic programming involved	0.956521739130434	0.628571428571428	0.758620684869203
Same as expert summary length	0.823529411764705	0.8	0.811594197899601
1.25 times expert summary length	0.956521739130434	0.628571428571428	0.758620684869203
30% of original document length	0.8	0.8	0.799999995
40% of original document length	0.823529411764705	0.8	0.811594197899601

From the above results, we see that PageRank might fail to capture some of the sentences of the document so we take a union of PageRank summary and the cluster summaries and then perform dynamic programming in the Union/Knapsack summarizer method.

Table 8.2: Intrinsic evaluation of Knapsack based summary for document 1

Length of Summary	ROUGE p	ROUGE r	ROUGE F1
No dynamic programming involved	0.956521739130434	0.628571428571428	0.758620684869203
Same as expert summary length	0.914285714285714	0.761904761904761	0.864864859956172
1.25 times expert summary length	0.941176470588235	0.809523809523809	0.894736837160664
30% of original document length	0.823529411764705	0.8	0.811594197899601
40% of original document length	0.8	0.8	0.799999995

In conclusion, PageRank successfully captures some of those parts of a document that are missed by LexRank.

Table 8.3: Intrinsic evaluation of summaries for 1.25E length

Length of Summary	ROUGE p	ROUGE r	ROUGE F1
Cluster	0.956607762616598	0.545670877367153	0.705310988953237
Union	0.895593029275713	0.629246986300369	**0.802470325397098**

8.2 Summarisation - Similarity analysis

As defined earlier, the Summarization-Similarity analysis considers the top-K retrieved similar documents for two documents and returns the fraction of common documents. This analysis is performed for LexRank and Knapsack-based summaries with the set baseline for the full document case.

When evaluating the effectiveness of information retrieval systems, various metrics are available to measure different aspects of performance. However, in some cases, the baseline list may have a lower confidence score, making it challenging to use certain metrics. In such cases, summarization with similarity analysis can be a useful tool.

One common scenario where the baseline list may have a lower confidence score is in multi-document summarization. In this case, the system is tasked with summarizing a set of related documents into a single coherent summary. The baseline list may be a collection of all the documents, each with varying levels of relevance and importance. In such cases, it can be difficult to use metrics such as precision, recall, or average precision since these metrics require a clear distinction between relevant and irrelevant documents.

Summarization with similarity analysis, on the other hand, can be useful in this scenario. By comparing the similarity between the original set of documents and the summary produced by the system, it is possible to evaluate the effectiveness of the summarization method. The similarity analysis can be performed using different measures, such as cosine similarity or Jaccard similarity, to identify the level of overlap between the original documents and the summary.

One advantage of using similarity analysis in this scenario is that it provides a more nuanced evaluation of the summarization method. Instead of simply measuring the number of relevant documents or their rank, similarity analysis takes into account the overall content and structure of the documents. This can be particularly useful when dealing with sets of related documents, where the relevance and importance of individual documents may vary.

Table 8.4: Summarisation - Similarity analysis of Knapsack based summary for document 1

Value of K	LexRank	Knapsack
K = 5	2	3
K = 10	7	5
K = 20	14	17
K = 30	26	27
K = 40	35	37

Hence the proposed summarizer improves similarity values. This result is in concordance with the triad relationship. We conclude empirically that better summarizers lead to better similarity ranking.

Chapter 9

Top - K Analysis Metric

9.1 Introduction

We now introduce a new evaluation metric called the Top - K analysis Metric for the evaluation of the summaries produced.

Top K analysis is a common evaluation metric used in information retrieval and summarization tasks to measure the effectiveness of a system in retrieving relevant documents. The metric is based on selecting a fixed number of documents, referred to as K1, from the full document list and then identifying the last document in the summary list whose similarity value exceeds a threshold. This document is referred to as K2, and the metric is computed based on the common documents between the two lists.

We have normalized the cosine similarity document-wise to overcome cases where the distribution of similar documents varies greatly for two documents:

$$\text{Normalising factor for } document_j = \frac{1}{\Sigma_1^{N-1} sim(document_i, document_j)}\ ; i! = j$$

9.2 Algorithmic formulation

The Top K analysis metric is computed as follows:

1. First, a subset of K1 documents is selected from the full document list. The documents in this subset are assumed to be the most relevant documents in the list.

2. For each document in the subset, the similarity value with all other documents in the list is computed. The maximum similarity value for each document is recorded, and the threshold is set to the value of the K1th document.

3. Next, the summary list is generated using the summarization method under evaluation.

4. The last document in the summary list whose similarity value is greater than or equal to the threshold is identified. This document is referred to as K2.

5. The metric is computed by calculating the number of common documents between the K1 subset and the summary list up to K2.

The usefulness of the Top K analysis metric is that it allows for the evaluation of systems when the exact number of relevant documents is unknown or difficult to determine. In some cases, the full document list may be too large to manually evaluate or may not contain a clear distinction between relevant and irrelevant documents. In such scenarios, Top K analysis can be used to evaluate the system based on a subset of relevant documents.

Another advantage of Top K analysis is that it provides a simple and intuitive way to evaluate the effectiveness of summarization methods based on the overlap between the original documents and the summary. The metric can be used to compare different summarization methods and identify the one that generates summaries with the highest overlap with the most relevant documents.

Algorithm 4 Top K analysis metric algorithm

Require: Full document list D, summary list S, number of documents $K1$ to select from D

Ensure: Precision and Recall metrics

$fullDocumentSublist \leftarrow \{d_{F_1}, d_{F_2}, \ldots, d_{F_{K1}}\}$ top $K1$ documents from list D
$threshold \leftarrow cosineSimilarity(d_{F_{K1}}, D)$
Sort the summary list S in descending order of similarity value
Find the index $K2$ of the last document in S whose similarity value is greater than or equal to $threshold$
$K2 \leftarrow 1$, $lower \leftarrow 1, upper \leftarrow N$
while $lower <= upper$ **do**
  $index = \frac{lower+upper}{2}$
  if $cosineSimilarity(d_{S_{index}}, D) >= threshold$ **then**
    $K2 \leftarrow index$
    $lower \leftarrow index + 1$
  else
    $upper \leftarrow index - 1$
  end if
end while
$K2 \leftarrow MAX(K1, K2)$
$summarySublist \leftarrow \{d_{S_1}, d_{S_2}, \ldots, d_{S_{K2}}\}$ top $K2$ documents from list S
$commonSublist \leftarrow fullDocumentSublist \bigcap summarySublist$
$common \leftarrow LEN(commonSublist)$
$precision \leftarrow \frac{common}{K2}$
$recall \leftarrow \frac{common}{K1}$
return $precision, recall$

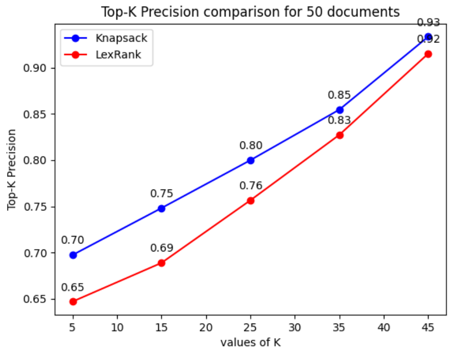
9.3 Intrinsic evaluation of Knapsack summarizer

TABLE 9.1: Top - K analysis on Gold standard data for LexRank summaries

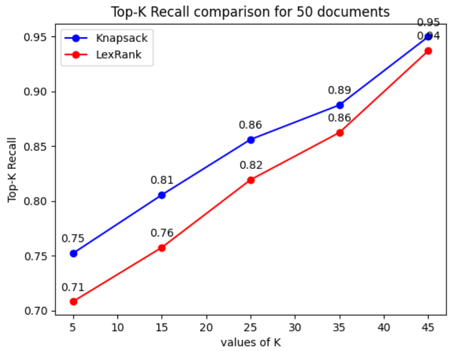
K1	Top-K Precision	Top-K Recall	Top-K F1 Score	K2
5	0.647027313266443	0.708	0.658491021770221	2.76
15	0.688842845592845	0.757333333333333	0.710964618485035	7.2
25	0.75650963736278	0.8192	0.781960417280232	14.94
35	0.827275961931677	0.862285714285714	0.84288051794843	26.62
45	0.915123724358691	0.936888888888888	0.925587717906971	41.52

TABLE 9.2: Top - K analysis on Gold standard data for Knapsack summaries

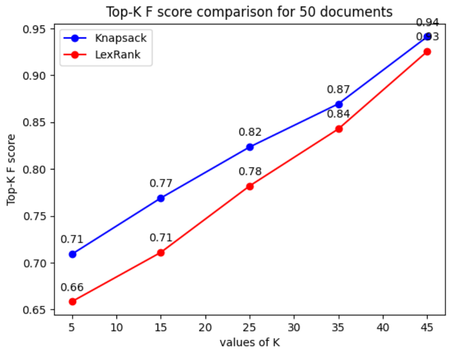
K1	Top-K Precision	Top-K Recall	Top-K F1 Score	K2
5	0.69737422126041	0.752	0.709110115937702	2.56
15	0.748088697719784	0.805333333333333	0.769005088809292	6.3
25	0.799878520246813	0.856	0.823518709244908	14.36
35	0.854526745650935	0.887428571428571	0.869653908199953	26.94
45	0.933432196190224	0.949777777777777	0.941327626572807	41.94



(a) Top - K Precision



(b) Top - K Recall



(c) Top - K F1 score

FIGURE 9.1: Top - K analysis on Gold standard dataset

In conclusion, we have established Top-K analysis as our method of analysis, and the knapsack-based method performs better in 50-document cases. This establishes the metric.

Chapter 10

Extrinsic Evaluation of Knapsack Summarizer

10.1 Need for extrinsic evaluation

In the legal domain, natural language processing (NLP) models are increasingly being utilized to perform various tasks such as document classification, sentiment analysis, and information extraction. However, due to the highly specialized and complex nature of legal language, the lack of annotated data can pose a significant challenge in evaluating the effectiveness of NLP models. This is where extrinsic evaluation comes in, as it allows for the assessment of the performance of NLP models in real-world tasks that are relevant to the legal domain. Without extrinsic evaluation, there is a risk of overfitting the limited annotated data that is available, which can result in poor generalization to new datasets and ultimately hinder the adoption of NLP models in the legal domain. Thus, extrinsic evaluation is crucial to ensure the accuracy and reliability of NLP models in the legal domain.

10.2 Coherency of the Top - K Analysis Metric

1. The performance of the Top K analysis metric algorithm is highly relevant and important for information retrieval and summarization tasks, especially when focused on smaller K values.

2. Furthermore, the precision and recall values obtained for smaller K values are more meaningful and statistically significant, as they represent the quality of the most relevant results that are presented.

3. In the context of metrics used for evaluating the performance of an algorithm, scaling is important because it ensures that the metric remains consistent across different data sets and problem sizes.

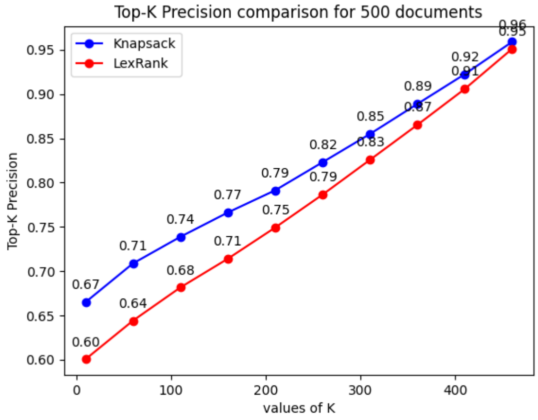
4. Scaling of a metric also implies correctness and coherence because it ensures that the metric is measuring what it is intended to measure.

10.3 Scaling of the Top - K Analysis Metric

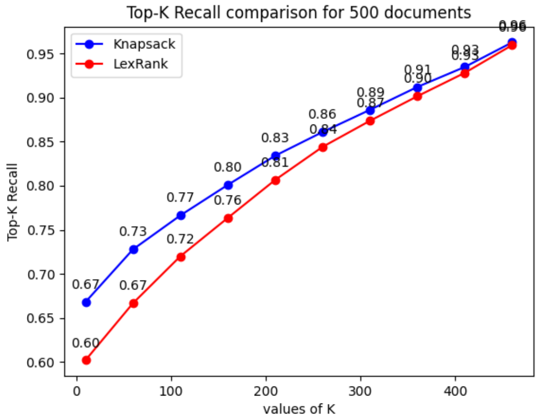
Now we scale the Top-K Analysis metric, defined earlier, to larger datasets and simultaneously perform an ablation study of the LexRank-based summaries and the Knapsack-based summaries. We have taken a corpus from Supreme Court judgments of sizes 500, 1000, and 2000. We have parsed the documents using a legal domain parser implemented earlier. Two important conclusions from the obtained plots are:

1. The Top-K Analysis metric scales and is thus coherent.

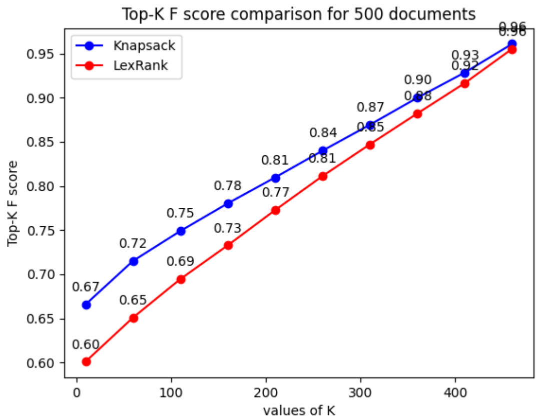
2. The Knapsack summarizer is significantly better than the LexRank summarizer, especially for lower values of K1.



(a) Top - K Precision

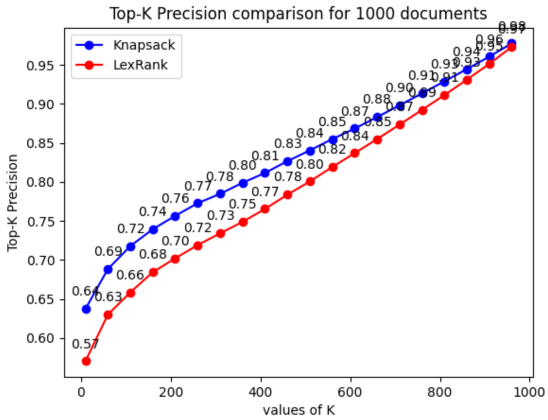


(b) Top - K Recall

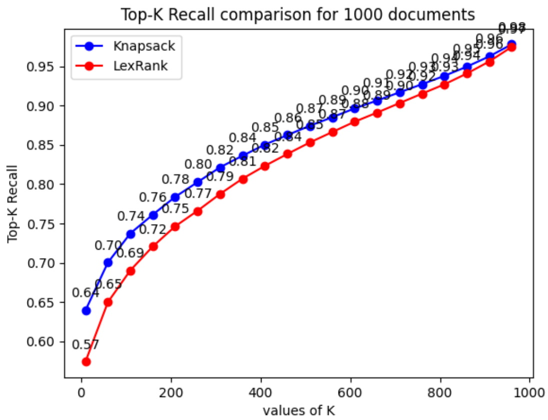


(c) Top - K F1 score

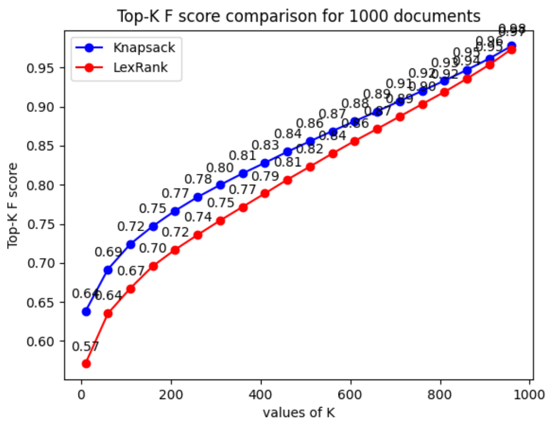
Figure 10.1: Top - K analysis on 500 document dataset



(a) Top - K Precision

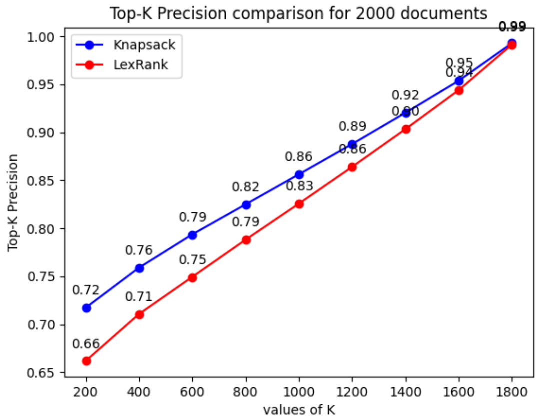


(b) Top - K Recall

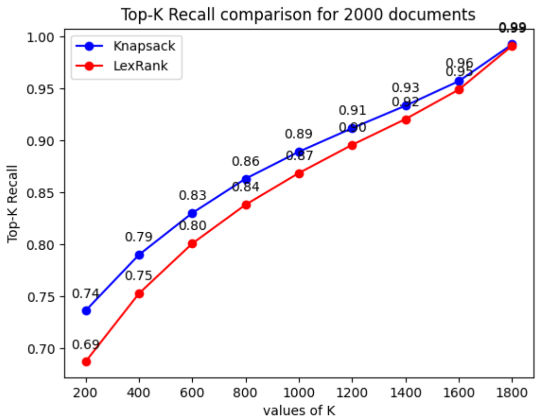


(c) Top - K F1 score

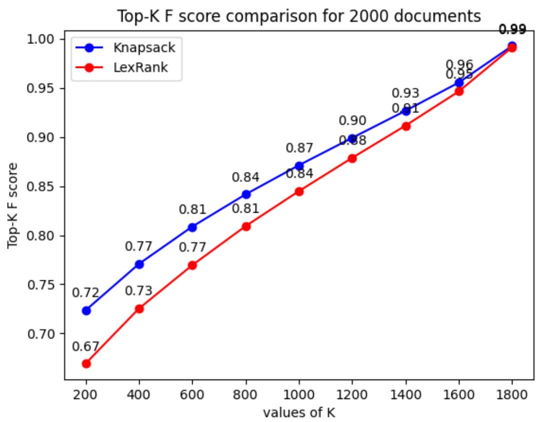
FIGURE 10.2: Top - K analysis on 1000 document dataset



(a) Top - K Precision



(b) Top - K Recall



(c) Top - K F1 score

FIGURE 10.3: Top - K analysis on 2000 document dataset

10.4 Correlation Analysis

10.4.1 Kendall's Tau

Kendall's tau (τ) is a non-parametric measure of the association between two variables. Given two variables X and Y, and a set of n pairs (x_i, y_i), Kendall's tau can be calculated as follows:

First, we define:

- concordant pairs (C): pairs of observations that have the same order both $x_i > x_j$ and $y_i > y_j$, or both $x_i < x_j$ and $y_i < y_j$
- discordant pairs (D): pairs of observations that have opposite order both $x_i > x_j$ and $y_i < y_j$, or both $x_i < x_j$ and $y_i > y_j$
- ties in X (T_X): the number of ties in the X variable
- ties in Y (T_Y): the number of ties in the Y variable

Then, Kendall's tau can be calculated as:

$$\tau = \frac{C - D}{\sqrt{(C + D + T_X)(C + D + T_Y)}}$$

The resulting value of τ is a number between -1 and 1, where:

1. $\tau = 1$ indicates a perfect positive association (i.e., all pairs are concordant)
2. $\tau = -1$ indicates a perfect negative association (i.e., all pairs are discordant)
3. $\tau = 0$ indicates no association (i.e., the number of concordant pairs equals the number of discordant pairs)

10.4.2 Spearman's Coefficient

Spearman's rank correlation coefficient (ρ) is another non-parametric measure of the association between two variables. Given two variables X and Y, and a set of n pairs (x_i, y_i), Spearman's correlation can be calculated as follows:

First, we rank the observations in each variable, from lowest to highest, assigning them the ranks 1, 2, ..., n. If there are ties, we assign them the average of the ranks they would have received if they were not ties. This results in two sets of ranked observations, R_X and R_Y.

Then, Spearman's correlation can be calculated as the Pearson correlation coefficient between the two sets of ranks:

$$\rho = \frac{\mathrm{cov}(R_X, R_Y)}{\sigma_{R_X}\sigma_{R_Y}}$$

where *cov*() is the covariance, and σ_{R_X} and σ_{R_Y} are the standard deviations of the ranks in X and Y, respectively.

The resulting value of ρ is a number between -1 and 1, where:

1. $\rho = 1$ indicates a perfect positive association (i.e. when X increases, Y increases)
2. $\rho = -1$ indicates a perfect negative association (i.e., when X increases, Y decreases)
3. $\rho = 0$ indicates no association

10.4.3 Results

A low p-value indicates that the observed correlation between two variables is unlikely to have occurred by chance alone. The p-value is the probability of observing the observed correlation or a more extreme correlation, assuming the null hypothesis is true. The null hypothesis is that there is no correlation between the two variables.

Table 10.1: Correlation Analysis on the 500 documents

	Kendall's Tau	**Spearman's correlation**
Avg Correlation in Knapsack	0.00976063660114243	0.0145168654372355
Avg Correlation in Lexrank	0.00782889321463971	0.0116695473126428
Avg p-value in Knapsack	0.478147319581072	0.480181434539862
Avg p-value in Lexrank	0.487971627662335	0.489520381038564

On the other hand, a low correlation coefficient indicates a weak or no linear relationship between the two variables. A value of 0 indicates no linear relationship, while a value of -1 or +1 indicates a perfectly negative or positive linear relationship, respectively.

Conclusion: In the Knapsack case, there is a more significant association between the two lists as opposed to the LexRank summarizer case. This implies that the Knapsack yields lists which are statistically more correlated as opposed to LexRank. This also shows that our Top-K Analysis metric is coherent.

Chapter 11

Ablation study of extractive summarizers

11.1 Extractive summarization of legal documents

The availability of annotated data in the legal domain is relatively low. Therefore, abstractive summarizers may not be the best option in this domain, as they require large amounts of annotated data to train the models accurately.

The summarization output from extractive summarizers is more transparent and easier to understand. We can also utilize existing role labeling if the original sentences are used. This is also inherent in the triad relationships we are trying to establish through this study.

Extractive summarizers scale easily; hence, we are considering them in our study. In the legal domain, scaling is a very important factor.

We have used the sumy library to perform the study between different domain-independent extractive summarizers. We have compared them to our Knapsack summarizer on the established Top - K Analysis metric.

Table 11.1: Features of various extractive summarizers

Algorithm	**Concept**	**Features**	**Advantages**	**Disadvantages**
TextRank	PageRank for text	Graph-based representation of sentences to find most important ones	Fast and effective doesn't require training data	May not perform well on small datasets
LexRank	Cosine similarity for sentence similarity	Graph-based approach takes into account the global context	Performs well on long texts, handles multiple languages	May produce redundant or incomplete summaries
LSA	Singular value decomposition to represent text as a matrix	Captures latent semantic information in text	Effective at capturing relationships between words and concepts	May require large amounts of data to be effective
Luhn's	Ranks sentences based on the frequency of important words	Uses simple word frequency analysis	Easy to implement and understand	Can be too simplistic and may miss important information
KLSum	Uses KL divergence to measure sentence similarity	Focuses on capturing most relevant information	Performs well on long documents	May be slow and computationally expensive
SumBasic	Ranks sentences based on the frequency of important words	Simple approach that can be effective	Fast and efficient	Can produce redundant or incomplete summaries

We are interested in the following points:

1. Should we use KL-divergence as the distance metric instead of cosine similarity?

2. How does our Knapsack-based algorithm perform as opposed to other extractive summarizers?

3. How are the results for lower values of K?

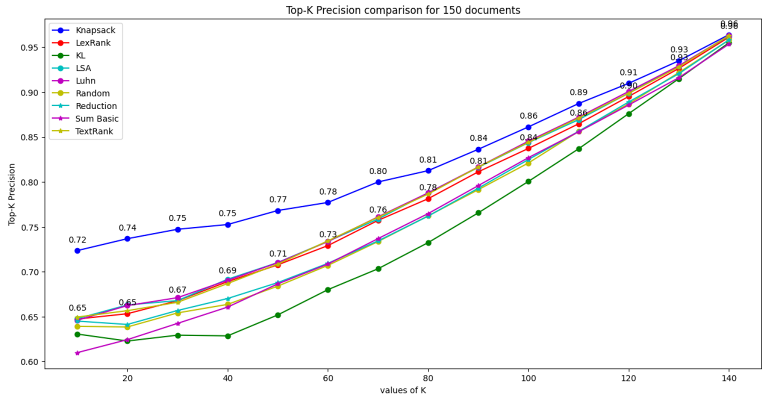
4. And for a larger corpus, does our metric scale?

Results:

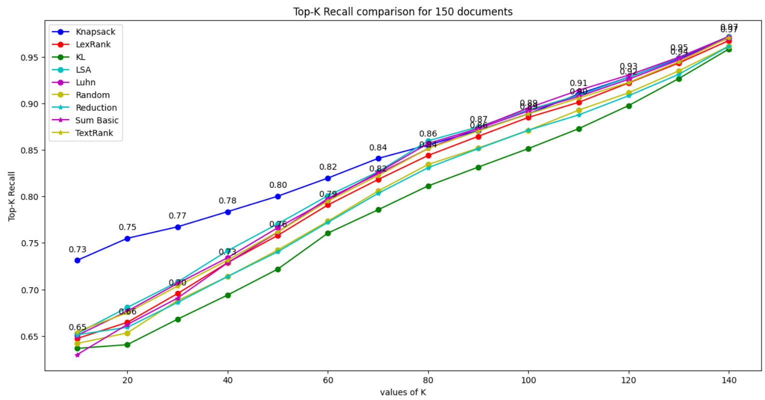
1. Cosine similarity is a better metric than KL divergence in the case of legal documents.

2, 3. Knapsack-based summarizer outperforms all the other extractive summarizers for all values of K and the results are very prominent for smaller values of K.

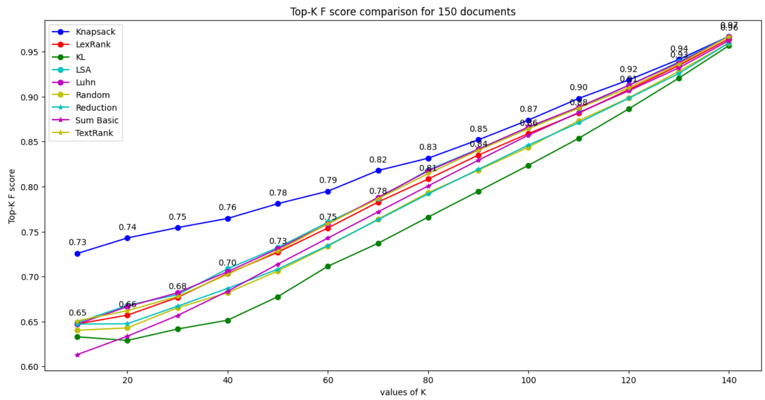
4. Our metric is coherent after scaling the corpus from 150 documents to 500 documents.



(a) Top - K Precision

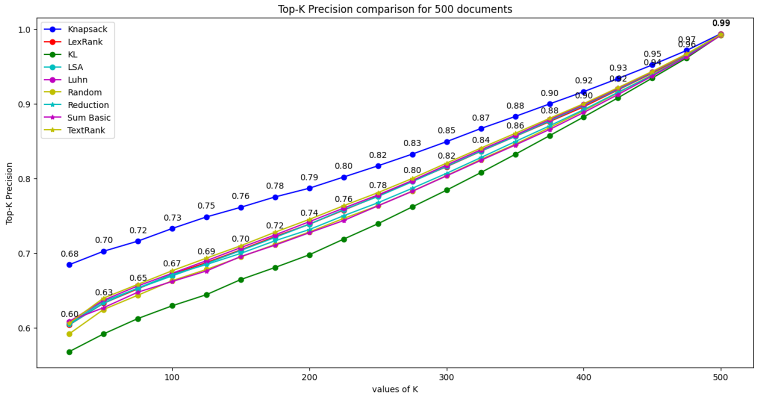


(b) Top - K Recall

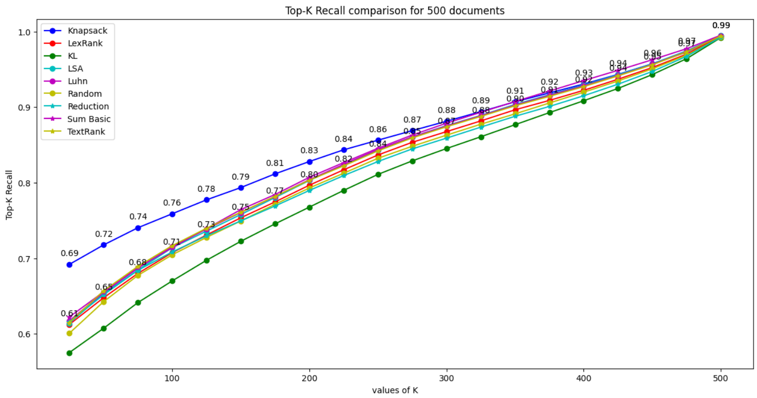


(c) Top - K F1 score

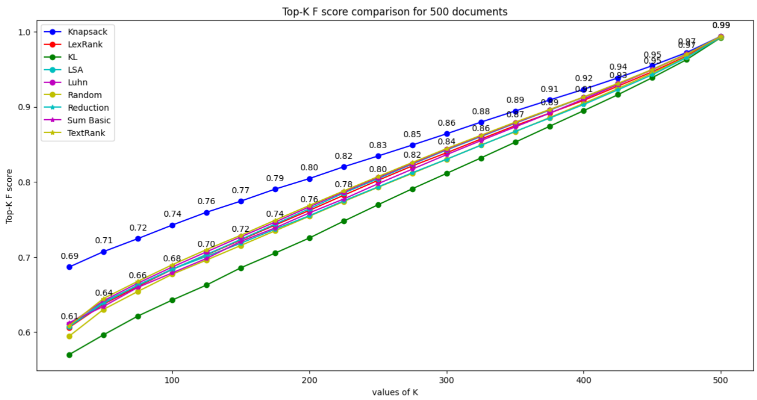
FIGURE 11.1: Top - K analysis on 150 documents: Domain independent summarizers



(a) Top - K Precision

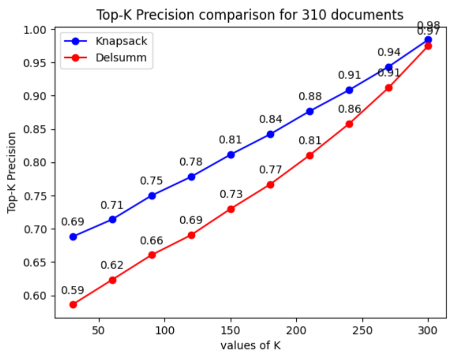


(b) Top - K Recall

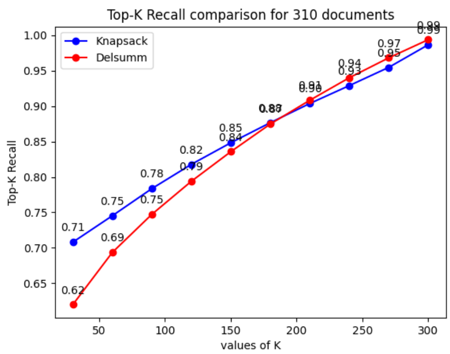


(c) Top - K F1 score

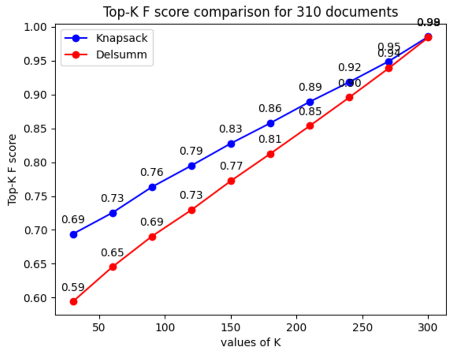
FIGURE 11.2: Top - K analysis on 500 documents: Domain independent summarizers



(a) Top - K Precision



(b) Top - K Recall



(c) Top - K F1 score

FIGURE 11.3: Top - K analysis on 310 documents: Domain-specific summarizers

11.2 Domain-specific extractive summarization

Delsumm was introduced in Bhattacharya et al. (2021b) and has outperformed all existing legal domain summarizers. We conduct an ablation study of DelSumm and our proposed Knapsack-based summarizer on the established Top-K Analysis metric. The analysis has been done on a corpus of size 310 documents.

In conclusion, the Knapsack-based algorithm outperforms both domain-independent and domain-specific extractive summarizers, especially for smaller values of K.

Chapter 12

Visual Summary

12.1 Components

To make the representation of legal domain documents easy, we have introduced a visual summary data structure. It is a hierarchical representation of the triad of summarization, similarity, and role labeling for a document. It includes an interactive graph on the left-hand side panel, which displays the content in the dynamic right-hand side panel. The graph is implemented using JsNetworkX, d3.js, and jQuery, and the complete webpage has been implemented using JavaScript, HTML, and CSS. The server has been set up using Python

12.2 Need for a visual summary

1. Reference: A visual summary provides an easy reference point for the document's key points.

2. Time-Saving: Visual summaries can quickly and efficiently convey key information without requiring a significant investment of time.

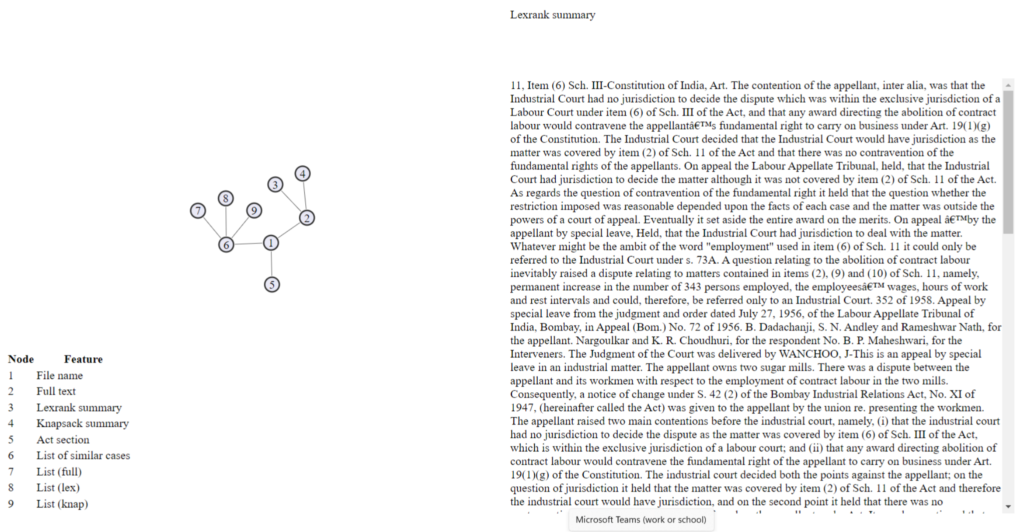


FIGURE 12.1: Visual summary

3. Clarity: By presenting information in a more visually engaging format, the summary can help ensure that everyone involved has a clear understanding of the document's intent.

4. Decision Making: A visual summary can help those making decisions to quickly and easily grasp the essential information and make informed choices.

5. Accessibility: By presenting information in a more accessible format, the document's main points can be conveyed to a wider audience.

6. Presentation: It can also be used as a tool for presentations, helping to convey key points in a more visually appealing format.

The technical aspects of summarization, similarity, and role labeling are crucial in creating effective visual summaries that accurately convey the document's main points.

Chapter 13

Fetching Similar Documents: An Algorithmic Approach

Efficiently searching and retrieving information from large amounts of data is crucial in today's information age. One method of achieving this is by retrieving similar documents based on a query document. This involves comparing the query document to the documents in a corpus and retrieving documents with high similarity scores.

This section will explore an algorithmic approach for retrieving similar documents from a corpus based on a query document. Additionally, we will discuss the initialization of the algorithm's parameters.

13.1 Initialization of the Parameters

Before discussing the algorithm for fetching similar documents, let's understand the initialization of the parameters used in the algorithm:

- *queryDocument*: The query document is to be compared with the documents in the corpus.

- *threshold*: This is the average value of all the similarity scores across the corpus.
- *doubtThreshold*: A threshold value is used for doubt entries. A document with a similarity score between the threshold and the doubt threshold is not considered similar or dissimilar. Rather it is put on the doubtful list.
- *similarDocsForQuery*: A dictionary to store the similar documents retrieved with their similarity scores.
- *visited*: A dictionary to keep track of the documents that have already been visited.
- *positiveQueue*: A priority queue to store the documents with high similarity scores. And hence they are more probable to be relevant.
- *doubtQueue*: A priority queue to store the documents with similarity scores between the threshold and the doubt threshold.
- *corpus*: The corpus of documents.
- *maxCorpusExploration*: A value to limit the number of documents to explore in the corpus. This was set to 0.8 in our case.
- *d*: A variable to store the current document being processed.
- *simScore*: The similarity score of the current document with the query document.
- *fixedFetch*: The fixed number of documents to fetch for an obtained similar document. This was set to 5 in our case.

13.2 Algorithm for Fetching Similar Documents

The algorithm is used to fetch similar documents from the corpus based on the query document. Here are the steps to execute the algorithm:

1. Initialize the above parameters based on the corpus and the query document.
2. Start a while loop until the number of visited documents is less than the corpus size multiplied by corpus explore.
3. Fetch the top fixedFetch documents from the corpus based on the similarity score with the query document.
4. Check if the similarity score is greater than the threshold. If it is, store the document and its similarity score in the similarDocsForQuery dictionary.
5. For each document in the fetched documents, check if it has been visited before. If not, calculate the similarity score with the query document and add it to the positive queue.
6. Mark the current document as visited.
7. If the similarity score is less than or equal to the threshold but greater than the doubt threshold, add the document to the doubt queue.
8. If the similarity score is less than or equal to the doubt threshold, mark the fetched documents as visited and continue to the next document.
9. If the positive queue is not empty, get the document with the highest similarity score from the queue, and calculate its similarity score with the query.

Algorithm 5 Fetch Similar Documents

Require: $queryDocument$, $threshold$, $doubtThreshold$, $fixedFetch$, $corpus$, $maxCorpusExploration$, $similarDocsForQuery$

```
d ← select document from corpus for exploration
visited[d] ← True
simScore ← cosineSimilarity(d, queryDocument)
while len(visited) < len(corpus) ∗ maxCorpusExploration do
    kList ← Fetch top fixedFetch similar docs of d from corpus
    if simScore > threshold then
        similarDocsForQuery[d] ← simScore
        for k ∈ kList do
            if k ∉ visited then
                if k ∉ positiveQueue then
                    positiveQueue.ADD(cosineSimilarity(k, queryDocument), k)
                end if
                visited[d] ← True
            end if
        end for
    else
        if simScore > doubtThreshold ∗ threshold then
            for k ∈ kList do
                if k ∉ visited then
                    doubtQueue.ADD(scoreParent(k), k)
                end if
            end for
        else
            for k ∈ kList do
                visited[k] ← True
            end for
        end if
    end if
    if positiveQueue is not empty then
        d ← positiveQueue.GET()
        simScore ← scoreParent(d)
    else if doubtQueue.empty() is not empty then
        d ← doubtQueue.GET()
        simScore ← cosineSimilarity(d, queryDocument)
    else
        d ← select next document from corpus
        simScore ← cosineSimilarity(d, queryDocument)
    end if
    visited[d] ← True
end while
return similarDocsForQuery reverse-sorted by key = simScore
```

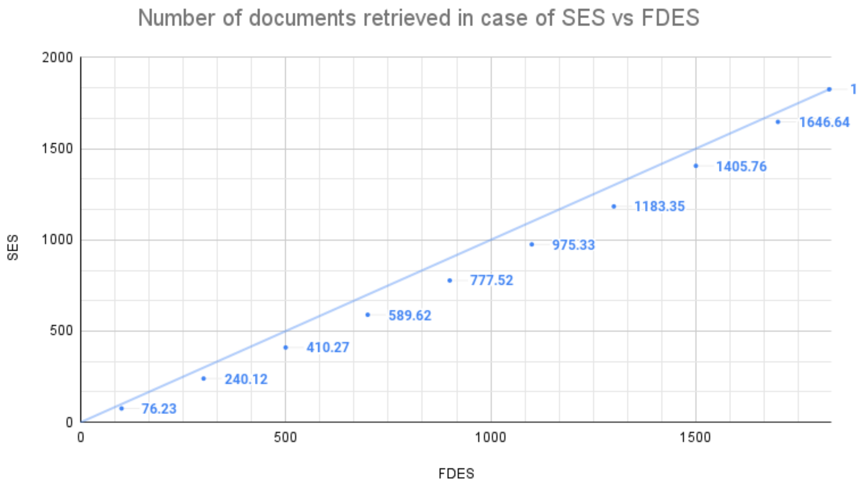


FIGURE 13.1: Number of documents retrieved in case of Summarization exhaustive search vs Full document exhaustive search

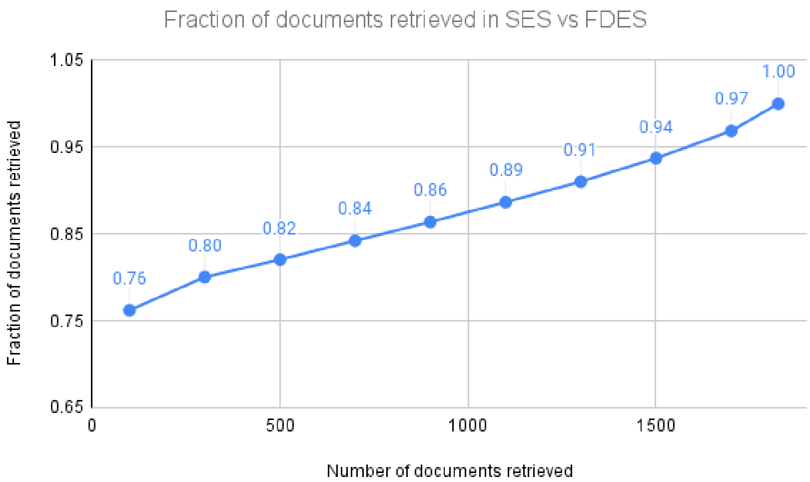


FIGURE 13.2: Fraction of documents retrieved in Summarization exhaustive search vs Full document exhaustive search

13.3 Results

The plots show that a comparable number of documents are retrieved in the case of a Summarization exhaustive search as compared to a Full document exhaustive search. Meaning that summaries can be used to evaluate the above algorithm. We have taken 50% LexRank summarized summaries in this case.

Chapter 14

Triad relationships

14.1 Nodes established in the triad

14.1.1 Role Labelling

1. Rule-based approaches were applied but they could not differentiate between argument(ARG), the ratio of decision(Ratio), and rulings by the present(RPC) and lower courts(RLC).

2. LDA Clustering was applied to legal documents for seven clusters and, they were qualitatively similar to role labelling-based labels. We have qualitatively established that 7 clusters are optimal based on the overlap between clusters. In an ablation study, we noted the amount of inter-cluster overlap across different numbers of clusters.

14.1.2 Summary

Unsupervised legal summary has been studied in a literature survey of domain-specific and domain-independent summarizers. We have addressed the difficulty in annotation and are trying to study summarizers giving comparable accuracy by unsupervised means. We have introduced a Knapsack-based algorithm for the same. We have taken Cluster summaries in union with LexRank Summaries and performed Knapsack. In cases where a summary is present, it compares reasonably well with other summarizers which are domain specific. We have also introduced a new metric called the Top - K Analysis metric which is also consistent with other metrics (Kendall's tau, Spearman's correlation, etc). Next, we have shown how this summary consistently scales up.

14.1.3 Similarity

A dynamic similarity algorithm has been introduced, the Top - K score analysis metric has been defined and its performance has been validated on 50 document corpus: by ROUGE score implicitly. It has been explicitly validated on a 500, 1000, and 2000 document corpus using Kendall's Tau and Spearman correlation.

14.2 Edges established in the triad

14.2.1 Similarity - Summarization

We have shown how better summaries give better similarity scores and reduce the computation time and space. We performed an analysis of LexRank vs LSA vs DelSumm based on the proportion of the same documents predicted for top K results to establish the same.

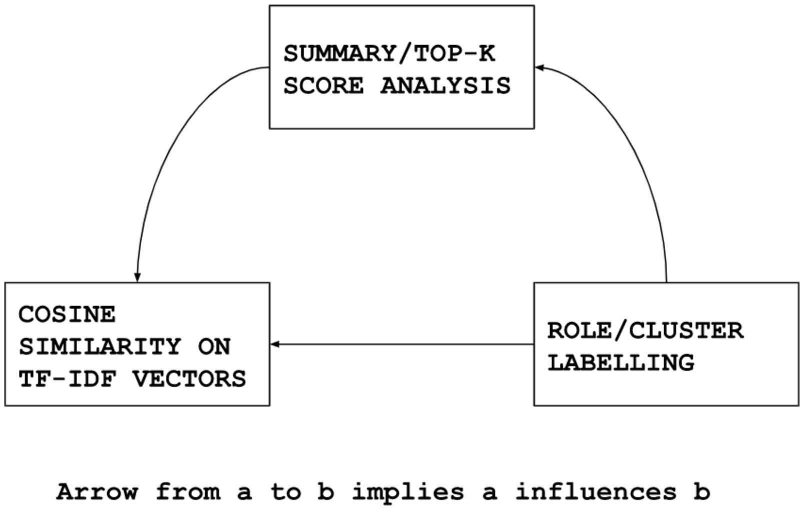


Figure 14.1: Established triad relationships

14.2.2 Summary - Role labelling

Knapsack summary performs better than LexRank summarizers. Hence we have established that role labelling gives a better summary. The length for best ROUGE-score on 50 document corpus was 1.25 times expert summary length 1.25*40% = 50% length. This was shown in a secondary ablation study.

14.2.3 Summary - Similarity

Union summaries (alternatively called Knapsack summaries) also perform better on our new proposed Top-K score analysis metric: 50, 500, 1000, and 2000 document corpus shows that better summaries give a better similarity score.

Hence all the nodes are now studied and the dependence can be shown in Figure 13.1.

Chapter 15

Future scope

The future scope of research, as well as the research questions, are as follows:

1) To explore the visual summary data structure in more detail and to add more features.

2) To implement a dynamic similarity algorithm while forming a similar document retrieval list and to perform an ablation study on the same.

3) To implement a structured case display based on a concept graph and to retrieve the similarity of documents based on different clusters/roles.

4) To establish entailment relationships in the document flow concerning role labels.

5) Perform a quantitative ablation study on clustering and its effect on the Knapsack summarizer.

Appendix A

Similar document retrieval on summaries and complete documents

Performance of similar document retrieval on summaries and complete documents: Below are the handpicked sentences chosen to decide the similarity relevance of each document pair manually, shown for two example files

Dataset: 505 document from SC corpus

A.1 1.txt

List of top 100 docs based on complete documents:

1, 10619, 10676, 10854, 10241, 10, 10864, 10281, 10514, 10702, 10166, 10517, 10342, 10759, 10276, 10779, 10484, 10711, 10618, 10272, 10182, 1078, 10551, 10801, 10739, 10239, 10769, 10855, 10719, 10288, 10744, 10670, 10568, 10217, 10313, 10786, 10734, 1081, 10646, 10841, 10710, 10366, 1082, 10327, 10455, 10625, 10424, 10357, 10572, 10575, 10488, 10651, 10537, 10411, 10859, 10168, 10745, 10114, 10736, 10748, 10733, 10720, 10176, 10362, 10775, 10765, 10541, 10414, 10321, 10331, 10632, 10839, 10267, 10291, 10673, 10333, 10211, 10505, 10771, 10295, 10459, 10764, 10792, 10250, 10866, 10593, 10545, 1085, 10215, 10181, 10587, 10493, 10674, 10595, 1077, 10789, 10364, 10358, 10542, 10534

List of top 100 docs based on summaries:

1, 10320, 10, 10619, 10748, 10517, 10424, 10676, 10739, 10759, 10769, 10625, 10568, 10360, 10276, 10514, 10789, 10861, 10327, 10779, 10646, 10256, 10541, 10247, 10319, 10217, 10462, 10182, 10551, 10168, 10272, 1074, 10618, 10841, 10295, 1082, 10816, 10265, 10747, 10207, 10411, 10854, 10757, 10839, 10622, 10450, 10651, 10281, 10665, 10764, 10176, 10744, 10808, 10641, 10297, 10630, 10598, 10666, 10834, 10552, 10307, 10628, 10702, 10267, 10455, 10433, 10299, 10477, 10806, 10617, 10486, 10432, 10482, 10181, 10790, 10175, 10259, 10754, 10775, 1078, 10387, 10389, 10742, 10862, 10582, 10166, 10238, 10781, 10483, 10845, 10173, 10565, 10410, 10579, 10241, 10480, 10488, 10588, 10736, 10842

For file '1.txt'

The appellant owns two sugar mills. There was a dispute between the appellant and its workmen with respect to the employment of contract labour in the two mills. that the industrial court had no jurisdiction to decide the dispute as the matter was covered by item (6) of Sch. III of the Act, which is within the exclusive jurisdiction of a labour court; and (ii) that any award directing abolition of contract labour would contravene the fundamental right of the appellant to carry on business under Art. 19(1)(g) of the Constitution. matter goes back before the industrial court as directed by the appellate tribunal

10619.txt

Appellants in their counter had pleaded that since they had been put to a loss of Rs.43,16,400/-; Claim by the respondent: a sum of Rs.66,09,669.36 was outstanding and had been wrongly withheld payment of Rs.23 lacs questions open inclusive of the claim to interest on all payments made delayedly to be settled before the civil court

10676.txt

Writ petitions highlight the faulty manner in which reservations have been provided and implemented by the Government of Uttar Pradesh and its authorities in the matter of admission to medical courses for the year 1994-95.

10854.txt

The appeal is preferred against an order of the Delhi High Court dismissing the writ petition filed by the appellant at the stage of admission. The appellant, Major General IPS Dewan, is aggrieved with, what he says, denial of promotion to the rank of Lt. General. He says that though he was the senior-most of the several candidates considered for promotion and his record of service was the best of all, he was not promoted because of and only on account of the adverse remarks made by Gen.S.F.Rodrigues, Chief of the Army Staff against him on 11th May, 1993. attack should fail for the reason that the memo containing adverse remarks in this case does set out the particulars in support of the same

10320.txt

conflict of opinion among the High Courts on the meaning and interpretation or clauses (i) and (ii) of sub-section (1) of Section 64 (as they stood prior to 1st April, 1976) of the Income Tax Act, 1961 The learned Judge also rejected the attack upon the constitutionality of the said provision based on Article 19(1) (g) The Income-tax Tribunal or the other concerned authorities under the Act, as the case may be, shall pass orders in each of these individual cases in accordance with the above legal position

10.txt

The respondents dealt in gunnies. They first entered into contracts with two Mills agreeing to purchase gunnies at a certain rate for future delivery, and also entered into agreement with third parties, by which they charged something extra from those third parties and handed over the delivery order known as kutcha delivery order. The Mills however did not accept the third parties as contracting parties, but only as the agents of the appellants and delivered the goods against the kutcha delivery orders, and collected the Sales Tax from the third parties.

10748.txt

These appeals pertain to the land pertaining to the respondents which has been acquired pursuant to Notification dated 30.1.1973 under Section 4 of the Land Acquisition Act, 1894 and Notification dated 24.7.1973 under Section 6 of the Land Acquisition Act,1894. Land Acquisition Collector by his award dated 5.9.1973 awarded compensation to the respondents for 'A' Category of land at the rate of Rs. 20,000/- per acre and for 'B' Category at the rate of Rs. 12,000/- per acre. enhanced the compensation amount and also granted the benefit of Sections 23(1A), 23(2) and 28 of the amended Land Acquisition Act to the respondents.

10241.txt

The respondent had retired from the post of Director, Institute of Animal Health & Veterinary Biological Products, Mhow on 31st July, 1983. His retiral benefits had been sanctioned by the appropriate authority. :: pension

10517.txt

prayer clause:: (a). The annual alcohol quota of 14.40 lakh bulk litres be released to the 1st petitioner, on compliance of the statutory provisions, within two weeks

A.2 10.txt

for complete doc TF-IDF

10, 10514, 10276, 10619, 10424, 10625, 10702, 10748, 10789, 10676, 10670, 10551, 10786, 10769, 10711, 1082, 1, 10739, 10256, 10841, 10327, 10855, 10447, 10568, 10299, 10320, 10490, 10720, 10779, 10681, 10667, 10319, 10360, 10646, 10433, 10389, 10410, 10640, 10230, 10166, 10295, 10326, 1077, 10324, 10204, 10697, 10704, 10486, 10190, 10798, 10296, 10574, 10176, 10593, 10701, 10534, 10677, 10376, 10461, 10174, 10790, 10318, 10355, 10280, 10450, 10859, 10842, 10648, 10726, 10250, 10765, 10572, 10862, 10224, 10366, 10622, 10856, 1079, 10806, 10512, 10114, 10733, 10505, 10181, 10183, 10610, 10590, 10836, 10387, 10448, 10341, 10333, 10414, 10368, 10787, 10337, 10521, 10429, 10493, 10541

for summary TF-IDF

10, 10320, 10357, 1, 10651, 10619, 10625, 10514, 10676, 10168, 10739, 10424, 10551, 10646, 10568, 10517, 10327, 10855, 10794, 10360, 10616, 10281, 10572, 10217, 10319, 10841, 10779, 10166, 10734, 10272, 10802, 10181, 10534, 10854, 10295, 10666, 10702, 10789, 1085, 10182, 10457, 10790, 10238, 10744, 10780, 10265, 10808, 10432, 10771, 1075, 10355, 10276, 10798, 10689, 10411, 10433, 10670, 10711, 10769, 10167, 10701, 10256, 10398, 10842, 10748, 10726, 10719, 10593, 10333, 10582, 10815, 10806, 1078, 10448, 10216, 10574, 10727, 10450, 10640, 10219, 10638, 10455, 10680, 10447, 10361, 10401, 1082, 10542, 10414, 10786, 10653, 10454, 10343, 10714, 10429, 10299, 10207, 10247, 10413, 10211

10.txt

The tax authorities treated these transactions between the appellant and the third parties as fresh sales and sought to levy sales-tax again, which the appellants contended, was not demandable as there were no second sales; the delivery of a kutcha delivery order did not amount to a sale of goods, but was only an assignment of a right to obtain delivery of gunnies which were not in existence and not appropriated to the contract There being two separate transactions of sale, one between the Mills and the original purchasers and the other between the original purchasers and third parties, tax was payable at both the points. s. 2(4) of the Sale of Goods Act In our opinion, there being two separate transactions of sale, tax was payable at both the points, as has been correctly pointed out by the tax authorities and the High Court.

10357.txt

question, viz., whether and, if so, in what circumstances, a medical practitioner can be regarded as rendering 'service' under Section 2(1)(o) of the Consumer Protection Act, 1986

10651.txt

(i) whether a person who is subject to the Army Act, 1950 ('Act' for short) can be dismissed from service for committing an offence under the Act even after he had retired on attaining the age of superannuation? and (ii) whether a Junior Commissioned Officer of the Indian Army who has to his credit the minimum period of qualifying service required to earn a pension or gratuity is eligible for the same in case he is dismissed from service under the provisions of the Act?

10514.txt

This is tenant's appeal. Respondent-landlord filed eviction-petition against the appellant under Section 10(2) (i) of the Tamil Nadu Buildings (Lease and Rent Control) Act, 1960 (the Act) on the ground that the tenant committed willful default in the payment of rent

10276.txt

Jairatum Bibi, wife of Jakim Ansari, residing in No.57, Sanjay Amar Colony is the legal representative of Saidur and Rabia. Delhi Administration has no objection to pay Rs.10,000/- towards compensation for the life of each of the two deceased to the sole legal representative daughter, Jairatun Bibi

10424.txt

meaning of the word 'each' in the expression "if the permission has not been granted by the stock exchange or each such stock exchange" used in sub-section (1A) of Section 73 of the companies Act, 1956

10625.txt

claimants are not entitled to the enhanced benefits under Sections 23(1-A), 23(2) and 28 of the Land Acquisition Act, 1894.

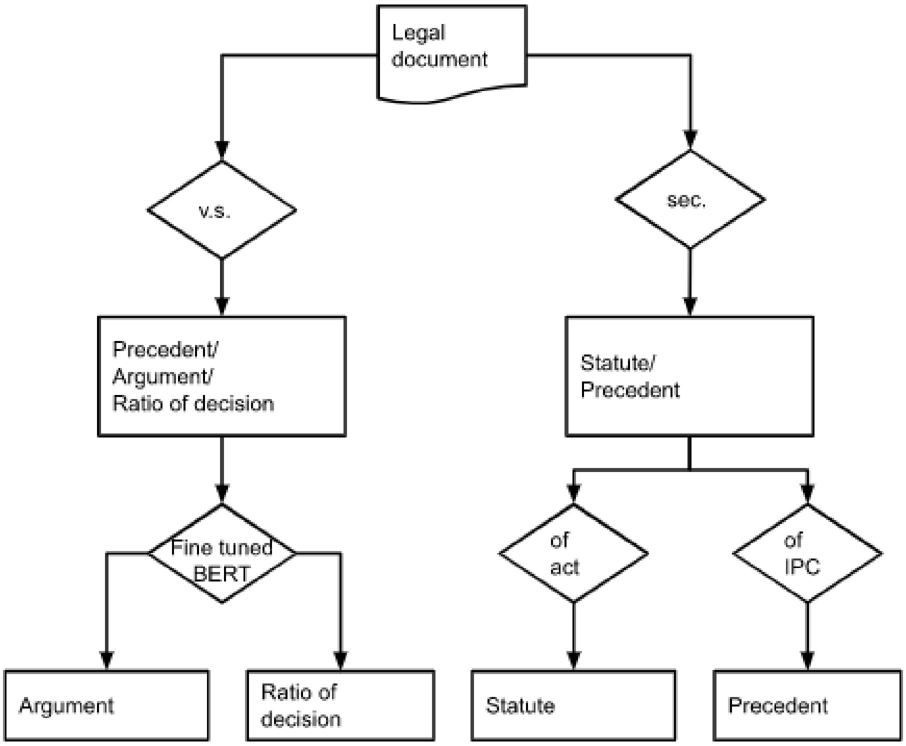


FIGURE B.1: BERT to distinguish ARG from Ratio

Appendix B

BERT for classification of Argument and Ratio of Decision

Figure B.1 shows the flowchart of the classification. Ratio was given a label 1 and ARG a label 0. Then BERT was fine tuned on the corpus of 50 expert role labelled documents. The train test split ratio was 90 to 10. A final accuracy of 87% is achieved.

```
True Positive = 22
True Negative = 4
False Positive = 4
False Negative = 1
Precision = 22/26 = 0.8461
Recall = 22/23 = 0.9565
F1 score = 44/49 = 0.8979
Accuracy = 0.87
```

Appendix C

Legal stop words using PageRank

PageRank was used to find prominent topics in legal documents and create a list of legal stop words. This gives more intuitive results as opposed to simple frequency based methods.

The list is as below:

court - 44.62161715648352

act - 29.077784519714914

case - 28.605137655045016

state - 27.519360182231722

respondent - 26.96237406771334

section - 26.62891669284112

appellant - 23.071582303988148

order - 21.048250585900444

™ - 19.713119640988733

government - 18.839190611482273

high - 18.205334990076594

date - 16.582155186213438

petitioner - 16.275719210717227

service - 15.686540156676319

decision - 14.783293358326805

time - 14.701854548715966

rules - 14.32726726016479

shri - 13.993350800258323

bench - 13.456791499108103

petition - 13.150559143643834

. - 13.087992599897074

india - 13.062885100651911

appeal - 12.992583057573635

question - 12.686391543722202

law - 12.41290240387716

respondents - 12.111225077281167

singh - 11.895340475545286

appellants - 11.704336195702439

basis - 11.600936619554775

tax - 11.556951577833944

election - 11.425570267962023

view - 11.352555880443235

11.271912853977653 judgment - 11.101333859418853

s - 10.895586117171622

land - 10.84774868975277

person - 10.78978742942055

rule - 10.308870469023947

purpose - 10.181929838675831

period - 10.075733319195846

interest - 10.002411352258417

ors - 9.872111037644533

appointment - 9.829007199013047

evidence - 9.765618909605774

article - 9.720038306369304

right - 9.588311379823963

counsel - 9.587000237705615

constitution - 9.486724248385599

matter - 9.310352048255307

officer - 9.258356895904878

sub - 9.246448079869902

post - 9.23889803091225

mr. - 8.99094081357754

respect - 8.935428987212065

fact - 8.915996442881616

provisions - 8.832783151759285

persons - 8.542187471009704

clause - 8.425726772854869

selection - 8.379504753205865

â€ - 8.245395540296055

candidate - 8.203054844594664

payment - 8.201901058230362

effect - 7.735183440344509

duty - 7.601028565417901

reference - 7.554160861832659

scc - 7.550667125882117

parties - 6.702786091218306

years - 6.687915627174109

Appendix D

Summarisation - Similarity analysis

K-scores analysis for summaries: Number of documents common in top-K documents for the document 2015-S-368

Table D.1: Summarisation - Similarity analysis of Knapsack based summary for document 2015-S-368

Value of K	LexRank	Knapsack
K = 5	2	3
K = 10	7	5
K = 20	14	17
K = 30	26	27
K = 40	35	37

Full document outputs:

2015-S-368, 1989-A-55, 1954-M-25, 1971-S-1, 2009-B-16, 1973-S-68, 1953-L-1, 1987-M-123, 1963-S-59, 2010-S-431, 2011-S-308, 2011-I-16, 2004-C-129, 2005-S-388, 2001-S-1131, 2007-S-608, 2001-A-234, 2008-P-8, 1987-C-108, 2007-S-632, 2008-A-260, 2008-C-166, 1987-S-26, 2008-I-54, 2010-J-55, 2007-B-76, 2008-S-1411, 2007-C-121, 2012-S-270, 2007-U-18, 2015-J-10, 2004-I-24, 1953-S-23, 2006-A-136, 2014-J-33, 1996-T-169, 1978-M-13, 1994-M-69, 2014-R-41, 1977-P-19, 2009-S-146, 2008-S-549, 1980-W-3, 2000-C-151, 1976-T-9, 1994-S-246, 2000-V-80, 1995-S-317, 2006-A-36, 1996-B-72

LexRank outputs:

2015-S-368, 1989-A-55, 1953-L-1, 1973-S-68, 2005-S-388, 1987-C-108, 1971-S-1, 2011-S-308, 2010-S-431, 1987-M-123, 2015-J-10, 1954-M-25, 2008-S-1411, 2008-S-549, 1987-S-26, 2001-A-234, 1963-S-59, 1977-P-19, 2007-C-121, 2011-I-16, 1953-S-23, 2007-U-18, 2010-J-55, 2009-B-16, 2001-S-1131, 2007-S-632, 2008-I-54, 2004-C-129, 2008-A-260, 2008-P-8, 1976-T-9, 2004-I-24, 2009-S-146, 2012-S-270, 2007-B-76, 2000-C-151, 2014-J-33, 2014-R-41, 1980-W-3, 1995-S-317, 1978-M-13, 2007-S-608, 2006-A-136, 1994-M-69, 2000-V-80, 2008-C-166, 2006-A-36, 1996-T-169, 1996-B-72, 1994-S-246

Knapsack outputs:

2015-S-368, 1989-A-55, 1954-M-25, 1987-C-108, 2005-S-388, 2011-I-16, 2007-S-632, 2011-S-308, 1987-M-123, 1971-S-1, 1973-S-68, 2010-S-431, 2008-I-54, 2008-A-260, 1963-S-59, 2001-A-234, 2010-J-55, 1953-L-1, 2001-S-1131, 2008-P-8, 2004-C-129, 2009-B-16, 2015-J-10, 2008-S-1411, 2007-S-608, 2004-I-24, 1987-S-26, 2012-S-270, 1977-P-19, 2007-U-18, 1978-M-13, 2008-S-549, 2009-S-146, 2007-C-121, 1996-T-169, 2007-B-76, 1953-S-23, 2006-A-136, 2000-C-151, 2014-J-33, 1995-S-317, 2000-V-80, 2006-A-36, 1994-M-69, 2008-C-166, 2014-R-41, 1976-T-9, 1980-W-3, 1994-S-246, 1996-B-72

Conclusion: Hence the proposed knapsack based summarizer improves similarity values. This result is in concordance with the triad relationship. We conclude empirically that better summarizers lead to better similarity ranking.

Bibliography

Bhattacharya, P., Ghosh, K., Pal, A., and Ghosh, S. (2020). Hier-spcnet: A legal statute hierarchy-based heterogeneous network for computing legal case document similarity. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1657–1660, New York, NY, USA. Association for Computing Machinery.

Bhattacharya, P., Paul, S., Ghosh, K., Ghosh, S., and Wyner, A. (2019a). Identification of rhetorical roles of sentences in indian legal judgments.

Bhattacharya, P., Paul, S., Ghosh, K., Ghosh, S., and Wyner, A. (2019b). Identification of rhetorical roles of sentences in indian legal judgments. In Araszkiewicz, M. and Rodríguez-Doncel, V., editors, *Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference, Madrid, Spain, December 11-13, 2019*, volume 322 of *Frontiers in Artificial Intelligence and Applications*, pages 3–12. IOS Press.

Bhattacharya, P., Poddar, S., Rudra, K., Ghosh, K., and Ghosh, S. (2021a). Incorporating Domain Knowledge for Extractive Summarization of Legal Case Documents. In *Proceedings of the 18th International Conference on Artificial Intelligence and Law (ICAIL).*

Bhattacharya, P., Poddar, S., Rudra, K., Ghosh, K., and Ghosh, S. (2021b). Incorporating domain knowledge for extractive summarization of legal case documents. *CoRR*, abs/2106.15876.

Mandal, A., Ghosh, K., Ghosh, S., and Mandal, S. (2021). Unsupervised approaches for measuring textual similarity between legal court case reports. *Artificial Intelligence and Law*, 29:1–35.

Pandey, S., Chandra, A., Sarkar, S., and Shankar, U. (2021). *Towards Reducing the Pendency of Cases at Court: Automated Case Analysis of Supreme Court Judgments in India*.

Wagh, R. and Anand, D. (2020). Legal document similarity: a multi-criteria decision-making perspective. *PeerJ Computer Science*, 6:e262.