LeCNet: A Legal Citation Network Benchmark Dataset

Pooja Harde
National Institute of Technology Kurukshetra
Kurukshetra, Haryana, India
pmharde29@gmail.com

Bhavya Jain
Indian Institute of Technology Bhilai
Durg, Chhattisgarh, India
bhavyaj@iitbhilai.ac.in

Sarika Jain
National Institute of Technology Kurukshetra
Kurukshetra, Haryana, India
jasarika@nitkkr.ac.in

Abstract

Legal document analysis is pivotal in modern judicial systems, particularly for case retrieval, classification, and recommendation tasks. Graph neural networks (GNNs) have revolutionized legal use cases by enabling the efficient analysis of complex relationships. Although existing legal citation network datasets have significantly advanced research in this domain, the lack of large-scale open-source datasets tailored to the Indian judicial system has limited progress. To address this gap, we present the Indian Legal Citation Network (LeCNet) - the first open-source benchmark dataset for the link prediction task (missing citation recommendation) in the Indian judicial context. The dataset has been created by extracting information from the original judgments. LeCNet comprises 26,308 nodes representing case judgments and 67,108 edges representing citation relationships between the case nodes. Each node is described with rich features of document embeddings that incorporate contextual information from the case documents. Baseline experiments using various machine learning models were conducted for dataset validation. The Mean Reciprocal Rank (MRR) metric is used for model evaluation. The results obtained demonstrate the utility of the LeCNet dataset, highlighting the advantages of graph-based representations over purely textual models.

1 Introduction

The rapid advancement of artificial intelligence (AI) has revolutionized numerous fields, including law, where the automation of legal document analysis has emerged as a critical area of research. Efficient retrieval, classification, and recommendation of legal cases are essential for assisting lawyers, judges, and legal researchers in navigating vast repositories of case law. However, developing AI-based systems for legal analysis demands high-quality datasets tailored to specific jurisdictions, along with robust methodologies to address unique challenges in legal text understanding and reasoning.

The increasing volume of cases adjudicated by judiciaries poses significant challenges in maintaining decision-making accuracy, consistency, and fairness (Varghese, 2024). Legal professionals, including lawyers and judges, require efficient tools to reduce the time and cost associated with legal research, improving the effectiveness and quality of the judicial process. Legal professionals traditionally rely on their expertise and critical thinking to identify and refer relevant prior cases when addressing a specific legal case (query case). While technology has been introduced to assist in this process, its role has largely been limited to rudimentary tools such as keyword searches and Boolean operations. Additionally, with the rapid growth of legal case databases, even the most experienced practitioners find it progressively more difficult to efficiently locate and cite pertinent past cases (noticed cases) (Joshi et al., 2023).

Incorporating machine learning into the recommendation of legal citations is a key step in utilizing AI to assist legal experts. Citations are essential in legal writing, particularly within common law systems, where they support arguments by referencing statutes, administrative regulations, and case law that interprets these laws in various scenarios. The importance of citations is particularly evident in legal education, where the process of selecting law journal editors often includes a stringent evaluation of citation formatting skills (Cross and Spriggs, 2010). Additionally, excelling in more complex tasks like legal text generation and summarization requires a deep understanding of citations, highlighting their crucial role in legal discourse.

(Huang et al., 2021) explores the application of link prediction with the context of legal citation recommendation, also sometimes referred to as

Proceedings of the 1st Workshop on NLP for Empowering Justice (JUST-NLP 2025), pages 18–28
December 24, 2025 ©2025 Association for Computational Linguistics

prior case retrieval (Kano et al., 2019; Joshi et al., 2023), to identify and suggest relevant legal precedents or references using citation networks in legal documents. In general, there are many open domain datasets available for the link prediction task, such as the citation network of articles (Wang et al., 2024), arxiv papers by MAG (Wang et al., 2024), papers100M (Wang et al., 2024), wikikg2 (Vrandečić and Krötzsch, 2014), collab (Wang et al., 2024), etc. Datasets that can be used for the legal citation recommendation task are available for US[1], German(Milz et al., 2021), and Canadian cases (COLIEE)(Kano et al., 2019), IL-PCR (Joshi et al., 2023). Similar works have also been carried out for the Indian judiciary by Paheli et. al. (Bhattacharya et al., 2022) and Khatri et. al. (Khatri et al., 2023). The main limitations of existing work are (1) the absence of a legal dataset specifically designed for the link prediction task and (2) the unavailability of large-scale legal citation datasets tailored to the Indian judiciary. The Supreme Court of India's hierarchical, multi-tier system (including High Courts) yields a sparse, hierarchical network where a few landmark cases dominate citations. These patterns differ from US or German legal networks and from prior document-similarity datasets. The existing datasets raise two research questions and leave them unaddressed. **RQ1:** How do dataset design choices (e.g. number of citations removed or negative sampling strategy) affect model performance and robustness? **RQ2:** What is the impact of using contextual document embeddings (versus other text features) on citation link-prediction in graph based models?

We aim to address the stated research questions and introduce a large-scale Indian Legal Citation Network dataset called LeCNet. The proposed dataset is the first of its kind developed for the Indian judiciary, where legal judgments are represented as nodes and citation relationships between cases as edges ($case \xrightarrow{cites} case$), creating a rich graphical representation of 26,308 nodes and 67,108 directed edges. Beyond structural representation in the form of nodes and edges, the dataset also incorporates advanced node features in the form of document embeddings generated by the Doc2Vec model. LeCNet is designed to benchmark standard citation link-prediction models, including both non-graph and graph-based models. This work advances legal research by offering the following contributions:

- Considering the lack of available benchmarks for the Indian legal setting, we create a new benchmark of Legal Citation Network for the Indian legal system (LeCNet) (section 3).
- We have performed experiments on different LeCNet configurations to determine the best-performing version for the link prediction task.
- We have performed the model-based validation of LeCNet using non-GraphML and GraphML models to understand the structural and contextual dependency for the legal citation recommendation task (section 5.3).

2 Related Work

Researchers have explored various approaches to the challenges of legal domain analytics, such as (Kalamkar et al., 2022; Brugman, 2018; Filtz, 2017; Tang et al., 2020), to improve the efficiency and precision of legal workflows. The field of legal document analysis has witnessed significant progress in recent years, driven by advances in natural language processing (NLP) and graph-based machine learning.

Legal citation recommendation has emerged as a key task in the streamlining of legal research, intending to help legal professionals identify relevant precedents and legal references. However, network-based methods primarily focus on the structural information of the constructed network and tend to overlook the textual content of legal documents. Additionally, these methods are less effective when the network is sparse (Liu and Hsu, 2019). Various commercial tools have been developed to aid legal research through citation-based functionalities. Zhang and Koppaka (Zhang and Koppaka, 2007) highlight a LexisNexis feature that facilitates navigation within a semantic citation network by utilizing textual similarity between citation contexts. Similarly, platforms such as Thomson Reuters' CoCounsel (formerly known as CaseText's CARA A.I.) (CoCounsel), Parallel Search (2020), and Thomson Reuters' Quick Check (Thomas et al., 2020) offer citation recommendation services, though the underlying techniques remain undisclosed. Winkels et al. (Winkels et al., 2014) propose a recommender system tailored to Dutch immigration law, which retrieves cases with

[1] https://www.courtlistener.com/

high between-ness centrality relative to selected legal provisions. Dadgostari et al. (Dadgostari et al., 2021) address the challenge of creating bibliographies for citation-free legal texts by modeling the process as a Markov Decision Problem, where an agent iteratively identifies relevant documents using Q-learning, outperforming simpler approaches for retrieving U.S. Supreme Court decisions. Other researchers (Fowler et al., 2007; Koniaris et al., 2017) have studied the structure of legal citation networks, exploring metrics such as authority, relevance, and network properties like degree distribution and shortest path lengths. Sadeghian et al. (Sadeghian et al., 2018) introduce a system to extract and classify legal citations, predicting their purpose based on predefined labels. Huang et al. (Huang et al., 2021) investigate the use of deep learning techniques like BiLSTM and RoBERTa for legal citation prediction, showing that integrating contextual information enhances accuracy and providing benchmarks for advancing legal natural language processing research.

Legal citation network serve as the foundational structure for citation recommendation systems. These networks represent legal documents (nodes) and their interconnections through citations (edges). In the scientific community, researchers have used citation networks for recommendations in many different domains, such as e-commerce, commercial, and academics (Wang et al., 2024).

In 2011, Kumar et al. (Kumar et al., 2011) inferred document similarity by calculating the Jaccard similarity index between sets of out-citations and in-citations within document clusters, known as Bibliographic Coupling and Cocitation. In 2015, Minocha et al. (Minocha et al., 2015) assessed whether sets of precedent citations (out-citations) appeared in the same cluster to determine document similarity. Liu (Liu, 2017) improved upon Bibliographic Coupling in 2017 by incorporating the titles of out-citation references, thus enriching the model's informational context. In 2020, Bhattacharya et al. (Bhattacharya et al., 2020) applied Node2Vec (Grover and Leskovec, 2016) to map legal cases into vector embeddings, evaluating similarity based on these embeddings. Further, in 2022, Bhattacharya et al. (Bhattacharya et al., 2022) acknowledged the critical role of legal statute hierarchy and integrated it into a heterogeneous graph alongside legal case documents. Pioneering contributions include those by James Fowler et. al. (Fowler et al., 2007) and subsequent efforts by researchers such as Katz et. al. (Katz et al., 2020) and Mike Bommarito et. al. (Bommarito II et al., 2010). Additionally, Hoadley et al. (Hoadley et al., 2021) conducted a large-scale international citation network analysis, although their commercial dataset is not publicly available. This broader ecosystem highlights that legal citation datasets have evolved considerably, offering both structural and textual modalities for research.

The relational structure of legal citation networks makes them well-suited for Graph Machine Learning (GraphML) techniques. In **Graph Machine Learning for Legal Citation Networks**, legal documents, statutes, and related entities are represented as nodes, while edges denote relationships such as cites, relatedTo, or refersTo. This graph-based representation enables the application of powerful models like Graph Neural Networks (GNNs), Graph Convolutional Networks (GCNs), and Graph Attention Networks (GATs), which have demonstrated effectiveness in key tasks such as link prediction, node classification, and representation learning. Recent efforts, such as the Open Graph Benchmark introduced by Hu et al. (Hu et al., 2020), have catalyzed progress in GraphML by providing standardized datasets and evaluation protocols.

Problem Statement

In the legal research community, the area of GraphML and related graph-based datasets is still underexplored. The above-mentioned works concentrate on the effectiveness of legal citation networks; however, the unavailability of datasets and the limited focus on applications in the GraphML domain remain unaddressed. The prior works highly consider the node classification task, which may not be suitable for learning the legal cases representation in the embedding space for a citation network. To spur research in legal analytics, we introduce a large-scale Indian Legal Case Citation Network (LeCNet) gold standard dataset that will serve as a benchmark for the link prediction task. This will further aid in downstream applications like legal citation recommendations, similar case recommendations, etc.

3 Legal Citation Network (LeCNet) Benchmark Dataset

The Legal Citation Network (LeCNet) Benchmark Dataset is a dataset of Supreme Court of India court case judgments in English containing 26308

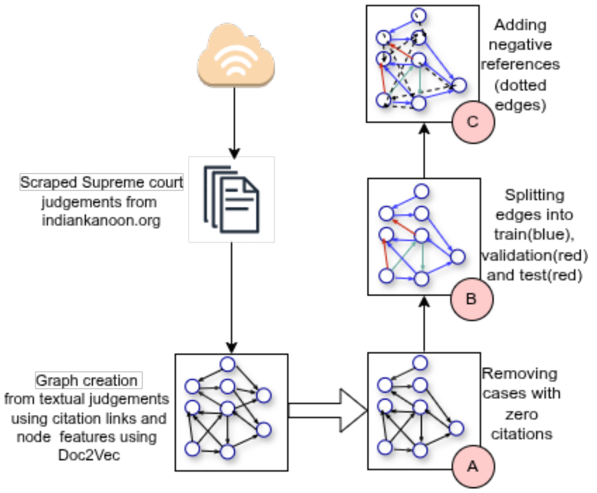
Adding negative references (dotted edges)
C
Scraped Supreme court judgements from indiankanoon.org
Splitting edges into train(blue), validation(red) and test(red)
B
Graph creation from textual judgements using citation links and node features using Doc2Vec
Removing cases with zero citations
A

Figure 1: Designing of LeCNet dataset from scraping to graph creation to preprocessing to edge splitting and finally addition of negative references

legal documents. The case citing other cases are connected via an edge between the nodes, thus creating a directed graph. The mathematical annotation of the dataset can be given as:

LeCNet is a directed graph $G = (V, E)$ where V represents the set of nodes (SCI judgment documents) and $E \subseteq V \times V$ represents the set of edges (citations within the corpus). The LeCNet dataset construction is a three-step process, as described in the following subsections.

3.1 Corpus Creation and Preprocessing

The first step begins with the corpus creation. The corpus is created by scraping legal judgment documents (in the public domain) from the IndianKanoon[2] website. While the site provides access to legal information, it doesn't have a specific section detailing copyright terms for the entire website. The Supreme Court of India (SCI) judgments are simplified in nature as compared to the other Indian courts (High Courts and Apex Courts). A benchmark dataset starts with clean subsets before incorporating more complex structures. So, the SCI has been used for the initial release as a strength to provide a focused, consistent, and reproducibile benchmark before tackling the noisier High Court or tribunal data. In future releases, we plan to extend LeCNet with hierarchical (Supreme–High Court), cross-jurisdictional, and temporal data, making the retrieval challenge progressively more realistic and complex.

We picked documents of the Supreme Court of India (SCI) cases ranging from the year 1950 to 2022 (considered as zero-hop set cases) using a custom Python-based web crawler library Selenium[3]. (Malik et al., 2021)). To gather citation cases, we scraped the documents appearing in the immediate citation neighborhood of each source paper. During the collection of citations, only SCI citations are considered. The created corpus is then preprocessed to provide a unique number to every document to represent as a node in the graph. The data was validated through structured checks guided by six law students and two faculty from Law Department of Kurukshetra University, Haryana. For example, we confirmed the court levels (focusing on Supreme Court cases), set the date ranges, and removed any cases with zero remaining citations (Refer Fig1 A).

Node Features. The node features in the LeCNet dataset are derived from 100-dimensional Doc2Vec embeddings trained on the textual descriptions or metadata associated with each node. This ensures semantically similar nodes have embeddings that are close in the feature space, allowing effective incorporation of textual information into the graph for tasks like link prediction and node classification. The 100-dimensional size balances representational capacity with computational efficiency.

Doc2Vec was chosen for its ability to capture semantic relationships across entire documents, making it well-suited for complex, domain-specific data like legal texts. Prior studies (Mandal et al., 2021; Zhang and Zhou, 2019; Bhattacharya et al., 2022) highlight its superiority over Word2Vec, Legal-BERT and Transformer-based models in representing the nuanced structure of legal documents. They confirmed that Doc2Vec outperformed other models for this domain, validating its use for generating node features in the LeCNet dataset.

3.2 Edge Selection for Dataset Splitting

Further, the dataset has been split into the train/test/validation sets for model development. We assume that all the available tuples form the training set in the beginning. The task is to select the edges to be moved to the test and the validation sets. To handle cases, where nodes have a small number of outgoing citations (e.g., one or two citations), we employed a data-splitting strategy such that if x outgoing citations are randomly

[2] https://indiankanoon.org/

[3] https://selenium-python.readthedocs.io/

moved from the training set for creating a test and a validation set, then at least one outgoing edge will be retained in the training set for that source node (Refer Fig1 B). Additionally, the x selected edges are equally divided between the test and the validation sets; therefore, $x = 2n, \quad n \in \mathbb{N}$. An optimal value of x has to be decided and is done during parameter setting (5.2). The steps for the data splitting are described in Algorithm 1.

Algorithm 1 Edge Selection for Data Splitting

```
1: Take some $x = 2n, \quad n \in \mathbb{N}$. We use x=2 and x=4 to ensure validation and test have enough samples.
2: We select source cases based on the number of outgoing citations they contain:
       $$V_{\text{source}} = \{s \in V \mid |E(s)| > x\},$$
   where $V$ is the set of all cases, and $E(s)$ is the set of edges emanating from source case $s$.
3: Repeat for all $s \in V_{\text{source}}$:
     (i) From $E(s)$, randomly select $x$ edges and mark them:
           $$\{e_1^+, \ldots, e_x^+\} \subseteq E(s).$$
     (ii) Move marked edges to validation/testing:
           $$E(s)_{\text{valid}} \subseteq \{e_1^+, \ldots, e_x^+\}$$
           $$|E(s)_{\text{valid}}| = \lfloor \frac{x}{2} \rfloor$$
           $$E(s)_{\text{test}} = \{e_1^+, \ldots, e_x^+\} \setminus E(s)_{\text{valid}}$$
     (iii) Remaining edges go to training:
           $$E(s)_{\text{train}} = E(s) \setminus \{e_1^+, \ldots, e_x^+\}.$$
```

3.3 Selection of Target Negative References:

The purpose of negative references is to provide a contrastive learning signal for the model. Here, the model learns why certain references exist and why others do not. Without negative examples, the model might simply learn to predict any link between cases, regardless of their content or context. Hence, using negative references, the model learns to identify the features that distinguish between true (positive) citations and carefully selected negative citations. Negative samples for the nodes are generated by first marking a constrained set of potential negative references and then randomly selecting NF references for every source node as depicted in Algorithm 2 and Fig1 C. An optimal value of NF has to be decided and is done during parameter setting 5.2.

Algorithm 2 Negative Reference Sampling

```
Repeat for all $s \in V_{\text{source}}$:
  (i) Define the candidate set $V^-(s)$, consisting of all nodes not directly cited by $s$:
        $$V^-(s) = \{v \in V \mid (s, v) \notin E\},$$
      where $V$ is the set of all cases and $E$ is the set of citation edges.
  (ii) From $V^-(s)$, randomly select NF nodes to form the negative reference set for s.:
        $$E^-(s) \subseteq V^-(s), \qquad |E^-(s)| = \text{NF}.$$
```

Thus, the model is trained to predict the missing references and rank them above the negative candidates, ensuring an effective citation recommendation system.

4 Link Prediction Task

Our Link Prediction Task is essentially a Missing Citation Recommendation Task where for each source node, we drop some citations and aim to rank these missing citations (i.e., outgoing edges). To unlock new frontiers for legal researchers in the area of Graph Machine Learning (GraphML) for legal citation recommendation, we consider three non-GraphML and five GraphML models as our baselines for validating the LeCNet dataset. Both the choice of baselines and the ranking formulation (positives ranked above negatives) follow the standard link-prediction setup introduced by (Hu et al., 2020). Legal citations often exhibit relevant legal context characteristics that these models are well-equipped to capture. While this work primarily serves as a resource-oriented benchmark, the chosen models are theoretically grounded and practically relevant for modeling citation behavior in legal texts. Below, we describe how each model obtains node embeddings:

Non-GraphML Models:

- **MLP:** Input node features are directly used as node embeddings.

- **NODE2VEC (N2V):** The node embeddings are obtained by concatenating input features and NODE2VEC embeddings (Grover and Leskovec, 2016; Perozzi et al., 2014).

- **MATRIX FACTORIZATION (MAT. FACT.:** The distinct embeddings are assigned to different nodes and are learned in an end-to-end manner together with the MLP predictor.

GraphML Models:

- **GCN:** The node embeddings are obtained by full-batch Graph Convolutional Networks (GCN) (Kipf and Welling, 2016).

- **GRAPHSAGE (G.SAGE):** The node embeddings are obtained by full-batch GraphSAGE (Hamilton et al., 2017), where we adopt its mean pooling variant and a simple skip connection to preserve central node features.

- **NEIGHBOR SAMPLING (N. SAMP.):** A mini-batch training technique of GNNs that samples neighborhood nodes when performing aggregation (Hamilton et al., 2017).

- **CLUSTERGCN (C.GCN):** A mini-batch training technique of GNNs that partitions the graphs into a fixed number of subgraphs and draws mini-batches from them (Chiang et al., 2019).

- **GRAPHSAINT (G.SAINT):** A mini-batch training technique of GNNs that samples subgraphs via a random walk sampler (Zeng et al., 2019).

4.1 Evaluation Metric

We evaluated model performance using Mean Reciprocal Rank (MRR), a standard metric for ranking tasks such as citation prediction using the link prediction task. For each source case s, the reciprocal rank of a true reference e_i^+ among all candidate cases is defined as:

$$\text{Reciprocal Rank}(s, e_i^+) = \frac{1}{\text{rank of } e_i^+ \text{ among } e^-(s)} \tag{1}$$

The MRR is computed by averaging the reciprocal ranks over all source cases S:

$$\text{MRR} = \frac{1}{|S|} \sum_{s \in S} \frac{1}{\text{rank of } e^+ \text{ among } e^-(s)} \tag{2}$$

MRR captures the position of the first correct citation in the ranked list, aligning well with practical legal retrieval scenarios where users typically seek the most relevant precedent quickly. Its sensitivity to top-ranked predictions makes it particularly suitable for evaluating the utility of citation recommendation models in a legal context. We report MRR scores to ensure comparability with prior work (Hu et al., 2020) and to reflect the effectiveness of our models in producing useful citation suggestions.

5 Dataset Validation

Once the dataset was developed as in section 3, we performed its performance tuning and model-based validation.

5.1 Computational Setup and Reproducibity details

Each experiment was conducted across 10 runs and the best result out of them was considered. The models were implemented in PyTorch v2.5.1 and run using CUDA v12.4 on an NVIDIA GeForce RTX 3060 GPU. We tuned hyperparameters manually based on performance on the validation set given in Table1. All models evaluated in our experiments were trained using Adam optimizer and had 3 layers in their architecture. Model selection was based on the best performance on the validation set. No hyperparameters were tuned using the test set. Specific packages used for graph data processing and model implementation include torch-geometric for GCN, GraphSAGE, GraphSAINT, and Cluster-GCN and node2vec from the stellargraph/torch_geometric.nn.models library. We release the dataset in the best-performing dataset configuration for research use via Drive. The license for the uploaded content is Creative Commons Attribution 4.0 International (CC-BY). We intend that this dataset be used for research purposes in various tasks, such as legal citation recommendation and prior case retrieval. Our implementation of the baseline models follows the general training and evaluation outlined in (Hu et al., 2020) so we do not release separate model-training code and refer readers to the original implementations for reproducibility.

5.2 LeCNet Dataset Performance Tuning

To rigorously analyze the performance and robustness of citation recommendation models on the LeCNet dataset, we conducted studies focusing on two key tunable parameters: (1) the number of outgoing citation edges (denoted as x) to be moved to the test and validation sets, and (2) the number of target negative reference candidates per source

Model	**Learning Rate**	**Epochs**	**Additional Hyperparameters**
MLP	0.01	100	–
N2V	0.01	100	Emb. Dim.=128, Walk len.=40, Walks per node=10
MAT.FAC.	0.01	150	Emb. Dim.=96
G.SAGE	0.001	50	–
N.SAMP.	0.005	150	–
C.GCN	0.001	100	–
G.SAINT	0.001	100	Walk length = 3, Num steps = 100
GCN	0.001	50	–

Table 1: Best-found hyperparameters for different models.

node (denoted as NF). These ablations enabled us to systematically evaluate model behavior under varying levels of structural sparsity and negative sampling complexity. For detailed mean and standard deviation values corresponding to each dataset configuration, one can refer to the tables provided in the Annexure.

5.2.1 Determining Optimal x

The first experiment is based on the edge selection for the dataset splitting into train, test, and validation sets as mentioned in 3.2. The number of negative reference candidates is kept fixed at ($DS_{NF=10}$), while the number of removed outgoing edges is varied ($DS_{x=2}$ and $DS_{x=4}$) to evaluate performance under increasing information sparsity. Algorithm 1 describes how the two dataset configurations $DS_{x\,=\,2}$ and $DS_{x\,=\,4}$ were created for the experiment whose statistics are described in Table 2. These strategies ensure that even for low-degree nodes, sufficient information is preserved in the training set for meaningful learning while still providing test and validation examples for evaluation.

Table 3 *(the first two out of the three major columns)* present the results of running various Machine Learning models on **LeCNet** for the link prediction task for the two dataset configurations to be tested. Contrary to initial expectations, increasing the number of removed citations (from 2 to 4) does not significantly degrade performance for most models as models have access to more structural information. It suggests that withholding a larger number of edges limits the structural information available during training. At the same time, the performance gap between the two settings is relatively small for most models, indicating that increasing x does not severely degrade performance. This balance implies that while the models remain robust even when more edges are shifted out of the training graph, $x = 2$ provides a more favourable trade-off between training signal and evaluation size, and is therefore the more suitable choice for subsequent experiments.

Dataset Config.	E_{train}	E_{valid}	E_{test}
$DS_{\text{x=2}}$	51186	7961	7961
$DS_{\text{x=4}}$	50760	8174	8174

Table 2: Statistics for no. of edges in train, validation, and test sets for the two dataset configurations, one with x=2, the other with x=4

5.2.2 Determining Optimal NF

Another experiment is based on the impact of adding a different number of target negative references to the citation network. We add NF target negative references for each source node in the validation and test sets and the aim of the model is to rank the true missing x citations higher than NF negative references. The negative references are randomly-sampled from all the previous cases that are not cited by the source node.

In this experiment, the number of removed edges is kept fixed ($DS_{x=2}$) *(as decided in section 5.1.1)*, and the number of negative references is varied ($DS_{NF=10}$ and $DS_{NF=25}$) to analyze how models perform under increasing difficulty in identifying correct links from negative references.

Table 3 *(the last two out of the three major columns)* present the results of running various Machine Learning models on **LeCNet** for the link prediction task for the two dataset configurations to be tested. Increasing the number of negative references (from 10 to 25) reduces performance reflecting the increased difficulty of distinguishing true citations from a larger pool of distractors. Notably, GCN maintains performance more effectively than the other architectures, indicating that it retains discriminative capability even when the neg-

Models	$DS_{\text{x=4, NF=10}}$			$DS_{\text{x=2, NF=10}}$			$DS_{\text{x=2,NF=25}}$		
	Train	**Validation**	**Test**	**Train**	**Validation**	**Test**	**Train**	**Validation**	**Test**
MLP	0.838	0.807	0.807	0.830	0.832	0.831	0.734	0.742	0.737
N2V	0.887	0.856	0.857	0.886	0.874	0.873	0.800	0.796	0.791
MAT. FACT.	0.998	0.373	0.372	0.998	0.378	0.377	0.99	0.382	0.380
G.SAGE	0.992	0.979	0.980	0.991	0.985	0.984	0.980	0.968	0.968
N. SAMP.	0.982	0.981	0.982	0.980	0.987	0.985	0.959	0.970	0.970
C.GCN	0.919	0.907	0.909	0.916	0.929	0.926	0.861	0.877	0.875
G.SAINT	0.959	0.955	0.953	0.953	0.963	0.963	0.916	0.933	0.933
GCN	0.991	**0.983**	**0.983**	0.989	**0.988**	**0.987**	0.989	**0.988**	**0.987**

Table 3: MRR performance of Open Graph Benchmark models on LeCNet across three dataset configurations: $DS_{x=4,NF=10}$, $DS_{x=2,NF=10}$, and $DS_{x=2,NF=25}$

ative set becomes substantially larger. $NF = 10$ provides a balanced and reliable configuration for evaluating link prediction performance.

5.3 Model-based Validation

The evaluation on several link prediction models helps us understand the dataset's suitability for link prediction tasks and its comparative performance against established benchmarks. The evaluation of various models on the LeCNet dataset under different configurations highlights the effectiveness of graph-based approaches for legal citation recommendation. Refer to the column 2 *($DS_{x=2,NF=10}$)* of Table 3 for the model-based validation of the LeCNet, as this is the most optimal dataset configuration.

We observe that GCN consistently outperforms other models, achieving the highest Mean Reciprocal Rank (MRR) scores in both validation and testing phases. The performance of NeighborSampling and GraphSAGE is also notable, demonstrating their ability to capture structural dependencies effectively. In contrast, Matrix Factorization exhibits severe overfitting, performing exceptionally well during training but failing to generalize, as indicated by its significantly lower validation and testing scores. Traditional approaches like MLP and Node2Vec lag behind graph-based models, reaffirming the importance of leveraging graph neural networks (GNNs) for this task.

Across varying graph configurations too, GCN demonstrates superior scalability and generalization, making it the most effective model for legal citation prediction.

6 Conclusion

Our findings supplement the significance of graph-based models in learning from citation networks, demonstrating their superiority over traditional methods. Specifically, GCN and GraphSAGE emerge as the most effective models on LeCNet, achieving high Mean Reciprocal Rank (MRR) scores across different data splits. Their ability to capture structural dependencies in citation networks allows for more accurate link prediction compared to non-graph-based approaches like MLP and Node2Vec. Additionally, this study underscores the importance of evaluating models under varying levels of network sparsity, as real-world citation graphs often suffer from missing or incomplete links. Ensuring robustness under such conditions will be crucial for deploying citation prediction models in practical legal applications.

Despite these strong performances, there remains room for further enhancements. We plan to incorporate the paragraph-level citation recommendation (e.g., case facts, citation reasoning, etc.) into the dataset as future work.

Limitations

In this paper, we evaluated the LeCNet dataset using Open Benchmark Dataset models for citation prediction. While the results demonstrate the effectiveness of GNN-based methods and scalable mini-batch training techniques, there remains significant room for improvement. One limitation of our approach is the reliance on official citations as the ground truth, which may not always capture the full scope of relevant citations due to the subjective nature of citation practices in legal writing. Exploring alternative ground truth definitions that account for implicit relevance could improve model performance.

Additionally, our evaluation primarily focused on MRR as the metric, which, while effective, might not fully capture other aspects of citation prediction, such as diversity or contextual relevance. Incorporating alternative metrics could provide a more nuanced understanding of model performance.

Another limitation lies in the dataset itself. The dataset contains only the citations from the Supreme Court of India cases, whereas a case might cite other judicial cases, also like High Courts, which are not considered during dataset creation. Incorporating citations across other courts or jurisdictions (e.g., High Court cases citing Supreme Court decisions) is an important extension that we plan for future releases. LeCNet is intended primarily as a benchmark for citation link-prediction methods in the Indian legal domain, and secondarily as training data for legal retrieval tasks. Note that our current evaluation assumes a fixed corpus; supporting truly new (unseen) documents is an important direction for future work. In future work, we plan to involve legal scholars for expert feedback to further validate the dataset's relevance and the models' real-world applicability.

An important direction for future work is to incorporate harder negative samples, such as cases from similar legal topics or issued in the same year. Such negatives would better reflect the real ambiguity present in citation decisions and create a more challenging and realistic evaluation setting for link prediction models.

Finally, while we conducted extensive experiments using existing Open Graph Benchmark models, further exploration of advanced techniques, such as contrastive learning, hybrid GNNs, or the inclusion of richer features like legal concepts, statutes, or temporal patterns, could enhance the task of citation prediction. Future work could also investigate models capable of incorporating domain-specific knowledge to address the unique challenges presented by citation prediction in the legal domain.

Ethical Considerations

This paper releases a dataset for recommending relevant citations in the legal domain. The purpose is not to replace researchers or legal experts, but to contribute a dataset that can be used to augment their work by facilitating efficient citation recommendations. For training and evaluation, we utilized the LeCNet dataset, which consists of publicly available citation data, ensuring that privacy concerns are not violated.

It is important to note that we did not explicitly address potential biases in the dataset or normalize the data concerning citation patterns. As such, the system's performance may reflect certain biases inherent in the data, such as overrepresentation of frequently cited papers or underrepresentation of niche topics. Future work could involve investigating these biases and exploring methods to ensure a more balanced and fair recommendation system.

References

CaseText 2020. Machine learning behind parallel search. Accessed: 2025-01-10.

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, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2022. Legal case document similarity: You need both network and text. *Information Processing & Management*, 59(6):103069.

Michael J Bommarito II, Daniel Martin Katz, Jonathan L Zelner, and James H Fowler. 2010. Distance measures for dynamic citation networks. *Physica A: Statistical Mechanics and its Applications*, 389(19):4201–4208.

Simon Brugman. 2018. Deep learning for legal tech: exploring ner on dutch court rulings. *Data Science Group Faculty of Science*.

Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 257–266.

CoCounsel. Cocounsel: One genai assistant for professionals. Accessed: 2025-01-10.

Frank B Cross and James F Spriggs. 2010. Citations in the us supreme court: An empirical study of their use and significance. *U. Ill. L. Rev.*, page 489.

Faraz Dadgostari, Mauricio Guim, Peter A Beling, Michael A Livermore, and Daniel N Rockmore. 2021. Modeling law search as prediction. *Artificial Intelligence and Law*, 29:3–34.

Erwin Filtz. 2017. Building and processing a knowledge-graph for legal data. In *The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part II 14*, pages 184–194. Springer.

James H Fowler, Timothy R Johnson, James F Spriggs II, Sangick Jeon, and Paul J Wahlbeck. 2007.

Network analysis and the law: Measuring the legal importance of precedents at the us supreme court. *Political Analysis*, 15(3):324–346.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Daniel Hoadley, M Bartolo, R Chesterman, A Faus, W Hernandez, B Kultys, AP Moore, E Nemsic, N Roche, J Shangguan, and 1 others. 2021. A global community of courts? modelling the use of persuasive authority as a complex network. *Frontiers in Physics*, 9:665719.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.

Zihan Huang, Charles Low, Mengqiu Teng, Hongyi Zhang, Daniel E Ho, Mark S Krass, and Matthias Grabmair. 2021. Context-aware legal citation recommendation using deep learning. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 79–88.

Abhinav Joshi, Akshat Sharma, Sai Kiran Tanikella, and Ashutosh Modi. 2023. U-creat: Unsupervised case retrieval using events extraction. *arXiv preprint arXiv:2307.05260*.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2019. Coliee-2018: Evaluation of the competition on legal information extraction and entailment. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2018 Workshops, JURISIN, AI-Biz, SKL, LENLS, IDAA, Yokohama, Japan, November 12–14, 2018, Revised Selected Papers*, pages 177–192. Springer.

Daniel Martin Katz, Corinna Coupette, Janis Beckedorf, and Dirk Hartung. 2020. Complex societies and the growth of the law. *Scientific reports*, 10(1):18737.

Mann Khatri, Mirza Yusuf, Yaman Kumar, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. 2023. Exploring graph neural networks for indian legal judgment prediction. *arXiv preprint arXiv:2310.12800*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Marios Koniaris, Ioannis Anagnostopoulos, and Yannis Vassiliou. 2017. Network analysis in the legal domain: A complex model for european union legal sources. *Journal of Complex Networks*, 6(2):243–268.

Sushanta Kumar, P Krishna Reddy, V Balakista Reddy, and Aditya Singh. 2011. Similarity analysis of legal judgments. In *Proceedings of the fourth annual ACM Bangalore conference*, pages 1–4.

Rey-Long Liu. 2017. A new bibliographic coupling measure with descriptive capability. *Scientometrics*, 110:915–935.

Rey-Long Liu and Chih-Kai Hsu. 2019. Improving bibliographic coupling with category-based cocitation. *Applied Sciences*, 9(23):5176.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripa Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. 2021. Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation. *arXiv preprint arXiv:2105.13562*.

Arpan Mandal, Kripabandhu Ghosh, Saptarshi Ghosh, and Sekhar Mandal. 2021. Unsupervised approaches for measuring textual similarity between legal court case reports. *Artificial Intelligence and Law*, pages 1–35.

Tobias Milz, Michael Granitzer, and Jelena Mitrović. 2021. Analysis of a german legal citation network. In *International Conference on Knowledge Discovery and Information Retrieval*.

Akshay Minocha, Navjyoti Singh, and Arjit Srivastava. 2015. Finding relevant indian judgments using dispersion of citation network. In *Proceedings of the 24th international conference on World Wide Web*, pages 1085–1088.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Ali Sadeghian, Laksshman Sundaram, Daisy Zhe Wang, William F. Hamilton, Karl Branting, and Craig Pfeifer. 2018. Automatic semantic edge labeling over legal citation graphs. *Artificial Intelligence and Law*, 26(2):127–144.

Mingwei Tang, Cui Su, Haihua Chen, Jingye Qu, and Junhua Ding. 2020. Salkg: a semantic annotation system for building a high-quality legal knowledge graph. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2153–2159. IEEE.

Merine Thomas, Thomas Vacek, Xin Shuai, Wenhui Liao, George Sanchez, Paras Sethia, Don Teo, Kanika Madan, and Tonya Custis. 2020. Quick check: A legal research recommendation system. In *Proceedings of NLLP '20*, volume 2645. CEUR-WS.

Dr John Varghese. 2024. Datafication in judicial case management in india. In *Symposium on Diversity in Legal and Judicial Profession and the Politics of Merit and Exclusion in India, RHUL, London*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Jie Wang, Kanha Bansal, Ioannis Arapakis, Xuri Ge, and Joemon M Jose. 2024. Empowering legal citation recommendation via efficient instruction-tuning of pre-trained language models. In *European Conference on Information Retrieval*, pages 310–324. Springer.

Radboud Winkels, Alexander Boer, Bart Vredebregt, and Alexander von Someren. 2014. Towards a legal recommender system. In *Proceedings of JURIX '14*.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2019. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*.

Haoyang Zhang and Liang Zhou. 2019. Similarity judgment of civil aviation regulations based on doc2vec deep learning algorithm. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–8. IEEE.

Paul Zhang and Lavanya Koppaka. 2007. Semantics-based legal citation network. In *Proceedings of ICAIL '07*, pages 123–130.

A Appendix

A1. Detailed MRR Results for $DS_{x=4,NF=10}$

Models	Train	Validation	Test
MLP	0.838±0.005	0.807±0.002	0.807±0.001
N2V	0.887±0.005	0.856±0.001	0.857±0.001
MAT.FAC.	0.998±0.000	0.373±0.011	0.372±0.010
G.SAGE	0.992±0.000	0.979±0.000	0.980±0.001
N.SAMP.	0.982±0.000	0.981±0.000	0.982±0.000
C.GCN	0.919±0.002	0.907±0.003	0.909±0.003
G.SAINT	0.959±0.003	0.955±0.003	0.953±0.003
GCN	0.991±0.000	0.983±0.000	0.983±0.001

Table 4: MRR results for $DS_{x=4,NF=10}$.

A2. Detailed MRR Results for $DS_{x=2,NF=10}$

Models	Train	Validation	Test
MLP	0.830±0.004	0.832±0.001	0.831±0.002
N2V	0.886±0.007	0.874±0.001	0.873±0.001
MAT.FAC.	0.998±0.000	0.378±0.015	0.377±0.013
G.SAGE	0.991±0.001	0.985±0.000	0.984±0.001
N.SAMP.	0.980±0.001	0.987±0.000	0.985±0.000
C.GCN	0.916±0.005	0.929±0.003	0.926±0.004
G.SAINT	0.953±0.002	0.963±0.001	0.963±0.002
GCN	0.989±0.001	0.988±0.000	0.987±0.001

Table 5: MRR results for $DS_{x=2,NF=10}$.

A3. Detailed MRR Results for $DS_{x=2,NF=25}$

Models	Train	Validation	Test
MLP	0.734±0.007	0.742±0.002	0.737±0.001
N2V	0.800±0.004	0.796±0.002	0.791±0.002
MAT.FAC.	0.990±0.000	0.382±0.012	0.380±0.001
G.SAGE	0.980±0.001	0.968±0.000	0.968±0.001
N. SAMP.	0.959±0.001	0.970±0.001	0.970±0.001
C.GCN	0.861±0.007	0.877±0.007	0.875±0.007
G.SAINT	0.916±0.004	0.933±0.004	0.933±0.005
GCN	0.989±0.001	0.988±0.000	0.987±0.000

Table 6: MRR results for $DS_{x=2,NF=25}$.