Department of Computer Science

Using LLMs for Legal Text Analytics

Capstone Project

Suyog Joshi

Supervisors: Dr. Lipika Dey, Dr. Partha Pratim Das

Presented December 16, 2024

Background

"Language is to lawyers what a piano is to the pianist: the tool of her trade." - Andrei Marmor

- Legal documents are complex, lengthy, and repetitive.
- Challenges in manual analysis due to large volumes of legal text.
- Need a method to automatically analyze legal documents.

Motivation

What is Legal Text Analytics?

- Use of computational tools to extract and analyze information from legal documents.
- Understanding of patterns, relationships, and trends across large volumes of legal text.
  - Why are violations happening? How are they happening? Who is involved?

Importance of Legal Analytics

- Legal Reforms
  - Generate data-driven insights into the nature of cases
  - Identify gaps in the law for recurring violations, inform policy decisions

Motivation

Food Safety Cases

- Understanding Food Violations:
  - Different types of violations (adulteration, illegal sale, chemical use, etc.)
  - Different foods (tobacco products, milk, spices, etc.)
- Organizational Involvement
  - Citizen, organization, government?

Why Knowledge Graphs?

- Natural Representation of Relationships
  - Legal cases are interconnected through statutes, precedents, provisions, and parties involved.
  - Knowledge graphs are a structured representation of these relationships.
- Graph-based Analytics
  - Natural framework to query and analyze legal data.
  - Facilitates insight extraction.

Objectives

Goal: Use large language models (LLMs) to extract entities to populate knowledge graphs (KGs) for insight extraction.

- **Entity Extraction:**
  - Identify key entities (e.g., statutes, provisions, parties, decisions) from legal documents.
  - Enables structured data extraction for downstream analytics.
- **Knowledge Graph Creation:**
  - Represent relationships and insights using a legal ontology (e.g., connections between cases, statutes, provisions, and judgments).
  - Structured representation of interlinked legal data, facilitating exploration and discovery of patterns.
- **Knowledge Graph Queries:**
  - Extract corpus-level insights, such as:
    - Most common statutes cited.
    - Similar cases based on shared citations or legal contexts.
  - Allows detailed analytics on legal data, revealing trends and patterns across the corpus.

Data

- **Source:** *IndianKanoon*
- **Timeframe:** January 2022 - December 2023
- **Focus:** Food Safety and Standards Act

Statistic	**Value (Tokens)**
Total Cases Collected	196
Mean Length	5,064
Maximum Length	53,941
Minimum Length	453
Median Length	1,121
Total Tokens	992,672

Table: Corpus Statistics

- Minimal preprocessing, removed irrelevant links and whitespace.

Methods for Entity Extraction

- **Traditional Statistical NER**: Relies on annotated data, need to be specifically trained for each domain.
- **Transformer models (BERT)** Improve accuracy over statistical NER, but still have to be trained on annotated data.
- **Recent advancements:**
  - Zero-shot & few-shot learning with LLMs (problem: corpus-level insights)
  - Solution: Knowledge graph for cross-document insights.

Entity Extraction

Why Entity Extraction?

- Useful for understanding legal context.
- Section numbers indicate the type of violation (e.g. Section 20 of FSS dictates laws around contaminants present in foods.)
- Respondent/appellant names indicate who is involved in the case (government, corporation, individuals).

Models:

1. **Baseline:** Transformer-based NER model trained on Indian court judgements (Kalamkar et al. 2021).
2. **LLM:** GPT-4o-mini through the OpenAI API.

Transformer-based NER

- **RoBERTa** trained on 14444 Indian court judgements.
- Used pre-trained model available on GitHub.
- Achieved F1 score of **0.911** on the dataset created by *Kalamkar et al.*
- **Problem:** Duplicate entities extracted for each document.
  - Leads to inaccurate analysis.
  - **Solution:** developed method to de-duplicate entities using pre-defined rules and text matching.

De-duplication

- **Rule-based:** Regular expressions to standardize common variations ("FSS Act" $\rightarrow$ "Food Safety and Standards Act, 2006")
  - Cannot make rules for all possible entities.
- **Similarity-based:** Character overlap algorithm to identify similar entities.
  - Entities with a similarity score ≥ 0.7 were considered duplicates.

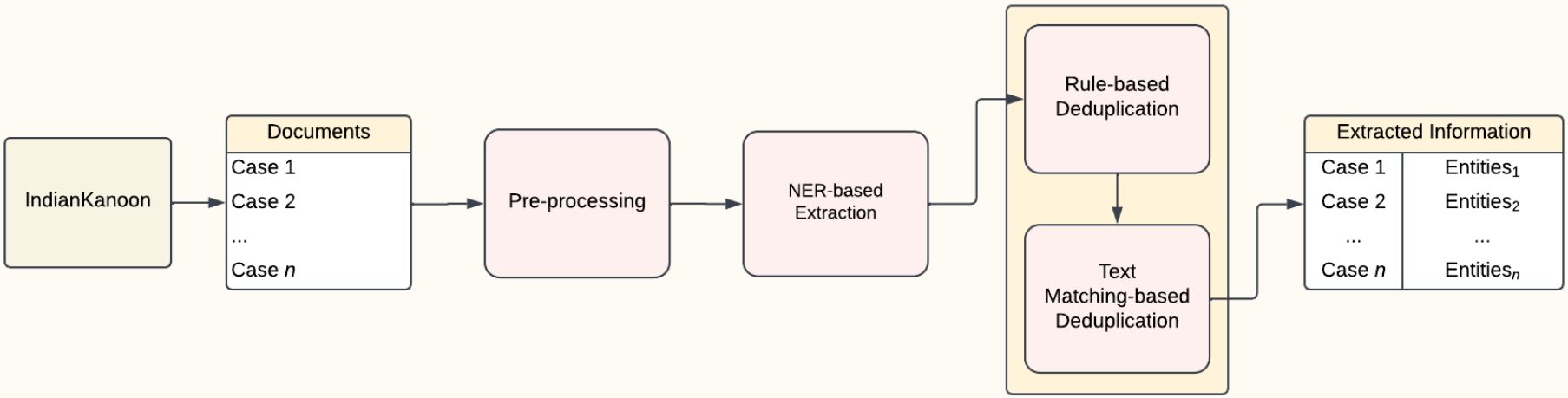


Figure: Pipeline for NER model-based entity extraction

Large Language Models for Entity Extraction

- Traditional NER models need to be re-trained on labeled data for different domains. Deduplication also an issue.
- **Solution:** Use LLMs for structured entity extraction.
- Used **GPT-4o-mini** through OpenAI API with structured outputs.
  - Structured outputs provide results in a predefined schema - easier management.
  - Prompted to only return one instance of each entity mentioned, even if it was mentioned multiple times.
- **Advantage:** can extract any new type of entity/information without needing to re-train model.

Comparison - NER and LLM

Entity Type	LLM	NER Model
Petitioners	✓	✓
Respondents	✓	✓
Judges	✓	✓
Court	✓	✓
Date of Judgement	✓	All dates mentioned
Organizations (ORG)	✓	✓
Locations (GPE)	✓	✓
Provisions	✓	✓
Statutes	✓	✓
Precedents	✓	✓
Key Facts	✓	–
Type of Case	✓	–
Decision	✓	–
Summary	✓	–

Table: Comparison of Entity Types Extracted by LLM and NER Models

LLM-NER Agreement

Performance of LLM on entity extraction was measured relative to NER-extracted entities.

- **True Positive:** The number of statutes that were identified by both the LLM and the NER results. These are the statutes that both the LLM and NER agree on.
- **False Positive:** The number of statutes that were identified by the LLM but were not present in the NER results. These are the statutes that the LLM identified incorrectly.
- **False Negative:** The number of statutes that were present in the NER results but were not identified by the LLM. These are the statutes that the LLM missed.

Strings were considered equivalent iff. the NER and LLM text matched exactly.

LLM-NER Agreement

Entity Type	Precision	Recall	F1-Score
GPE (Locations)	0.2883	0.2026	0.2380
Judges	0.7706	0.4363	0.5571
ORG (Organizations)	0.2911	0.0339	0.0607
Petitioners	0.4722	0.3730	0.4168
Precedents	0.0665	0.0268	0.0382
Provisions	0.1786	0.0919	0.1214
Respondents	0.1329	0.0734	0.0945
Statutes	0.4479	0.2211	0.2960

Table: Comparison of Metrics for Extracted Entities (LLM vs. NER)

- Low scores across classes because:
  1. NER accuracy low, especially on names of respondents/appellants.
  2. Strict matching rule—NER output was not perfectly deduplicated
     - e.g. NER: "AP Excise Act", LLM: "Andhra Pradesh Excise Act"
  3. Both may be partially correct—LLM might have missed something NER detected, and vice-versa
- Need to check LLM against ground truth.

LLM Accuracy

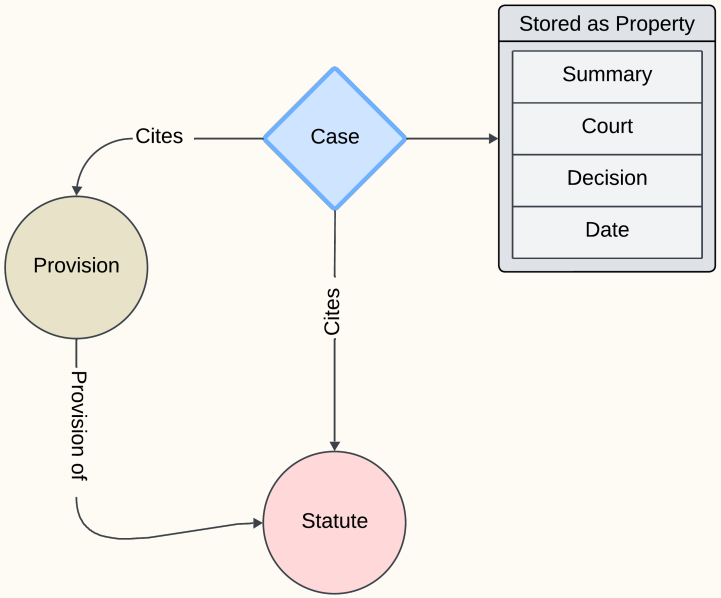
- Manually labeled a small subset of the corpus (15 documents) with the statutes mentioned.
- LLM performs very well, matching the F1 score reported for the NER model by Kalamkar et al.
- However, labeled set is small. More documents should be annotated for better picture.

Entity Type	**Precision**	**Recall**	**F1-Score**
Statutes	0.9615	0.9090	0.9346

Table: Metrics for Extracted Statutes (LLM vs. Ground Truth)

Knowledge Graph Design

- **Purpose:** Represent entities and their relationships in a structured format.
- **Nodes:** Cases, statutes, provisions.
  Edges: Relationships between nodes.
- Focused on generic relationships to ensure generalizibility across legal domains.
- Used Neo4j to store and interact with KG, Cypher as query language.



Full Pipeline

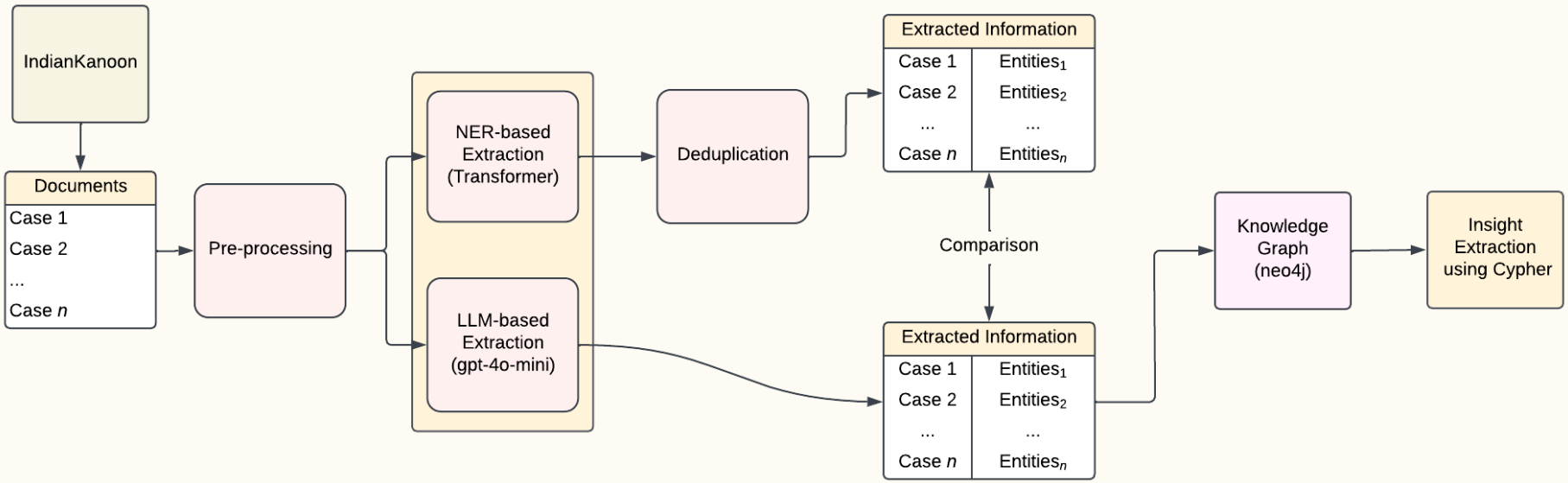


Figure: Full Extraction and KG Population Pipeline

Knowledge Graph

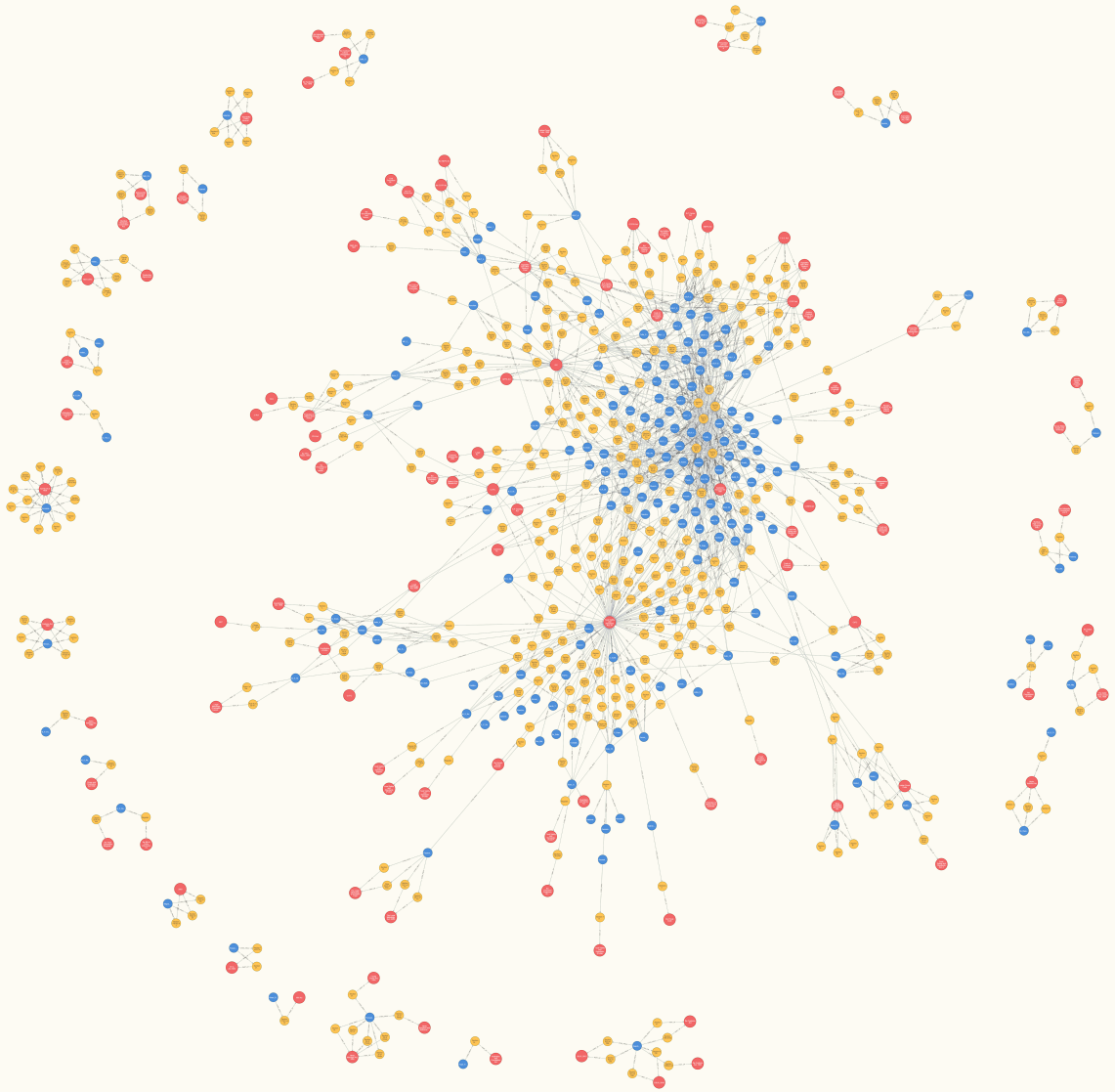


Figure: Full knowledge graph

Zoomed In

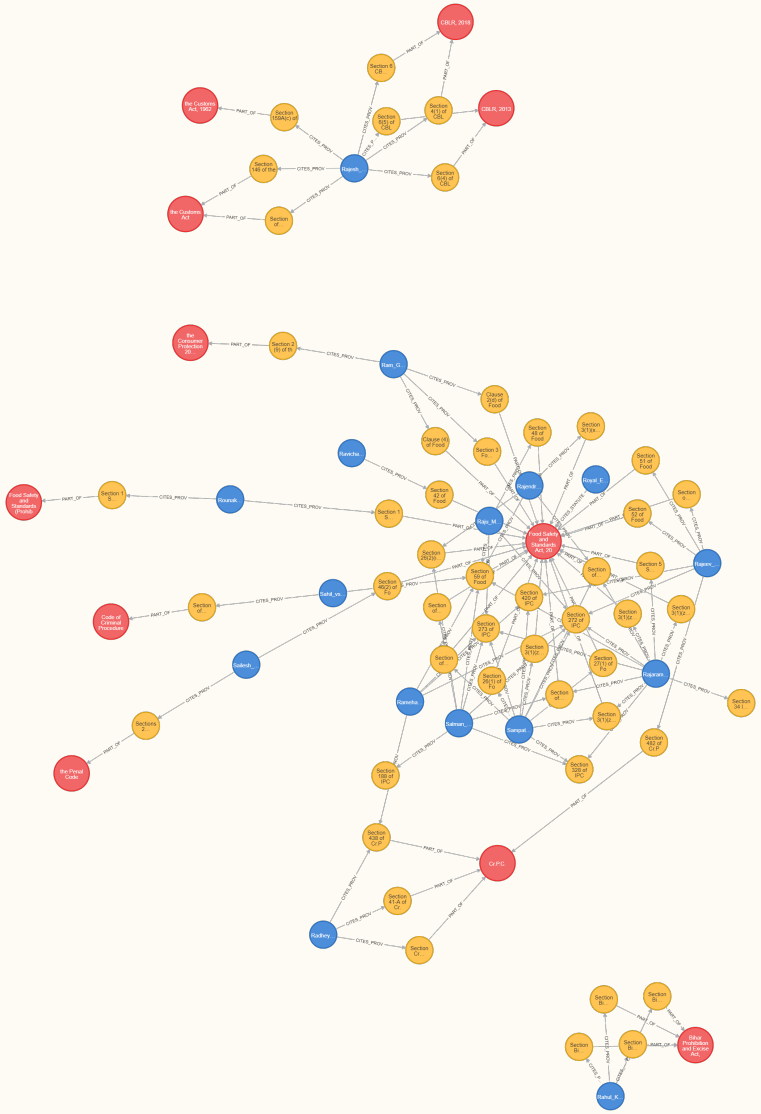


Figure: Knowledge graph limited to 100 relationships

Knowledge Graph Queries

Examples of Insights Extracted:

1. **Most common provisions:** Statutes and sections frequently cited.
2. **Similar cases**: Cases sharing common citations or arguments.
3. **Jaccard Similarity:** Group cases based on overlapping citations to reveal trends.

Insights: Common Provisions

Provision	**Citation Count**
Section 273 of IPC (Sale of noxious food or drink)	76
Section 272 of IPC (Adulteration of food or drink)	63
Section 188 of IPC (Disobedience of order)	49
Section 328 of IPC (Causing hurt by means of poison)	49
Section 59 of Food Safety and Standards Act, 2006 (Punishment for unsafe food)	47
Section 420 of IPC (Criminal deceit)	33
Section 482 of Cr.P.C. (Quashing of FIR)	22
Section 63 of Food Safety and Standards Act, 2006 (Unlicenced sale of food)	19

Table: Top Provisions Cited in Food Safety Cases

Insights: Similar Cases Based on Citation Count

Cases with the highest number of shared citations.

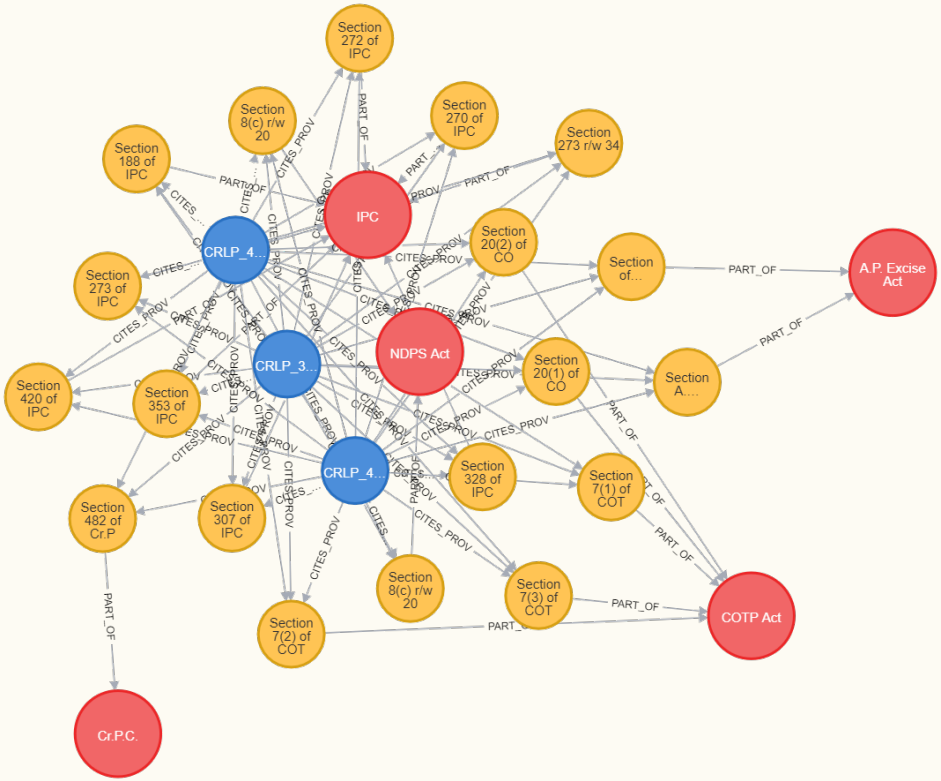
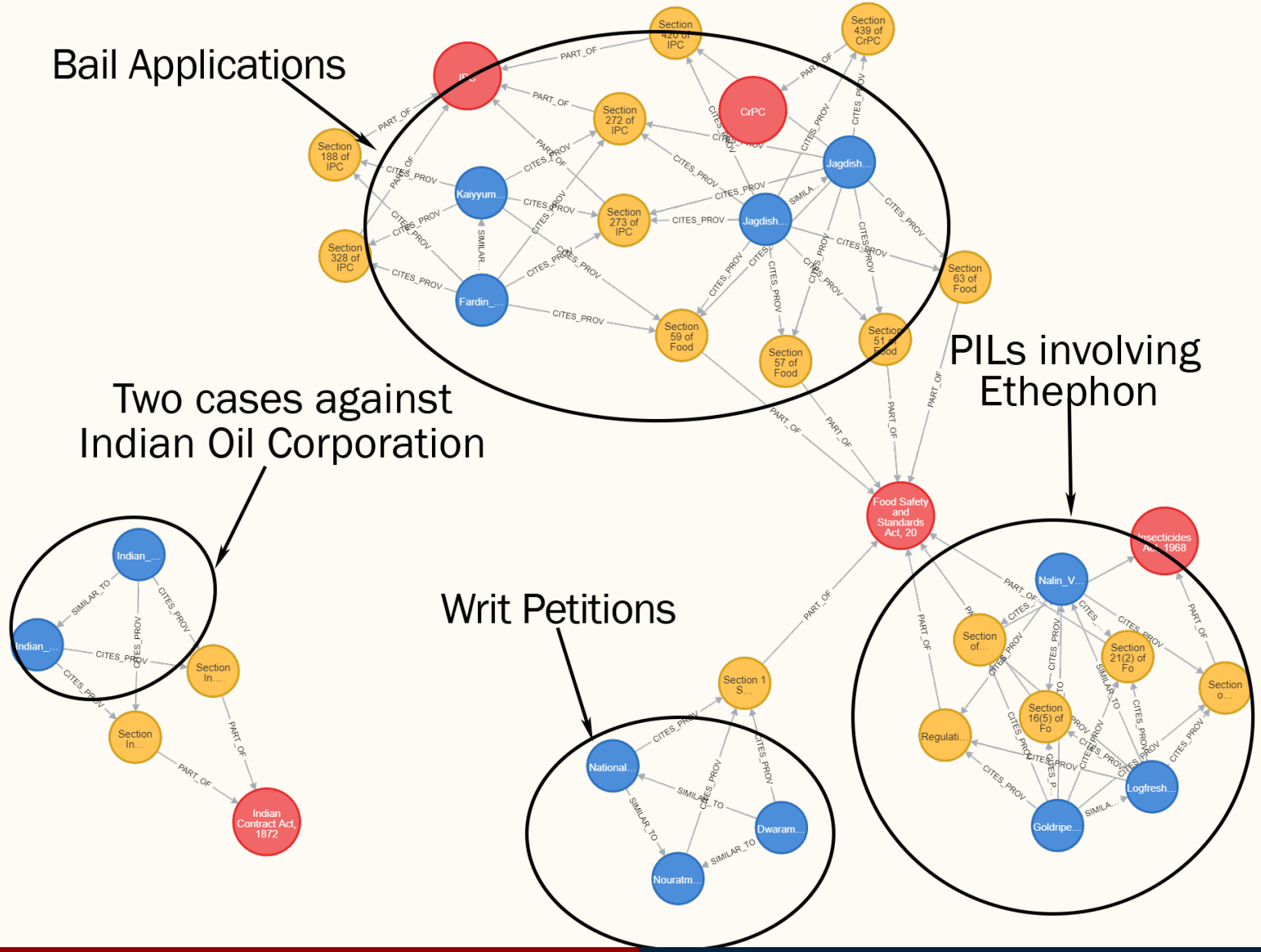


Figure: Graph of the three most similar cases based on citation count

Insights: Clustering using Jaccard Similarity

Used Jaccard Similarity to group together highly similar cases.



Limitations & Future Work

Dataset Scope

- Analysis was limited to food safety cases between January 2022 and December 2023.
- Many cases collected were bail applications rather than initial case hearings.
- A broader dataset could provide more generalizable insights.

Knowledge Graph Complexity

- Current graph focuses on basic relationships (e.g., citations, provisions).
- More detailed relationships and metadata could improve analysis.

Accessibility

- Knowledge graph querying requires technical knowledge of Neo4j and Cypher.
- Natural language querying or Graph-augmented LLM.

Conclusion

- Used LLM to populate a Knowledge Graph to analyze legal texts.
- Demonstrated usefulness of LLMs for entity extraction from long texts and compared performance against traditional NER model and ground truth.
- Extracted insights from KG generated.
- **Looking Forward:**
  - Expand datasets and include more entity types in the KG.
  - Explore more insights that can be extracted.
  - Integrate natural language interface for querying.

Thank you!

Questions?