IL-PCSR: Legal Corpus for Prior Case and Statute Retrieval

Anonymous ACL submission

Abstract

Identifying relevant statutes and prior cases/precedents for a given legal case are two of the most common tasks exercised by legal practitioners. Researchers till date have addressed the two tasks independently, thus developing completely different datasets and models for each of the task, making it difficult to compare models across both tasks despite both being legal document retrieval problems. Given the paucity of such corpora, in this resource paper, we propose a new corpus **IL-PCSR** (Indian Legal corpus for Prior Case and Statute Retrieval), which is a unique corpus that provides a common testbed for developing models for both the tasks (Statue Retrieval and Precedent Retrieval). We experiment extensively with several baseline models on the tasks, including lexical models and semantic models. Results show that the ensemble of a semantic model (GNN) and a lexical model (BM25) gives the best performance.

1 Introduction

In the legal domain, laws and prior cases are considered to be the fundamental sources of knowledge that guide principles of jurisdiction (Joshi et al., 2023). In practice, a legal practitioner when faced with a legal case, typically uses their experience and knowledge to identify prior precedents and applicable statutes in the given situation. It can be a time-consuming activity. The problem gets worse with the growing number of legal cases in populous countries like India (Malik et al., 2021). Hence, there is an imminent need to automate the process to make it fast and more efficient.

Two tasks have been proposed independently in this regard: Legal Statute Retrieval (LSR) (that aims to identify statutes that are applicable in a given query case) (Paul et al., 2024) and Prior Case Retrieval (PCR) (that aims to identify relevant prior cases/precedents that should be cite in the given query case) (Joshi et al., 2023). Traditionally, the two tasks have been modeled separately leading to creation of different corpora and models for each of them. It makes it difficult to compare models across tasks. However, both the tasks are essentially legal document retrieval tasks, hence it would be interesting to explore if it is possible to solve each of the task using the same model architecture. This requires a common corpus for both the tasks having query cases (with missing precedent citations and missing statutes) and corresponding pool of candidate prior cases and pool of candidate statutes. We address this gap in this paper. In a nutshell, we make the following contributions:

- In this resource paper, we propose **IL-PCSR** (Indian Legal Corpus for Prior Case and Statute Retrieval) a large corpus of query cases along with candidate pool of prior cases and statues in English for the Indian legal setting. To the best of our knowledge, we are first to develop such a corpus having both prior cases and statutes for the same queries.
- We experiment with a variety of models to solve the tasks of LSR and PCR, including semantic models (e.g., transformer-based models and GNNs) and lexical models (e.g., BM-25).
- Experiments and analyses on the corpus brings out some interesting observations – lexical methods perform well in one task (PCR) while semantic methods perform well in the other (LSR). A probable reason for this surprising observation is the difference in the nature of the two tasks, such as, statutes are short and abstract, and the overall semantics of the query help to identify the relevant statutes. However, precedents are long and have similar language as the query (allows better lexical matches). Moreover, different portions of the query text (that have different local semantics) match accurately with different precedents. For these reasons, multi-task approaches (solving both the tasks simultaneously) also do not show much improvement (§5).

- Based on experimental observations, we propose an ensemble-based approach for combining scores coming from semantic and lexical methods respectively. These techniques use a linear interpolation of semantic and lexical scores, with a parameter α controlling the weightage between the two scores. We derive the α value by grid-search. Using this simple ensemble technique, we are able to obtain gains over both the individual methods. This observation holds true for both statute and precedent retrieval, across different combinations of lexical and semantic approaches.
- We further experiment with datasets from other jurisdictions. We experiment with the **COLIEE** datasets, specifically `COLIEE 2024 Task 3` (Statute Retrieval on Japanese law) and `COLIEE 2024 Task 1` (Precedent Retrieval on Canadian Federal cases). We verify that the same trends as **IL-PCSR** hold for **COLIEE**, and that the ensemble approach provides an improvement here as well.
- The **IL-PCSR** corpus, along with all model implementations, will be released publicly upon acceptance. For now, we make a small sample of the dataset available anonymously.[1]

2 Related Work

LSR and PCR are fundamental tasks in legal document processing and several research works have been done in this area, for example, techniques based on n-grams and features (Salton and Buckley, 1988; Zeng et al., 2007), doc embeddings (Le and Mikolov, 2014), transformer-based (Vold and Conrad, 2021) and among others (Salton et al., 1975; Robertson et al., 2009; Liu et al., 2023; Hofmann et al., 2013; Wang et al., 2018; Ma et al., 2022). Various works have been done for identifying legal statues (Wang et al., 2018, 2019; Chalkidis et al., 2019; Zhong et al., 2018; Wu et al., 2023; Paul et al., 2024). Similarly, various works have focused on prior case retrieval (Rabelo et al., 2022; Joshi et al., 2023; Tang et al., 2024b,c; Qin et al., 2024; Bhattacharya et al., 2020; Ma et al., 2021a, 2024). Due to space limitations, we provide more details about above-mentioned works in App. A. To the best of our knowledge, the two tasks have only been attempted independently making it difficult to compare models across the two tasks. In this paper, we are the first ones to provide a common corpus that provides an opportunity to develop a common model architecture for both the tasks, since both the tasks are legal document retrieval tasks.

[1] https://anonymous.4open.science/r/AnonymousSampleDataset-67B6/readme.md

3 IL-PCSR Corpus and Tasks

Our dataset consists of three sets of text data: (i) **Statute Candidate Pool:** 936 Statutes – Articles/Sections of law from 92 Central (Federal) Acts of Government of India; (ii) **Precedent Candidate Pool:** 3,183 Prior Cases from the Supreme Court of India (SCI) and state-level High Courts of India (HCI); (iii) **Query Set:** 6,271 Case Judgment documents from SCI and HCI. The procedure of creating the dataset is described below.

Dataset Construction: We collected a corpus of $20K$ case judgments from the website `indiankanoon.org` (a reputed legal search engine in India) through their API service. Indian legal documents are in public domain and accessible to all. These documents were chosen such that these were prominent judgments (having cited large number of times by other documents) of the Supreme Court of Indian and various High Courts of Indian states, spanning a time frame of 1950–2019. This gives us an initial corpus of important case documents that have both spatial and temporal variance.

We cleaned the documents and remove very short and very long documents (<5 and >95 percentile as per number of tokens). For creating list of candidate statues and precedents, we considered only those statutes and cases that were cited at least a certain number of times (5 and 3 respectively). For the query set, we chose those cases that cite at least one candidate statute and two candidate precedents. We further added a set of statutes and precedents that are *not* cited by any query to the candidate pools, to conform to a real-world setting where there can be many non-relevant candidates (more details in App. B).

Final dataset (IL-PCSR): Above process resulted in a final statute pool of 936 statutes, precedent pool of 3,183 cases, and a query set of 6,271 queries. The query set is randomly divided into train/validation/test splits in the ratio of 80%:10%:10% (5021:627:627). Of the 936 statutes, 19 are not cited in any query, and 29 are cited in the test set but *not* in the train set (zero-shot candidates). Similarly, of the 3,183 precedents, 94 are not cited in any query, and 155 are zero-shot candidates.

Anonymization and Masking of Citations: We masked the portions of the query documents where statute/precedent citations occur, to prevent models from associating the queries with the statute and case titles. We also anonymized (using LegalNER

Dataset	Jurisdiction	Query Type	#Queries	#Statutes	Avg. Stat citations	#Precedents	Avg. Prec citations
ECHR2021	EU	Case facts	11478	66	1.78	-	-
FLA-CJO	China	Case facts	60k	321	3.81	-	-
CAIL'18	China	Case facts	2.6M	183	1.09	-	-
CAIL-Long	China	Case facts	229K	574	5.77	-	-
ILSI	India	Case facts	66K	100	3.78	-	-
COLIEE'24 Task 3	Japan	Law Questions	554	746	1.27	-	-
COLIEE'24 Task 1	Canada	Case Judgment	1678	-	-	5529	4.10
LeCard	China	Case Judgment	107	-	-	100	10.33
CAIL'19-SCM	China	Case Judgment	8964	-	-	2 per query	1.0
IL-PCR	India	Case Judgment	1182	-	-	7070	6.8
IL-PCSR (ours)	India	Case Judgment	6271	936	2.69	3183	3.87

Table 1: Comparison of **IL-PCSR** with other LSR and PCR datasets. All existing datasets are meant for either statute identification or precedent identification, but not both. I**IL-PCSR** is the first dataset for both tasks together. Average Stat (Prec) citations refers to the average number of gold-standard statutes (precedents) cited per query.

tool (Kalamkar et al., 2022a)) all documents with regard to person names to prevent ethnic/religious biases. We replaced the identified text portions with placeholders such as [SECTION], [ACT], [PRECEDENT] and [ENTITY].

Comparison with other Corpora: To the best of our knowledge, **IL-PCSR** is the first dataset for identification of *both* relevant statutes and precedents for the *same query*, where the query is a real case document. Table 1 compares existing datasets with **IL-PCSR**. Note that all prior datasets are meant for either statute retrieval (LSR) or precedent retrieval (PCR), but not both. Prior LSR datasets, especially those in English, have mostly worked with a limited candidate set of statutes. For instance, the ECHR2021 dataset (Chalkidis et al., 2021) consists of cases from the European Court of Human Rights, and a statute set of only 66 articles. Datasets from China have mostly used cases from China Judgment Online, and have usually involved more articles, like FLA-CJO (Luo et al., 2017) (321 criminal articles), CAIL2018 (Xiao et al., 2018) (183 criminal articles) and CAIL-Long (Xiao et al., 2021) (244 articles for Criminal law, 330 articles for Civil law). In the Indian setting, the ILSI dataset (Paul et al., 2022) consists of 100 articles from the Indian Penal Code. All these datasets are devised in a multi-label classification setup, where given a query case, the task is to predict whether each article is relevant or not. The **COLIEE** (Li et al., 2024) family of datasets contains a task on statute retrieval, where the objective is to return a ranked list of the candidates. Here, the candidate sets are fairly large (746 statutes from Japanese law in COLIEE 2024), but the queries are simple, short questions asking directly about the specific law, and not the details of real life cases. For PCR, in common law jurisdictions (including India, Canada, UK), cases cited from the query case are considered relevant. The **COLIEE** (Li et al., 2024) family of datasets consist of query and candidate cases from Canadian Federal law. Recently, Joshi et al. (2023) released a PCR dataset based on cases from the Supreme Court of India. For the Chinese jurisdiction (based on civil law), LeCard (Ma et al., 2021b) and CAIL2019-SCM (Xiao et al., 2019) were created by combining lexical retrievers like BM25 with human annotation.

Tasks Formulation: Identifying the statutes and precedents cited in a query case can both be modeled as retrieval problems, as follows. Given a query Q and a pool of candidate statutes $\mathcal{S} = \{S_1, S_2, \ldots, S_{|\mathcal{S}|}\}$, the task of statute retrieval involves retrieving the set of statutes $S(Q) \subset \mathcal{S}$ that are relevant for Q. Similarly, the task of precedent retrieval requires one to identify the set of relevant precedents $P(Q) \subset \mathcal{P}$, where $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$ is the pool of precedent candidates. Historically, statute identification has mostly been modeled as a multi-label classification problem, but it can also be framed as a retrieval task especially when the number of candidate statutes is large. Precedent identification has almost always been modeled as a retrieval/ranking task. It is important to note that the concept of *relevance* can be quite restrictive in the legal domain. For instance, given a case related to kidnapping, many statutes can be relevant. Still, the exact statute applied would depend on other factors, like, whether hurt was caused, or if hurt was the primary intention of the crime, etc. Similarly, for precedents, lawyers would like to cite cases that are relevant not only in terms of facts but also in terms of the desired solution. For instance, the defense and prosecution lawyers are likely to cite relevant cases with opposite outcomes. In most prior works, the queries are

taken as the case judgement full texts (Li et al., 2023b; Joshi et al., 2023), although some studies considered only the case fact portions as the queries (Li et al., 2023a). In this work, we consider the full case judgement texts as queries, since in the Indian case judgements, the fact portions are *not* marked, and use of automated methods to extract the facts often lead to errors (since facts are often interleaved with other types of information) (Bhattacharya et al., 2023; Malik et al., 2022; Kalamkar et al., 2022b). The gold standard precedent and statute sets for a certain query Q are usually considered as the set of precedents and statutes that have actually been cited from Q (these information are masked from the query text). We follow the same approach in this work.

4 Methods for Legal Retrieval

We experimented with an array of methods, supervised and unsupervised, lexical, semantic, and summary based methods.

Lexical Methods: We experimented with lexical methods based on BM25 (a strong baseline for legal retrieval (Joshi et al., 2023)).

Vanilla BM25: Ths is an advanced version of the TF-IDF algorithm (unsupervised), relying on lexical matches between n-grams of the query and candidate to generate scores (Robertson et al., 2009).

SpaCy Events + BM25: Prior works have demonstrated that both the queries and precedent candidates tend be long and noisy, with only small portions of text leading to a strong match. Joshi et al. (2023) used SpaCy to extract events (subject, action, object triplets) and filtered out only the sentences containing matching events from both queries and precedents, leading to a better match via BM25. We performed the same experiment.

LLM Events + BM25: We observed that SpaCy tends to over-generate events and is more noisy. To address this, we employed an LLM (`gemma-7b-it`) to extract important events. We passed definitions from the SALI (Standards Advancement for the Legal Industry) (`https://www.sali.org/`) ontology to guide the LLM in event generation. We observed that LLM events are fewer than SpaCy events, but the elements in the (subject, action, object) triplets are larger (entire phrases/clauses) than that of SpaCy (one/two words at maximum). These events were subsequently used to filter out sentences for BM-25 baseline.

Semantic Methods: Semantic methods involve deep-learning models trained on the train set. For all the fine-tuning methods, we try two settings: (i) Learning statutes and precedents separately, i.e., essentially two different models for LSR and PCR, and (ii) Performing LSR and PCR simultaneously in a multi-task learning environment.

SAILER: Li et al. (2023a) pre-trained a BERT-based model on legal documents to provide legal understanding by tasking the model to generate the reasoning and judgement of a case given the facts. This model was fine-tuned with the case retrieval objective on the COLIEE 2021 dataset (Li et al., 2023b). We used this model directly for inference and also tried fine-tuned (over our train-set) version.

Event-GNN: As an alternative approach, instead of SALI ontology, we used `GPT-4-turbo` for creating event triplets for a small number ($\sim$ 400) of documents and these triplets were subsequently used to fine-tune the `gemma-7b-it` model for creating triplets for the entire corpus (Li, 2023). Since the subjects and objects extracted from the triplets are overlapping in some cases, we constructed a *graph* out of the triplets, with each subject and object being the nodes, and edges labeled with the action connecting the nodes (see App. Fig. 2). To reduce sparsity, we also introduced a global node to represent the whole document and connected each other node with the global node. All texts (nodes, edges) are encoded using SentenceBERT (Reimers, 2019). We then employed a 2-layer Graph Attention Network (Tang et al., 2024a) and perform dot product over the global nodes to capture document-level similarity (between a query and candidate).

Para-GNN: Different paragraphs in court case documents can usually be categorized into functional parts / rhetorical roles (from a legal perspective), such as Facts of the case, Arguments by lawyers, Ruling by the judge, etc. (Bhattacharya et al., 2023). IndianKanoon provides the rhetorical role for each paragraph, which can be used directly. Thus, we construct a setup where for each query/candidate, we have a global node representing the entire document and nodes for each paragraphs, connected to the global node with the edge labeled with the rhetorical role. The texts are converted to embeddings using SentenceBERT, and then a two-layer Graph Attention Network is employed on top. Dot product is calculated over the global nodes only. For statutes, each different subsection is taken as individual paragraphs, and the rhetorical role of each such paragraph is set to 'None'.

Summary-based methods: Using full case documents as both queries and precedents not only leads to large computation overload, but also introduces additional challenges for semantic models (Tang et al., 2024c,a; Qin et al., 2024; Yue et al., 2024). Full cases contain a lot of details that act as noise for statute and precedent retrieval. Thus, we conducted summarization experiments to both reduce the noise and focus on the relevant contextual information. We used `GPT-4o-mini` (prompts in App. C.2) to summarize documents based on the two different retrieval needs.[2] On the precedent-side, we asked the LLM to focus on the legal rulings and findings of the case, which usually form the core reasons for future cases to cite these. On the query-side, through experimentation on the validation data, we found that it is difficult to generate one single coherent summary containing contextual information for both statute and precedent citations. This is because different portions of the query text provide the contextual information needed for statute retrieval as compared to precedent retrieval. Thus, for each query, we create two summaries, one focusing on statute retrieval and the other focusing on precedent retrieval. For statute-retrieval summaries, we asked the LLM to focus on the facts and legal issues of the case. For precedent-retrieval summaries, we asked the model to focus on the legal issues and arguments by lawyers, as well as findings of lower court (if any). For joint retrieval, we concatenated the statute-retrieval and precedent-retrieval summaries to create a larger summary.

Paragraph-level methods: Since, original query and candidate documents are organized in the form of paragraphs. We can run unsupervised methods directly at paragraph level for a query-candidate pair, and then aggregate the paragraph-level scores into a single document level score. Specifically, for a query Q having n paragraphs and a candidate C having m paragraph, we obtain a paragraph level $n \times m$ score matrix S. Then, we use two different aggregation measures: (i) *Max-All:* $\max(S)$, and (ii) *Max-Sum:* $\sum_{i=1}^{n} \max(S_i)$. We obtain the paragraph-level scores through BM25 2-gram and SAILER. We did not experiment with higher order n-grams since it was computationally prohibitive. Also, we could not fine-tune in the paragraph-level setup since we do not have the gold-standard information at the paragraph level.

[2]Note that we do not attempt to summarize the statute-texts, since statute-texts are usually short and well structured (as compared to case documents).

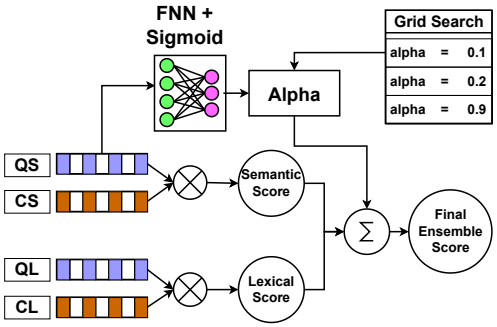


Figure 1: Pipeline of the Ensemble-based approaches. QS and CS represent query and candidate semantic embeddings, while QL and CL represent the query and candidate lexical embeddings.

Ensemble of Lexical & Semantic Methods: Combining lexical and semantic features can be beneficial to retrieval tasks (Bruch et al., 2023; Sumathy et al., 2016; Mandikal and Mooney, 2024). In the simplest setup (Figure 1), we can combine the prediction score assigned to a particular (query Q, candidate C) pair by a lexical method with that given by semantic methods. Formally, $\text{Score}(Q, C) = \alpha \times \text{Semantic Norm Score}(Q, C) + (1 - \alpha) \times \text{Lexical Norm Score}(Q, C)$. Here α is a hyper-parameter. Since score of each model has a different range, we used Z-score normalization across all candidates for each query. In the ensemble approaches, we used BM25 5-gram as the lexical method and Event-GNN or Para-GNN as the semantic method. We vary the values of $\alpha = \{0, 0.1, \ldots, 0.9, 1\}$ to determine the optimal value over the validation set.

Usage of LLMs in retrieval methods: It is difficult to use LLMs directly for retrieval purposes, since the model must be fed the query text as well as *all the candidate texts* in the same prompt window. This is not straight-forward in the legal domain since we have a large number of candidates, and the texts are quite lengthy. We instead use LLMs to restructure the input for downstream retrieval models, by creating events or summaries.

Setup and Hyperparameters: For BM25-based experiments, we used different n-grams ($n = 2, 3, 4, 5$). In case of the fine-tuning approaches, we use a contrastive learning setup, with BM25 hard negatives and in-batch negative sampling. We used AdamW (Loshchilov, 2017) optimizer and a linear scheduler (more details in App. C).

5 Experiments, Results and Analysis

The results of all methods are presented in Table 2 in terms of standard evaluation metrics macro-F1@k, MAP and MRR (details in App. C.3). We observe a stark difference in the performance of different methods for statutes and precedents. Methods based on BM25, which focus on lexical matches and hence can capture legal phrases or keywords, are very effective for precedent retrieval. Whereas, fine-tuned transformer models are able to perform better for statute retrieval than unsupervised lexical methods. The language used in statute descriptions is more formal than that used in case documents, and thus direct keyword matching can be difficult. This could explain why supervised fine-tuning is performing the best. Also, we observe that summarization leads to an improvement in performance only for statutes, but not precedents (compared to taking the full documents as input).

Performance over LSR: For LSR, we observe that the lexical approaches perform rather poorly. SpaCy event filtered sentences do not perform as well as the vanilla BM25, while using sentences filtered with LLM events performs slightly better. For both BM25 (2-gram) and SAILER (inference), the para-wise scoring approaches outperform the respective doc-wise approaches. The Max-All aggregation measure works much better than Max-Sum for both models. For supervised approaches, we observe that SAILER (fine-tune) performs significantly better than inference mode, whereas Para-GNN also performs similarly. The LLM events are able to summarize the queries effectively from the perspective of statute identification, and leads to high score during fine-tuning for Event-GNN. However, using summaries generated by LLMs seem to be even more effective, leading to the best performance on statute retrieval (with Para-GNN).

Performance over PCR: For PCR, we observe reverse trends, where the BM25-based lexical matching approaches significantly outperform the fine-tuning approaches. All the BM25 methods perform well, leaving SAILER and Event-GNN behind in terms of performance. The only semantic approach that is able to achieve relatively decent performance is Para-GNN, which effectively applies an attention network over all the paragraphs. We postulate this is possibly occurring since precedent matching, unlike statute matching, actually relies on matching small portions of the large query with small portions of the document. There can be different matching aspects of the case, with regard to different precedents that are ultimately cited. The generic fine-tuning methods are not able to generate representations of the query and precedent candidate that can capture all these aspects (except Para-GNN, to some extent). Also, summarization using LLMs does not improve PCR as much as LSR, leading to poorer performance compared to using full documents as input. PCR heavily depends on exact lexical matches over short contextual windows, and we suspect that the abstractive summaries generated by the LLMs is leading to some information loss. However, the drop in performance compared to using full documents is not too high. Finally, we also observe that para-level scoring improves the performance for both BM25 (2-gram) and SAILER (fine-tune) compared to doc-level scoring, with Max-All performing better than Max-Sum (same trends as observed for LSR).

LSR and PCR together: We wanted to investigate the possibility of identifying statutes and precedents together, due to the intrinsic relationship between the statutes and precedents cited in a query case. Accordingly, we attempt to fine-tune SAILER, Event-GNN and Para-GNN for both tasks under a multi-task learning setup (the rows marked 'fine-tune multi-task' in Table 2). However, it appears that the multi-task learning setup does *not* work as well as when we tune two different models (for the same approach) for LSR and PCR (the rows marked 'fine-tune separately' in Table 2). This trend is observed for all of SAILER, Event-GNN and Para-GNN. Although statute retrieval may actually improve in some cases (like Para-GNN), PCR performance always reduces in a multi-task environment. The reasons behind this anomaly can be attributed to the inherent differences between the LSR and PCR tasks. Moreover, we have already observed that semantic fine-tuning methods are better for LSR, since these models are better able to map real case incidents to the abstract statute definitions. On the other hand, PCR requires the query representation to capture all the different aspects that match it with the precedents (see the Section above). Possibly in the multi-task environment, models are likely to learn query representations better suited for LSR than PCR.

Results of Ensemble Models: The ensemble approaches achieve improvements for both statutes and precedents, compared to both the individual lexical and semantic methods. We also observe that the final ensemble performance greatly

Method	Setting	Statutes			Precedents		
		F1	MAP	MRR	F1	MAP	MRR
Lexical Methods							
Vanilla BM25	2-gram	13.15	14.54	32.36	24.89	32.41	47.21
	4-gram	17.06	19.13	40.88	30.54	40.24	54.65
	3-gram	17.80	19.39	41.74	32.21	43.44	57.52
	5-gram	16.98	17.88	40.08	33.29	43.98	58.55
Vanilla BM25 (para-wise)	2-gram, Max-All	18.59	21.82	44.32	27.87	38.15	52.32
	2-gram, Max-Sum	14.66	16.67	35.84	26.23	35.26	49.41
Spacy events + BM25	2-gram	12.65	14.48	32.08	24.91	33.02	48.22
	3-gram	10.67	11.93	28.12	28.31	37.22	52.70
	4-gram	10.18	10.47	26.40	28.28	35.30	50.98
	5-gram	9.78	9.52	25.28	26.99	33.71	50.22
LLM events + BM25	2-gram	13.08	14.47	32.49	24.55	31.78	46.10
	3-gram	16.84	18.76	40.60	29.47	38.59	52.97
	4-gram	17.45	18.92	41.34	32.61	42.99	57.39
	5-gram	16.41	17.35	39.33	33.29	43.43	58.14
Semantic Methods							
SAILER	inference	7.15	9.31	19.40	9.94	13.90	20.49
	fine-tune separately	21.69	28.62	45.73	12.64	17.93	25.85
	fine-tune multi-task	20.45	25.36	41.44	11.88	17.52	24.85
SAILER (para-wise)	inference, Max-All	13.11	14.70	31.49	19.37	25.71	38.84
	inference, Max-Sum	5.42	6.57	16.06	11.16	16.42	24.67
Event-GNN	fine-tune separately	28.67	38.69	58.39	12.08	15.91	22.18
Event-GNN	fine-tune multi-task	18.43	24.11	43.23	11.74	15.59	22.56
Para-GNN	fine-tune separately	20.72	28.54	46.06	24.54	33.07	45.01
Para-GNN	fine-tune multi-task	23.74	29.79	49.84	24.67	32.88	44.17
SAILER (summaries)	inference	5.48	7.66	16.86	10.21	14.80	22.82
SAILER (summaries)	fine-tune separately	23.49	31.42	50.25	15.00	20.43	27.72
Para-GNN (summaries)	fine-tune separately	32.85	44.03	62.51	22.60	29.49	39.22
Para-GNN (summaries)	fine-tune multi-task	31.81	43.17	64.08	22.69	29.25	39.19
Ensemble Methods							
Event-GNN + BM25	5-gram, Grid Search	33.87	45.17	67.26	34.45	43.32	58.76
Para-GNN + BM25	5-gram, Grid Search	28.10	36.14	59.57	**36.93**	**48.62**	**62.83**
Para-GNN (summaries) + BM25	5-gram, Grid Search	**36.17**	**48.64**	**70.49**	36.35	48.27	61.76

Table 2: Results (%) of Statute retrieval and Precedent retrieval. Metrics are macro-F1@k, MAP and MRR. Best value for each metric in boldface. Best values for individual methods (not ensemble methods) underlined.

relies on the semantic method's inherent performance. For instance, Event-GNN performs very well on statutes (28.67% F1) compared to Para-GNN (20.72% F1), and thus under the ensemble setting, there is much higher statute performance for Event-GNN (33.87% vs. 28.10% F1). The reverse trend can be seen for precedents, where Para-GNN inherently performs much better than Event-GNN, and consequently the ensemble with Para-GNN is superior. This also holds true when using summaries as the inputs. Using summaries leads to more improvement as compared to using full documents for LSR (32.85% vs. 20.72% F1), and thus the same trend is seen in the ensemble setting (36.17% vs. 28.10% F1). For PCR, both with and without ensemble settings, using the full documents lead to slightly better performance as compared to using summaries.

Effect of different α on ensemble methods: We investigate the optimal weightage to be assigned to the lexical and semantic methods to achieve an improvement. We calculated the F1 values for both ensemble methods, Event-GNN and Para-GNN (both with full doc and summaries as input), at different values of α. We also varied the n-grams being used for the BM25 method. We observe that, for both PCR and LSR, the optimal score is mostly achieved at a high value of α (usually $\alpha \geq 0.8$), indicating that greater weightage has to be placed on the semantic method, but BM25 values cannot be ignored either (details in App. D).

Effect of candidate frequencies and text lengths on ensemble methods: We further analyze the performance of the best-performing ensemble methods based on the frequency of candidates. We observe that LSR performance drops significantly for the rare statutes, but PCR performance remains largely unaffected by the frequency. This could be attributed to the fact that for LSR, the semantic model is more dominant (which is prone to per-

Model	Statutes		Precedents	
	IL-PCSR	COLIEE	IL-PCSR	COLIEE
BM25 (5-gram)	16.98	54.49	33.29	30.61
Event-GNN	28.67	-	12.08	11.48
Para-GNN	20.72	17.64	24.54	21.24
Para-GNN (summaries)		-		18.52
Para-GNN + BM25	28.10	55.96	**36.93**	**34.52**
Event-GNN + BM25	33.87	-	34.45	34.51
Para-GNN (summaries) + BM25	**36.17**	-	36.35	30.25

Table 3: Results of the best methods on **COLIEE** datasets compared to ILPCSR. All results are in terms of macro-F1@K. Event-GNN and summaries could not be run for **COLIEE** statutes since the queries are too small for meaningful events or summaries.

forming poorly over rare candidates), but for PCR, the unsupervised lexical model is dominant (details in App. E). We also study the effect of both query and candidate lengths (details in App. F). We again observe that LSR performance drops for longer queries and statutes. For PCR, performance drops for longer queries but remains unaffected by the length of the precedent.

Key Finding: We observed that Para-GNN (either with full document or summaries) + BM25 works best for both LSR and PCR. However, prediction in LSR and PCR done via a common model (e.g., via multi-tasking) does not perform as well as when the tasks are performed separately.

6 Experiments on COLIEE dataset

Till now, all our observations are drawn over the **IL-PCSR** dataset constructed in this work. We would like to verify whether the above trends are also seen on other legal datasets/jurisdictions. Since no current dataset allows the identification of both statutes and precedents together from the same query, we have to work with two separate datasets for LSR and PCR. We choose to work on the well-known COLIEE datasets (Li et al., 2024).

LSR dataset: We use the `COLIEE 2024 Task 3 (Statute Law Retrieval)` dataset consisting of 746 statutes from *Japanese law*, and 554 queries. Note that the queries here are typically one or two sentences long, asking specifically about the laws. In contrast, the queries in **IL-PCSR** are real-life cases, which makes the setting more practical and challenging. We opted for this dataset since other existing datasets in English (ECHR2021 and ILSI) have too less statutes (66 and 100 respectively) to be evaluated in the retrieval setup.

PCR dataset: We use the `COLIEE 2024 Task 1 (Legal Precedent Retrieval)` dataset consisting of 1,678 queries and 5,529 precedent candidates, all of which are real-life case judgments from *Canadian Federal law*. This setting is similar to the queries in **IL-PCSR**.

Results: We choose some of the methods that performed highly over the **IL-PCSR** dataset, and applied these methods on the COLIEE datasets. The results on **COLIEE** vis-a-vis **IL-PCSR** are presented in Table 3. The trends on the **COLIEE** dataset are almost similar to what we observed for **IL-PCSR**. The only difference being that, for **COLIEE**, even for LSR, lexical methods such as BM25 perform the best (whereas semantic methods outperformed lexical approaches for LSR in **IL-PCSR**). This difference is possibly because the queries of **COLIEE** are short sentences, asking directly about the statutes, whereas for **IL-PCSR** the queries are real-world long case judgments. Both for LSR and PCR, we see improvements when using an ensemble setup for **COLIEE** as well. The improvement is limited in case of statutes, possibly because the performance of Para-GNN is poor. This is possibly because short queries do not have enough structure for the GNN to exploit. For precedents, where the semantic methods perform better, the improvement obtained by ensembling is high. This agrees with the trend we see on **IL-PCSR** (see Table 3). Finally, for both **IL-PCSR** and **COLIEE**, in the case of PCR, we have observed that using summaries does not perform as well as using the full texts. We observe the same key findings as **IL-PCSR**.

7 Conclusion and Future Work

In this resource paper, we create a new corpus **IL-PCSR** that brings together PCR and LSI tasks for the first time. We experimented with a wide variety of methods for each of the task. Our experiments show that ensemble of lexical (BM-25) and deep semantic method (GNN) perform the best. However, modeling both the tasks in a multi-task setting degrades the performance a bit for one of the task. In future, we plan to work more with the ensembling techniques, including devising fine-tuning approaches to incorporate BM25 scores. We also plan to use multiple representations/features to construct the models, since we believe this will help in tuning statutes and precedents together. We also wish to explore the use of LLMs to better exploit the inherent connections between statutes and precedents cited from a given query.

Limitations

In this paper, we conduct a thorough research into the relationship between legal statutes and precedents. Specifically, we have made the first attempt (to the best of our knowledge) to solve the tasks of LSR and PCR simultaneously from the same query. All the prior works have either taken isolated approaches to solve the two tasks, or considered statute semantics while understanding PCR, but no work has tried to model the simultaneous retrieval of both statutes and precedents.

Our experiments have revealed that this could be a difficult exercise, since different types of features (lexical vs. semantic) are important for the two different tasks. The multi-task results are counter-intuitive, since despite the inherent relationship between statutes and precedents cited from the same query, independently trained models fare better in most settings. Despite our best engineering efforts, such as, using variable learning rates for different layers, advanced negative sampling techniques including in-batch sampling, etc., we are not able to improve multi-task results for any of the models, even in the ensemble setup. Although we have discussed some of the possible reasons behind this in Section 5, this needs more investigation and thorough studies.

We also need to consider the fact that the concept of relevance in the legal domain can be quite narrow, as in, all prior cases similar to the query are not necessarily cited. Similarly, only a particular statute from a family of similar statutes is usually applied based on the exact circumstances of the case. In fact, based on some consultations we have had with legal experts in India, two legal experts may differ in the exact cases they choose to cite for a query. Thus, there is a need to also conduct manual (human) annotation to explicitly verify/broaden the set of statutes or precedents cited from a given query. Human evaluation may further be needed to understand the difficulty of the tasks themselves, and would put a clear perspective to the results achieved by different models.

Ethical Considerations

In this work, we propose a system that allows for the simultaneous retrieval of statutes and precedents given a query case. Both these tasks are extremely crucial for the legal domain, and legal professionals desperately require technological assistance to reduce the search space of candidate statutes/precedents. These methods are designed to provide relevant recommendations to the legal professionals, and are not expected to be integrated directly into the decision-making process of the judicial system.

Further, we ensured that all case documents used in our dataset **IL-PCSR** are publicly available. We also took steps to pre-process the documents by removing entity mentions that can lead to biases in the models.

References

Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 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*, pages 1657–1660.

Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2023. Deeprhole: deep learning for rhetorical role labeling of sentences in legal case documents. *Artificial Intelligence and Law*, pages 1–38.

Sebastian Bruch, Siyu Gai, and Amir Ingber. 2023. An analysis of fusion functions for hybrid retrieval. *ACM Transactions on Information Systems*, 42(1):1–35.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. *arXiv preprint arXiv:1906.02059*.

Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. 2021. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 226–241, Online. Association for Computational Linguistics.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. 2013. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16:63–90.

Abhinav Joshi, Akshat Sharma, Sai Kiran Tanikella, and Ashutosh Modi. 2023. U-CREAT: Unsupervised case retrieval using events extrAcTion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13899–13915, Toronto, Canada. Association for Computational Linguistics.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022a. Named entity recognition in indian court judgments. *arXiv preprint arXiv:2211.03442*.

Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022b. Corpus for Automatic Structuring of Legal Documents. In *Proceedings of the 13th Language Resources and Evaluation Conference -Association for Computational Linguistics (ACL-LREC)*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023a. Sailer: Structure-aware pre-trained language model for legal case retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 1035–1044, New York, NY, USA. Association for Computing Machinery.

Haitao Li, You Chen, Zhekai Ge, Qingyao Ai, Yiqun Liu, Quan Zhou, and Shuai Huo. 2024. Towards an in-depth comprehension of case relevance for better legal retrieval. In *JSAI International Symposium on Artificial Intelligence*, pages 212–227. Springer.

Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023b. Thuir@coliee 2023: Incorporating structural knowledge into pre-trained language models for legal case retrieval.

Victor Xiaohui Li. 2023. Findkg: Dynamic knowledge graph with large language models for global finance. *Available at SSRN 4608445*.

Yifei Liu, Yiquan Wu, Yating Zhang, Changlong Sun, Weiming Lu, Fei Wu, and Kun Kuang. 2023. Mlljp: multi-law aware legal judgment prediction. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1023–1034.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2727–2736, Copenhagen, Denmark. Association for Computational Linguistics.

Shengjie Ma, Chong Chen, Qi Chu, and Jiaxin Mao. 2024. Leveraging large language models for relevance judgments in legal case retrieval. *ArXiv*, abs/2403.18405.

Yixiao Ma, Qingyao Ai, Yueyue Wu, Yunqiu Shao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. Incorporating retrieval information into the truncation of ranking lists for better legal search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 438–448.

Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021a. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021*.

Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021b. Lecard: a legal case retrieval dataset for chinese law system. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2342–2348.

Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Angshuman Hazarika, Shubham Nigam, Arnab Bhattacharya, and Ashutosh Modi. 2022. Semantic Segmentation of Legal Documents via Rhetorical Roles. In *Proceedings of the Natural Legal Language Processing Workshop (NLLP) EMNLP*.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021. ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Priyanka Mandikal and Raymond Mooney. 2024. Sparse meets dense: A hybrid approach to enhance scientific document retrieval. *arXiv preprint arXiv:2401.04055*.

Shounak Paul, Rajas Bhatt, Pawan Goyal, and Saptarshi Ghosh. 2024. Legal statute identification: A case study using state-of-the-art datasets and methods. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2231–2240.

Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2022. Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11139–11146.

Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. 2024. Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2210–2220, New York, NY, USA. Association for Computing Machinery.

Juliano Rabelo, Mi-Young Kim, and Randy Goebel. 2022. Semantic-based classification of relevant case law. In *JSAI International Symposium on Artificial Intelligence*, pages 84–95. Springer.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

KL Sumathy et al. 2016. A hybrid approach for measuring semantic similarity between documents and its application in mining the knowledge repositories. *International Journal of Advanced Computer Science and Applications*, 7(8).

Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2024a. Casegnn++: Graph contrastive learning for legal case retrieval with graph augmentation. *arXiv preprint arXiv:2405.11791*.

Yanran Tang, Ruihong Qiu, Yilun Liu, Xue Li, and Zi Huang. 2024b. Casegnn: Graph neural networks for legal case retrieval with text-attributed graphs. In *European Conference on Information Retrieval*, pages 80–95. Springer.

Yanran Tang, Ruihong Qiu, Hongzhi Yin, Xue Li, and Zi Huang. 2024c. Caselink: Inductive graph learning for legal case retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2199–2209, New York, NY, USA. Association for Computing Machinery.

Andrew Vold and Jack G. Conrad. 2021. Using transformers to improve answer retrieval for legal questions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 245–249, New York, NY, USA. Association for Computing Machinery.

Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019. Hierarchical matching network for crime classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 325–334, New York, NY, USA. Association for Computing Machinery.

Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and ShaoZhang Niu. 2018. Modeling dynamic pairwise attention for crime classification over legal articles. In *the 41st international ACM SIGIR conference on research & development in information retrieval*, pages 485–494.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023. Precedent-enhanced legal judgment prediction with llm and domain-model collaboration. *arXiv preprint arXiv:2310.09241*.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, 2:79–84.

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

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Zhen Hu, Heng Wang, et al. 2019. Cail2019-scm: A dataset of similar case matching in legal domain. *arXiv preprint arXiv:1911.08962*.

Linan Yue, Qi Liu, Lili Zhao, Li Wang, Weibo Gao, and Yanqing An. 2024. Event grounded criminal court view generation with cooperative (large) language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2221–2230.

Yiming Zeng, Ruili Wang, John Zeleznikow, and Elizabeth Kemp. 2007. A knowledge representation model for the intelligent retrieval of legal cases. *International Journal of Law and Information Technology*, 15(3):299–319.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3540–3549.

Appendix

Table of Contents

List of Tables

List of Figures

A Related Work

Identifying the legal statutes and relevant prior cases given a legal fact or situation is one of the most fundamental tasks in law. Traditionally, researchers have used statistical and lexical approaches to solve both tasks independently. The advent of deep learning NLP approaches has led to renewed efforts in both tasks using advanced architectures.

A.1 Overview of Prior Works

Traditional approaches for identifying relevant statutes and precedents have mostly involved exploiting lexical features such as n-grams of words (Salton and Buckley, 1988), hand-crafted features (Zeng et al., 2007) or embeddings from pre-trained models like Doc2vec (Le and Mikolov, 2014). Lately, transformer-based embedding methods have been used for directly calculating dot product scores between the query and statute/precedent (Vold and Conrad, 2021). While most unsupervised approaches have utilized methods like Vector Space Model (Salton et al., 1975) and BM25 (Robertson et al., 2009), supervised approaches for both tasks can broadly be divided into classification (Liu et al., 2023; Hofmann et al., 2013) (model predicts similarity between query and statute/precedent) and ranking based (Wang et al., 2018; Ma et al., 2022) (model ranks a list of statutes/precedents based on relevance to the query) approaches.

A.2 Identifying Legal Statutes

Historically, researchers have used multi-label learning frameworks to identify relevant statutes for a query (Wang et al., 2018, 2019; Chalkidis et al., 2019). In many jurisdictions, identifying the relevant statutes is considered to be a subtask of the broader task of Legal Judgment Prediction (Zhong et al., 2018), which could entail predicting the legal charges and term of punishment as auxiliary tasks. Some approaches have only considered the text of the queries in the classification pipeline, relying on the encoder to generate good quality representations of the query (Chalkidis et al., 2019). Others have incorporated the text of the statutes as well, in generating statute-aware query representations which are then used for classification (Wang et al., 2018, 2019). It should be noted that most of these approaches have worked in a setup with limited number of statutes (<200), and hence the classification approach suffices. Lately, LLMs have been used to perform the task of statute identification (Wu et al., 2023), and these models can utilize their superior language understanding capabilities as well as knowledge of legal statutes to excel in the task of statute identification.

A.3 Identifying Precedents

Unlike statutes, most prior works on prior case retrieval have modeled the task in a ranking framework. The major challenge in this task is the fact that both the queries and precedents are very long. Additionally, it has been observed that the query consists of several legal aspects, and each individual aspect leads to matching with certain precedents that eventually get cited (Rabelo et al., 2022). Mostly, researchers have tried to reduce the noise in the query text by using event information (Joshi et al., 2023; Tang et al., 2024b), or extracting salient portions of the document (Qin et al., 2024). Rabelo et al. (2022) took a granular approach, by dividing both the queries and precedents into paragraphs/sentences, scoring each pair of query and precedent sentence, and then generating aggregate scores. Other approaches have involved usage of GNNs (Tang et al., 2024b,c), citation network structures (Bhattacharya et al., 2020), making use of the statutes cited from the precedent cases (not the queries) (Qin et al., 2024), and re-ranking approaches based on some first stage retriever like BM25 (Ma et al., 2021a). LLMs have also been lately used to summarize the queries and precedents (Qin et al., 2024), or perform query expansion based on its inherent domain knowledge (Ma et al., 2024).

B Dataset Construction Details

This section describes the procedure of constructing the IL-PCSR dataset in details. As described briefly in Section 3, we collected 20k case judgment documents from `indiankanoon.org`.

We first perform simple pre-processing of the collected documents, such as removal of consecutive punctuations, whitespace, etc., spelling correction, and filtering gibberish text patterns. These errors are seen in legal case documents since in many cases the digital versions of the cases available on IndianKanoon might have been generated using automated OCR methods.

To designate our query set and statute and precedent candidate pools, we next follow the steps de-

scribed below. Also, to note, in our setting, training, validation and testing all share the same candidate pool.

(i) **Filtering by length:** We measured the length (in terms of number of tokens after NLTK (Loper and Bird, 2002) tokenization) of all the documents in our corpus, and removed very small (< 5 percentile, approx. 400 tokens) and very large (> 95 percentile, approx. 10k tokens) cases, giving us approx 18k cases.

(ii) **Intermediate Statute Pool:** We collected all the statutes (Sections/Articles from Central Govt. Acts) cited across all the 18k cases. We only choose those statutes that are cited at least 5 times across 18k cases, giving us an intermediate statute set of around 1200 statutes. Additionally, we add 19 statutes to the candidate set that are not cited in any query, to conform to a real-world setting where many non-relevant candidates may be present in the pool.

(iii) **Intermediate Precedent Pool:** We also enumerated all the prior cases cited from these 18k cases, which are also part of our corpus. We only choose those precedents that are cited at least 3 times across 18k cases, giving us an intermediate precedent set of around 5k documents. We also add 94 precedent cases that are not cited in any query.

(iv) **Final Query Set:** From our pool of 18k cases, we choose those cases that cite at least one statute and two precedents from their respective intermediate pools. This gives us a final query set of 6271 queries. This set was randomly divided into train/dev/test splits in the ratio of 80%:10%:10%, giving us train, dev and test sets of sizes 5021, 627 and 627 respectively.

(v) **Final Statute and Precedent Pools:** Finally, we filter our intermediate candidate pools further, since we want those candidates that are cited more across the final query set. However, we also want a few zero-shot candidates to conform to a real-world setting. For statutes, we choose only those statutes that are cited at least 4 times across train/dev/test are chosen; and we sample few statutes that are cited less to satisfy the zero-shot property. This process gives us a final statute pool of 936 statutes. We apply a similar policy for precedents, choosing those cases cited at least 3 times across the query set, and some more for zero-shot, giving us a final precedent pool of 3183 cases. Of the 936 statutes, 19 are not cited from any query, and 29 are cited only from the test set but not the train set (zero-shot). Similarly, of the 3183 precedents, 94 are not cited from any query and 157 are zero-shot candidates.

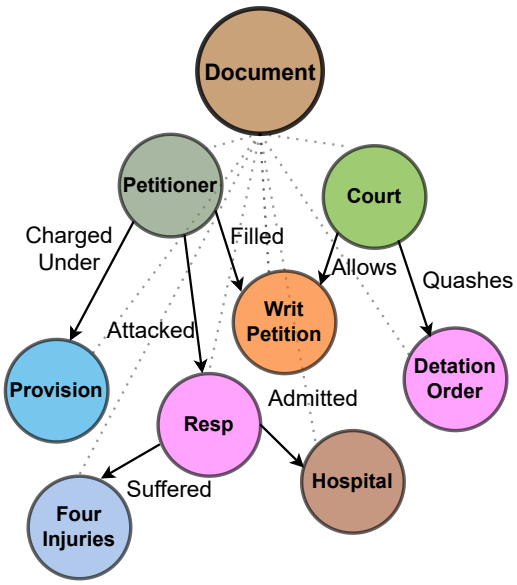


Figure 2: Part of the graph of a case document based on LLM-generated events (input for Event-GNN)

(vi) **Anonymization and Masking Citations:** We mask the portions of the query document where the citation occurs, to prevent deep learning models from associating the queries with the Section numbers, Act names or Case titles. Apart from these, we also anonymize the documents with regard to person names to prevent ethnic/religious biases. We utilize the LegalNER (Kalamkar et al., 2022a) tool by Opennyai, which is capable of extracting mentions of Section numbers, Act names, Case titles, as well as entity names, to fair degree of accuracy. We replace the actual text with placeholders such as [SECTION], [ACT], [PRECEDENT] and [ENTITY].

C Details of Implementation & Experimental Setup

All GPU-based experiments were conducted on a single Nvidia RTX A100 80 GB GPU. The details of individual models are as follows. The compute costs and time for different experiments are listed in Table 4.

BM25: For BM25 experiments, both in default settings as well as event-filtered sentences (SpaCy or LLM), we experimented with n-gram=2. The vocabulary construction was performed with min_df $= 1$ document and max_df $= 65\%$ of the corpus. We also set $b = 0.7$ and $k_1 = 1.6$.

SpaCy Events: For event extraction, we used the

Experiment	Cost	Time
Prediction Tasks		
SAILER (inference)	20 GB	5m
SAILER (fine-tuning)	80 GB	4h
SAILER (summary inference)	20 GB	2m
SAILER (summary fine-tuning)	80 GB	2h 30m
Event-GNN	30 GB	35m
Para-GNN	64 GB	1h 20m
Para-GNN (summary)	45 GB	45m
Event and Summary Generation		
GPT-4 (events)	25 USD	3h
Gemma (fine-tuning)	80 GB	4h 30m
Gemma (inference)	40 GB	12h
GPT-4o-mini (summaries)	30 USD	30h

Table 4: Compute costs and expenses incurred for training/inference using different models. Costs are represented either in terms of GPU Compute (in GBs) for free models or API costs (in USD) for paid models. Time represents the time taken for each epoch in the case of training experiments.

`en-core-web-trf` model for SpaCy and followed the same steps as Joshi et al. (2023).

LLM Events: As described in Section 4, we used the SALI ontology to obtain definitions of events to be provided to LLMs. We chose 18 top-level nodes as entity types, and asked the LLM to extract meaningful verbs/phrases signifying the action or relation between the head and tail entities. Specifically, the exact prompt is described in Table 5. Figure 2 shows a sub-graph of the events generated for a sample document. We first used `GPT-4-turbo` to obtain the events for a small set of documents (~ 400). Thereon, we used a smaller, open model `gemma-7b-it` and fine-tuned over the GPT outputs, to be able to mimic the performance of GPT for event extraction. This fine-tuned Gemma model was subsequently used to obtain the events for the entire dataset (queries and precedents). We followed this pipeline to replicate the superior performance of GPT at lesser financial expense.

Hyper-parameters for Gemma: For fine-tuning the Gemma model, we used a batch size of 4, l.r. 2e-4 and trained the model for 10 epochs using PEFT. We used 4-bit quantization and $r = 8, \alpha = 16$ for LoRA parameters. During inference, we used a batch size of 1 and greedy search decoding.

SAILER: For SAILER, we used the fine-tuned English model available at `CSHaitao/sailer-en-finetune`. We encoded each paragraph using the model (first 512 tokens), and then took the average embedding as the final representation. We then calculated dot product over these embeddings.

C.1 Fine-tuning Setup

For fine-tuning experiments, we used a contrastive learning setup. To elaborate, for every query, we sampled a single positive candidate and a fixed number of BM25 hard negative candidates. We also used in-batch sampling, where in, positives for other queries in a batch can be considered as negatives for the current query. We also ensured each positive candidate for each query was seen during training. For the multi-task experiments, we took a ratio of 3:7 for statute to precedent loss.

SAILER: For SAILER fine-tuning, we used a batch size of 4, 1 positive and 3 negative examples per query, and trained the model for 20 epochs with a peak l.r. of 5e-6.

GNN-based methods: For both GNN-based methods, we use a 2-layer Graph Attention Network. All node and edge embeddings are initialized with SentenceBERT (Reimers, 2019). We used a batch size of 32, 1 positive and 999 negative examples per query, and trained the model for 100 epochs with a peak l.r. of 1e-4.

C.2 Setup and Prompts for Summarization

As described in Section 4, we used an LLM (`GPT-4o-mini`) to summarize both the precedents (candidate cases) and queries to reduce noise and ease computation overhead during both training and inference. We asked the LLM to perform a retrieval-focused summarization, focusing on the reasons for citation.

For precedents, we asked the model to focus on the legal findings and rulings of the court (prompt given in Table 6). For queries, we used two separate prompts to focus on the legal facts and issues (for statutes – see Table 7) and arguments and lower court findings (for precedents – see Table 8).

C.3 Evaluation Metrics

We use macro-F1@k scores for evaluation. We follow the same evaluation scheme as followed by Joshi et al. (2023), wherein the scores for a particular method are calculated for all $k \in \{1, 2, \ldots, 10\}$ for the validation set, and the best k is chosen for evaluation on the test set for that particular method. Apart from F1, we also report the Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) scores for all models.

As an Indian lawyer, your job is to understand legal documents. Right now, you're building a detailed knowledge graph based on information in a given legal document. It's crucial that this graph includes all the fact, evidences, observations from the document, so nothing important is left out. The goal is to make legal analysis easier by focusing on the key information and skipping the obvious stuff.
Each triplet should be in the form of (h:type, r, o:type), where 'h' stands for the head entity, 'r' for the relationship, and 'o' for the tail entity. The 'type' denotes the category of the corresponding entity.
The Entities should be non-generic and can be classified into the following categories:
- Actor / Player: A person who has a role in a legal matter (e.g., Buyer, Provider, Lawyer, Law Firm, Expert, Employer, Employee, Buyer, Seller, Lessor, Lessee, Debtor, Creditor, Payor, Payee, Landlord, Tenant).
- Area of Law: The practice area into which a legal matter or legal area of study falls (e.g., Criminal Law, Real Property Law, Mergers and Acquisitions Law, Personal and Family Law, Tax and Revenue Law).
- Asset Type: Type of resource that is owned or controlled by a person, business, or economic entity
- Communication Modality: Entities' chosen communication method (e.g., written, email, telephone, portal), as well as time (e.g., synchronous, asynchronous).
- Currency: A standardization of money that is used, circulated, or exchanged (e.g., banknotes, coins).
- Document / Artifact: A written, drawn, presented, or memorialized representation of thought or expression, including evidence such as recordings and other artifacts.
- Engagement Terms: Terms to define an engagement for providing legal services.
- Event: A matter's events, as well as collections of those events (often noted as "phases").
- Forums and Venues: Organization or government entity that administers proceedings.
- Governmental Body: Administrative entities of government or state agency or appointed commission, as a permanent or semi-permanent governmental organization that oversees or administers specific governmental functions.
- Industry: An economic branch that produces a related set of raw materials, goods, or services (e.g., Agriculture Industry, Pharmaceuticals Industry).
- Legal Authorities: Documents or publications that guide legal rights and obligations (e.g., caselaw, statutes, regulations, rules) or that can be cited as providing guidance on the law (e.g., secondary legal authorities).
- Legal Entity: A person, company, organization, or other entity that has legal rights and obligations.
- Location: The name of a position on the Earth, usually in the context of continents, countries, and their political subdivisions (e.g., regions, states or provinces, cities, towns, villages).
- Matter Narrative: A textual narrative of a matter's factual and legal details.
- Objectives: Specific aims, goals, arguments, plans, intentions, designs, purposes, schemes, etc. that are constructed by a party in a legal matter, and the legal or other professional frameworks that support their execution.
- Service: The legal work performed, usually by a Legal Services Provider, in the course of a legal matter.
- Status: The state or condition of a proceeding, legal element, or legal matter (e.g., open, closed, canceled, expired).
The Relationships r between these entities must be represented by meaningful verbs/actions and its properties like cause purpose manner etc .
Remember to conduct entity disambiguation, consolidating different phrases or acronyms that refer to the same entity. Simplify each entity of the triplet to be no more than three or four words.
Include triplets that are implicitly inferred from the document's context but not explicitly stated, in order to ensure the graph is both connected and dense.

Table 5: Prompt used for LLM events

Summarize the key points from a provided case document that contributed to the final judgment. These summaries will later be used to identify the reasons why this case might be cited as a precedent. Please process the given legal precedent and focus on the following instructions: Objective: Identify and extract the key legal findings, principles, or rules established in this precedent that could serve as the basis for its citation in other judgments. Structure: Each key points should be phrased in a concise and neutral manner.Avoid including case-specific details (e.g., names, dates, or specific statutes cited). Ensure the summaries comprehensively capture the reasons, enabling effective matching with those from the queries. Focus Areas: Prioritize the sections where legal principles are established, clarified, or interpreted, focusing on the parts likely to be cited as precedents.

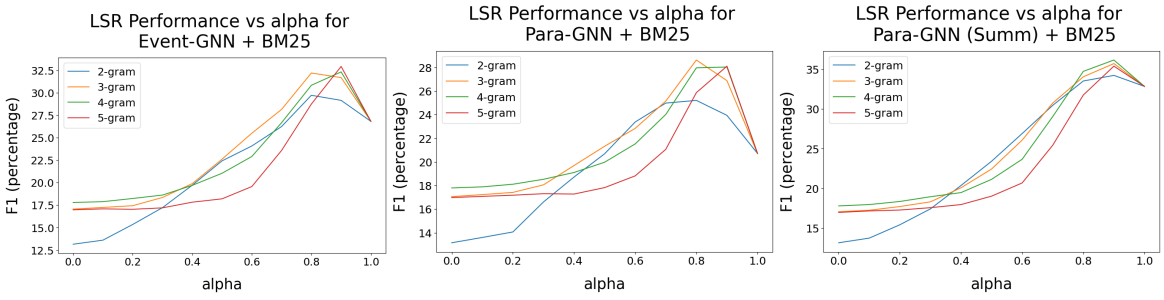
Table 6: Prompt used for precedent summarization

Extract legal incidents from a given judgment to understand why specific sections or articles of law were cited. These extracted incidents will later be matched with relevant sections and articles. Please process the given legal judgment and focus on the following instructions: Objective: Identify and extract all legal incidents referenced in the judgment, focusing on the key facts and legal issues of the case. Structure: Phrase each incident concisely and neutrally. Exclude case-specific details (e.g., names, dates, case numbers). The extracted incidents should be rich in legal reasoning and sufficiently descriptive to enable accurate section/article matching. Focus Areas: Capture the core facts and issues underlying the case.

Table 7: Prompt used for query summarization w.r.t. LSR

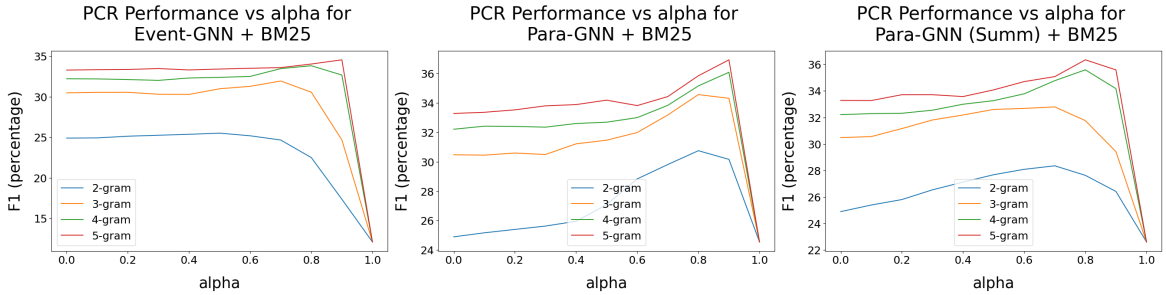
Extract reasons from a legal judgment (query) explaining why the judge cited specific precedents , to later match these reasons with findings from the cited precedents for retrieval tasks. Please process the given legal judgment and focus on the following instructions: Objective: Identify and extract all the legal reasons cited in the given judgment, focusing on the legal principles, rules, or questions of law discussed or evaluated. Exclude any specific factual context or case-specific details. Structure: Each reason should be phrased in a concise and neutral manner. Avoid including case-specific details (e.g., names, dates, or specific statutes cited). Ensure the reasons are comprehensive enough to match with similar principles from other precedents. Focus Areas: While extracting reasons, focus only the places where the precedents and cited text is present.

Table 8: Prompt used for query summarization w.r.t. PCR



(a) LSR on Event-GNN + BM25 (b) LSR on Para-GNN + BM25 (c) LSR on Para-GNN (summaries) + BM25

Figure 3: Grid Search F1(%) of the ensemble methods for LSR task. Each figure shows the plot of performance vs. different α values when combining different models with BM25.



(a) PCR on Event-GNN + BM25 (b) PCR on Para-GNN + BM25 (c) PCR on Para-GNN (summaries) + BM25

Figure 4: Grid Search F1(%) of the ensemble methods for PCR task. Each figure shows the plot of performance vs. different α values when combining different models with BM25.

D Details of the Grid Search Experiment

To further understand the weightage being placed to the lexical and semantic approaches when merging the scores, we vary $\alpha = \{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$ and plot both statute and precedent scores in Figures 3, 4 respectively. For the semantic method, we used Event-GNN for statutes, whereas we used RR-GNN for precedents, since these were the best performing methods for LSR and PCR respectively (Table 2). In both cases, vanilla BM25 was used as the lexical method, and we calculated the metrics for different n-grams of BM25 ($n = 2, 3, 4, 5$). Here, $\alpha = 0$ represents pure lexical score while $\alpha = 1$ represents pure semantic score.

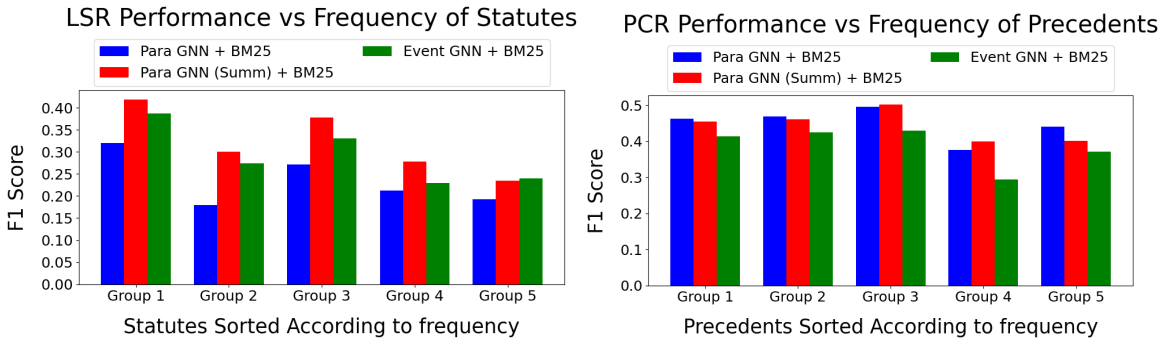
E Analyzing the effect of candidate frequencies

Figures 3, 4 demonstrates that a hybrid approach indeed works the best, for both statutes and precedents respectively. This holds true all the semantic methods, Event-GNN and Para-GNN (both with full documents and summaries as input), and also different n-grams of BM25. In almost all scenarios, the best performance is achieved at high values of α, with the peak performance being achieved as high as $\alpha = 0.9$ in most cases. This indicates that for the semantic models, the scores being assigned to each query, candidate pair have less variance across candidates that are actually relevant versus those that are not, whereas BM25 scores have much higher variance. We also observe significant gains at the optimal α (ensemble score) compared to $\alpha = 0$ (pure lexical score) or $\alpha = 1$ (pure semantic score), for almost all settings. The only exception is Figure 4a (PCR on Event-GNN ensemble), possibly since the pure semantic PCR score by Event-GNN is very low (12.08% F1).

We also observe for all models, the fine-tuned model is able to achieve a performance close to the grid-search approach (see Table 2), which suggests that there is enough signal in the data to learn the optimal α ratio. On inspecting the average α values generated over the test set, we notice that $\alpha = 0.94$ for statutes and $\alpha = 0.78$ for precedents in the case of Event-GNN. For Para-GNN, the average values are $\alpha = 0.85$ and $\alpha = 0.82$ for statutes and precedents respectively. In case of Para-GNN with summaries, the average values are $\alpha = 0.95$ and $\alpha = 0.80$ for statutes and precedents respectively. Nevertheless, the ensemble models outperform the individual lexical and semantic approaches.

Both statutes and precedents are not uniformly distributed in the dataset, usually both these follow a long tail distribution (some candidates are cited by most queries, most other candidates are cited by few). Thus, it is interesting to analyze the performance of different models on different candidates based on their frequency of citation. Figure: 5 shows the performance of the Para-GNN + BM25 Ensemble (Grid Search) model with two variations, using full documents vs. summaries as input. We divide the candidate spaces into 5 equal-sized, disjoint groups based on frequency, and plot the F1 scores for each group considering only those candidates in the given group. On the query side, we only include those queries for evaluation that cite at least one candidate in the group. Naturally, groups containing frequently cited candidates are larger in size than the ones containing rare candidates.

We observe that for statutes, the frequency plays a very crucial role, with the performance decreasing sharply across the most rare groups (Groups 4-5). The difference in performance of the model,



(a) LSR (b) PCR

Figure 5: Performance in terms of F1(%) compared to frequency of candidates. On the X-axis, the candidates are sorted from left to right according to frequency and divided into groups (most frequent group 1, most rare group 5). We compare the Ensemble model (Para-GNN + BM25) under two settings, using full documents vs. summaries. Figure 5a shows LSR performance and Figure 5b shows PCR performance.

Method	Statutes	Precedents
Event-GNN + BM25	**37.03**	42.88
Para-GNN + BM25	28.70	53.18
Para-GNN (summaries) + BM25	**37.03**	**54.30**

Table 9: Performance in macro-F1@K (%) for the best ensemble methods on the zero shot candidates for **IL-PCSR**

when using summaries compared to full documents, is larger for the frequent groups, which got tuned better with the summaries. For precedents, frequency seems to play a much lesser role, with the performance on the rare groups being similar to the frequent groups. It is also interesting to note that there is a significant drop in performance for the summary-based model over Group-5 (most rare precedents). Thus, for both LSR and PCR, the summary-based methods lose their efficacy over the rare candidates.

Performance over Zero-Shot Candidates: As discussed in Section 3, the **IL-PCSR** dataset contains some candidates that are not cited by any training queries but stil cited by some test queries, which can be considered as zero-shot candidates. There are 29 such statutes and 155 such precedents.

Table 9 shows the performance of the best-performing ensemble methods for the zero shot candidates. The trends for the zero-shot candidates follow those for the entire candidate space – Event-GNN and Para-GNN (summaries) perform best for LSR, whereas Para-GNN (full doc) and Para-GNN (summaries) perform well for PCR. Overall, for both tasks, Para-GNN (summaries) perform the best for the zero-shot candidates.

F Analyzing the effect of text lengths

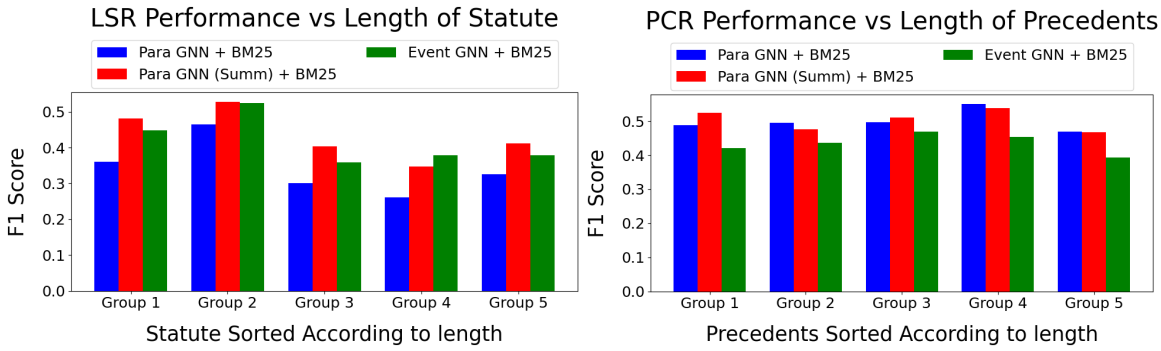
We also try to analyze the effect of text lengths on performance. Here, the text lengths of both queries and candidates need to be analyzed.

Varying Candidate Lengths: Firstly, we try to conduct the analysis from the perspective of candidates. We sort the candidates according to length, and then divide into 5 groups and plot the performance in Figure 6.

We observe that in general, LSR performance drops for the lengthy statutes (lower performance on Groups 3-5 – see Figure 5a), but there is no noticeable drop for PCR (Figure 5b). Trends in relative performance of the 3 models remain fairly consistent across the groups.

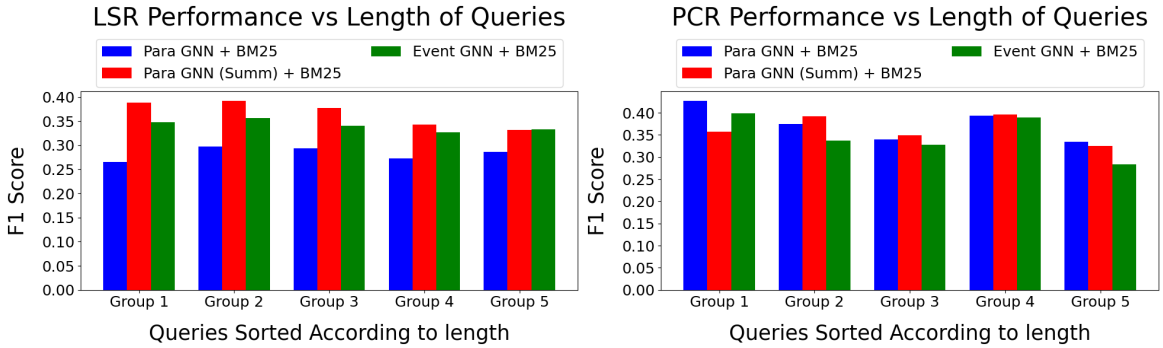
Varying Candidate Lengths: Similarly, we repeat the analysis from the perspective of queries. We sort the queries according to length, and then divide into 5 groups and plot the performance in Figure 7.

Here, we observe that both LSR (Figure 7a) and PCR (Figure 7b) performance drops with increasing length of the query, with the possible exception of Group 4 in case of PCR. The dip in performance is much higher for LSR though. Again, similar to the trends seen for the candidate side analysis, the relative performance between the models remain fairly consistent across all groups.



(a) LSR (b) PCR

Figure 6: Performance in terms of F1(%) compared to text lengths. On the X-axis, the candidates are sorted from left to right according to text length and divided into groups (shortest candidates group 1, longest candidates group 5). We compare the Ensemble models (Event-GNN + BM25, Para-GNN + BM25 and Para-GNN (summaries) + BM25). Figure 6a shows LSR performance and Figure 6b shows PCR performance with varying candidate (statute and precedent respectively) lengths.



(a) PCR (b) PCR

Figure 7: Performance in terms of F1(%) compared to text lengths. On the X-axis, the queries are sorted from left to right according to text length and divided into groups (shortest queries group 1, longest queries group 5). We compare the Ensemble models (Event-GNN + BM25, Para-GNN + BM25 and Para-GNN (summaries) + BM25). Figure 7a shows LSR performance and Figure 7b shows PCR performance with varying query lengths.