Constructing a Knowledge Graph from Indian Legal Domain Corpus

Sarika Jain[1], Pooja Harde[1], Nandana Mihindukulasooriya[2], Sudipto Ghosh[3], Abhinav Dubey[3] and Ankush Bisht[3]

[1]*National Institute of Technology Kurukshetra, India*
[2]*IBM Research, Dublin, Ireland*
[3]*University of Delhi, India*

Abstract

While being an important pillar of human society, legal domain consists of large corpora of complex documents about different aspects such as laws or court judgements. In recent years, knowledge graphs have become a prominent solution to represent such complex information in semantically rich machine readable manner allowing access to other AI powered downstream applications. In this work, we aim to construct a reliable knowledge graph from Legal domain corpus that may be utilized by researchers and the application developers working in legal domain.The source dataset chosen is the Indian Legal Court Judgements and NyOn[1] (Nyaya Ontology) has been utilized for conceptualization. A framework that consists of entity extraction, relation extraction, triple construction is used to convert the legal text into RDF triples. The knowledge graph thus built has been quantitatively evaluated over a small random sample with reasonable results.

Keywords

Knowledge Graph Construction, Entity Extraction, Relation Extraction, Legal Domain

1. Introduction

There is a boom in digitizing the legal domain for use cases like classifying judgments [1, 2, 3], predictions [4, 5], questions answering [6, 7], finding similarities between judgments, and many more. Although much data is available in the legal domain, from court judgments to acts and deeds, its unstructured nature makes it inefficient and costly to process it to useful results. Accessibility and transparency of the key information have always been an issue. There is a requirement for a central place to keep this vast knowledge, and that too in an interoperable format.

With recent technological advances, AI applications can now process, understand and interpret the human language. Knowledge graphs [8] can represent large volumes of knowledge with their semantics providing easy access and structured querying abilities. Furthermore, when modeled following the Semantic Web standards, they can be used to reason, infer new

[1] https://w3id.org/def/NyOnLegalOntology#

Text2KG 2022: International Workshop on Knowledge Graph Generation from Text, Co-located with the ESWC 2022, May 05-30-2022, Crete,Hersonissos, Greece

jasarika@nitkkr.ac.in (S. Jain); pmharde29@gmail.com (P. Harde); nandana@ibm.com (N. Mihindukulasooriya)
https://sites.google.com/view/nitkkrsarikajain/ (S. Jain)

information, or find any inconsistencies in data. Moreover, such knowledge graphs enable downstream applications such as question answering, dialogue, prediction, and classification systems.

In the literature, we find different models for creating a knowledge graph from unstructured documents. In recent years, neural networks such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and Sequence-to-Sequence (Seq2Seq) models have shown promising results when used in the field of natural language processing (NLP) for sentiment analysis, information retrieval, and document classification. The state-of-the-art Information Extraction (IE) models use a supervised machine learning method that requires a huge amount of labeled training data. For general open-domain IE, publicly available labeled data is present, but domain-specific IE such as legal domain requires time-consuming data labeling tasks to create a large amount of training data. Although much work has been done in the legal domain for IE, the dataset available is not quite useful for every region. As the judicial system changes from country to country, so do changes in the legal terms, facts, artifacts, and law. There are many legal datasets available like in Chinese [9] and German [10] [1]. All these are specific to their region law; therefore, other regions beyond their boundaries cannot use it. For countries like India, where digitization of the judicial system is an ongoing process, it is challenging to find any proper dataset that will be useful for Information extraction. According to [11], creating the annotated data for a large corpus is time-consuming and too expensive. Moreover, while typing during the hearings in the courts, the court judgments get noisy due to misspellings, wrong placement of punctuation, and non-uniformity in the document structure. As for the linguistic part of the document, every judge has their vocabulary and interpretation; therefore, it becomes more difficult for the non-domain experts to comprehend legal text corpora and develop a dataset that satisfies all the system's requirements to process the documents with the uniform annotated data.

This work aims to explore a viable rule-based approach to systematically extract, annotate, and store the key entities of the legal domain as a knowledge base. The first step is to extract entities from the legal corpora using Information Extraction (IE) involving Named Entity Recognition (NER) and Relation Extraction (RE). The second step is to store, maintain, and update the retrieved knowledge in a graph database in a set of triples. As metadata, we have developed NyOn[1] (Nyaya Ontology) [12], a multilingual modular ontology for legal court judgments. NyOn has been developed taking entities from the Indian Supreme Court judgments taken from indiankanoon.org[2] and is currently available in five languages (English, German, French, Spanish, and Hindi). NyOn is referred as the schema required for information extraction. It gives the broader aspect of the types of entities and the relations we need to identify from the legal court judgment documents. The knowledge graph thus constructed will be put to question answering to answer questions like:

- What are the different courts that are referred to in a specific case?
- List of documents that fall under a specific type of jurisdiction?
- Who were the parties involved in Union Of India vs Ex No. 3192684 W Sep. Virendra judgment on 7 January 2020?

[2] https://indiankanoon.org/

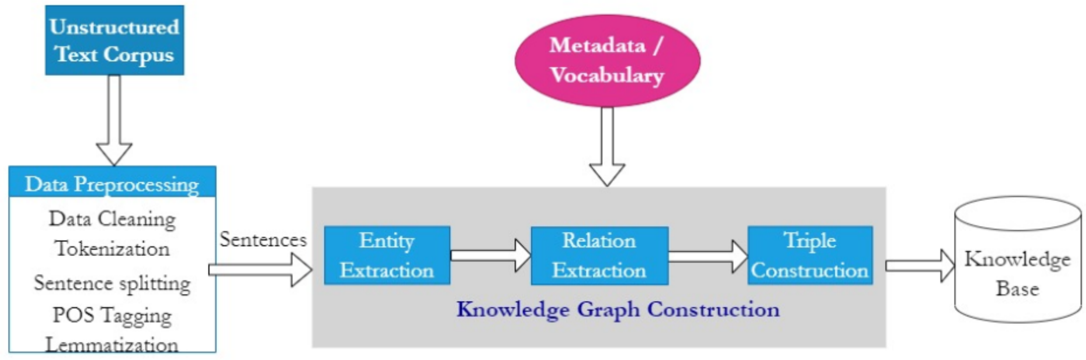


Figure 1: Architecture for Knowledge Graph Construction

- How many judgments were passed in 2002?
- List all judgments ordered by CJI N V Ramana.
- What are the names of the witnesses in a specific case?

The main contributions of this paper are (1) An analysis of existing approaches for information extraction and knowledge graph construction, and (2) Construction of a knowledge graph for the Indian Supreme Court judgments and a quantitative evaluation of the proposed framework. The dataset and the source code are publicly available on GitHub[3]. The rest of the paper is organized as follows. Section 2 provides a approach followed for constructing the knowledge graph from Indian Supreme Court Judgments, Section 3 provides the evaluation results, while Section 4 discusses the related work focusing on legal knowledge graph construction. Section 5 elaborates the conclusions derived from the work and presents future work ideas.

2. Construction of Knowledge Graph of Indian Supreme Court Judgements

In this paper, we use the Indian Supreme Court Judgements to create the Knowledge Graphs, which will facilitate downstream applications such as question answering, judgment prediction, etc. Figure 1 demonstrates the general architecture of the knowledge graph construction approach we followed. It mainly consists of a three-step pipeline consisting of data prepossessing, entity extraction, relation extraction, and triple construction steps. Each of these steps will be described in the following subsections.

2.1. Dataset

The input corpus to our Knowledge Graph construction pipeline contains 44,366 reportable and non-reportable court judgments published by the Supreme Court of India from 1947 to

[3] https://github.com/semintelligence/Text2KG

2020 in plain text. The documents have been scraped using a Python-based scraper from IndianKanoon.org, a free-to-use case search engine for Indian legal documents sourced from the various court and tribunal repositories. The dataset comprises court judgments for cases from various legal subdomains (civil law, criminal law, property law, etc.). All text documents in the dataset are assumed to be in English language.

Fig. 2 refers to a snapshot of one Indian Court Decision document. We call this data the caption data as no sentences are present in this part of the document. The text highlighted in orange represents the data, and the text highlighted in cyan color represents the entity type. Even though this data is not present in the sentence format, it represents most of the legal court decision documents useful for the knowledge graph creation. In the relation extraction part of the paper, we have discussed how we identify the relationship between the available entities in the caption format.

2.2. Data Preprocessing

After selecting documents, we perform the data preprocessing on the corpus. In the GATE[4] software (General Architectural of Text Engineering), we already have the Language Processing tools in ANNIE[5] (A Nearly-New Information Extraction System) such as POS Tagger, ANNIE Sentence Splitter, ANNIE English Tokenizer, etc. All these processing tools are preloaded into the software for data preprocessing.

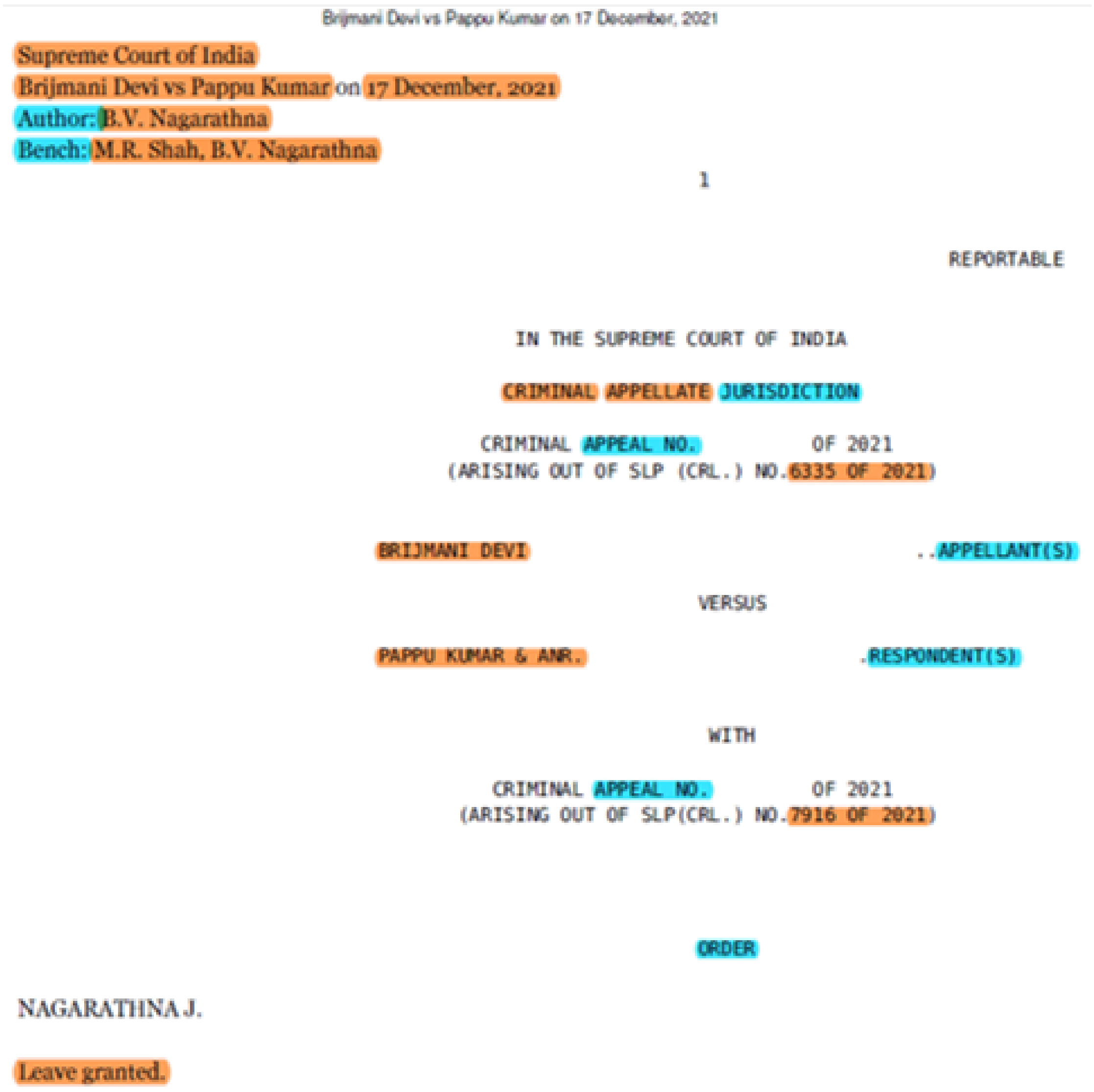
2.3. Entity Extraction

Entity extraction is the step performed after data preprocessing for information retrieval. Here all the useful entities are identified from the unstructured data that forms relevant, useful information about the context of the document. After the data is scraped from indiankanoon.org[2], the rules are written in JAPE (Java Annotation Patterns Engine) for the GATE software (General Architectural of Text Engineering) with the help of NyOn [12] Ontology for recognizing the entities in the documents. As discussed in the above subsection, these rules are given as a JAPE Transducer to ANNIE as a processing resource with other processing resources like POS tagger and Sentence Tokenizer. While annotating the entities using GATE API, we observed that it takes approximately one hour to annotate 100 documents as the length of the documents is quite long. One of the rules used for entity recognition is listed in Table 1.

After the data is annotated using the JAPE rules, the corpus is stored with the annotation state having the inline XML format of all the documents. We use a small Python script to convert all the inline XML documents into standard text documents. The converted standard text documents are then saved separately for the next phase of the system, which is relation extraction. The list of the entities identified in the court decision documents is presented in Table 2.

[4] https://gate.ac.uk/

[5] https://gate.ac.uk/sale/tao/splitch6.html



Brijmani Devi vs Pappu Kumar on 17 December, 2021

Supreme Court of India
Brijmani Devi vs Pappu Kumar on 17 December, 2021
Author: B.V. Nagarathna
Bench: M.R. Shah, B.V. Nagarathna

1

REPORTABLE

IN THE SUPREME COURT OF INDIA

CRIMINAL APPELLATE JURISDICTION

CRIMINAL APPEAL NO. OF 2021
(ARISING OUT OF SLP (CRL.) NO.6335 OF 2021)

BRIJMANI DEVI ..APPELLANT(S)

VERSUS

PAPPU KUMAR & ANR. .RESPONDENT(S)

WITH

CRIMINAL APPEAL NO. OF 2021
(ARISING OUT OF SLP(CRL.) NO.7916 OF 2021)

ORDER

NAGARATHNA J.

Leave granted.

Figure 2: Court Case Document Snapshot with text in orange highlighting the data and text in cyan highlighting the entity.

2.4. Relation Extraction

The entities extracted in the previous phase are passed as input in this phase. Here the relations are identified between the extracted entities. The list of the various relations that are identified between different entity types (Object Properties) are listed below in Table 3.

As discussed in the earlier section, in the caption data portion of the document, there are no sentences in between the entities; we use a small code to map the relation between such entities and the data with no text around. To refer to the type of relation, we refer to the NyOn

Table 1
JAPE Rule for the entity BENCH

```
Rule: BENCH
Priority: 20
(
  (
    {Token.string == ”BENCH”}
  ):Type
  {Token.string == ”:”}
  (
    {Token.kind == ”word”, Token.category != ”CC”} |
    {Token.string == ”.”, Token.category != ”CC”}
  )+ :Name
  ({Token.kind == ”punctuation”})?
  ((
    {Token.kind == ”word”, Token.string != ”judgment”} |
    {Token.string == ”.”, Token.sting != ”judgment”}
  )+) :Name2
)
–>
:Type.COURT_OFFICIALS = {kind = ”BENCH”, rule = ”BENCH”},
:Name.JUDGE = {rule = ”BENCH”},
:Name2.JUDGE = {rule = ”BENCH”}
```

Ontology, where the relation between the entities has been described. The NyOn ontology works as a base for us for entity extraction and relation extraction from the data. The list of relations between the entities and the values (Datatype Properties) is also provided in Table 4.

2.5. Triple Construction

A knowledge graph consists of triples where every triple is represented in the form of subject-predicate-object. After extracting the entities and the relations from the data, the next step is to create the triple to form a knowledge graph. The extracted entities and relations are stored corresponding to each other in the form of lists. $E = \{e_1, e_2, e_3, ..., e_n\}$ represents the entity set and $R = \{r_1, r_2, r_3, ...r_m\}$ represents the relation set. To form a triple, we follow the given format $T = \{e_a, r_k, e_b\}$ where $\{e_a, e_b \epsilon E, r_k \epsilon R\}$.

Once the RDF model is constructed, it can be materialized into any RDF serializations such as RDF/XML, Turtle, or N Triples. The triples then can be loaded into an industry-standard triple store such as Blazegraph, Apache Jena, and Virtuoso and queried through a SPARQL endpoint. We are also currently working on exposing such triples as Linked Open Data with dereferenceable HTTP URIs.

Table 2
Entities used to Annotate Dataset

Entity Labels	Description
BENCH	Bench of Judges delivering the judgment
CASE	Identifier of the form A v. B
CASE_TYPE	Civil or Criminal
COURT	Judicial Entity
COURT_DECISION	Orders in the judgment
CRIME_VIOLATION	Instances of crimes and violations
CUSTODY	Instances of judicial or police custody
DOCUMENTS	Appeal, Petition, FIRs, etc.
EVIDENCE	Weapons, Documents
JURISDICTION	Original, Advisory, Appellate, Review
LAW	Instances of Acts, IPC, CrPC sections
PARTIES	Plaintiffs, Judges, Parties in judgment
PARTY_TYPE	Individual, Organization, State, Government, etc
COURT_OFFICIALS	Legal people involved in CASE (Judge, Solicitor, etc)
LOCATION	Geographical location of State, District, Village, Place (for Evidences)
DATE	Documented and relevant dates

Table 3
Object properties used to annotate relations between entities

Relation Labels	Entity Type 1 (domain)	Entity Type 2 (range)
hasCourtOfficials	CASE	COURT_OFFICIALS
hasParties	CASE	PARTIES
hasPartyType	PARTIES	PARTY_TYPE
hasBench	CASE	BENCH
hasAuthor	CASE	AUTHOR
hasCourt	CASE	COURTS
hasActs	CASE	ACTS
hasEvidences	CASE	EVIDENCES
documentType	CASE	DOCUMENTS
hasJurisdiction	COURT	JURISDICTION
hasLocation	COURT EVIDENCE	LOCATION LOCATION
isA	PRECEDENT_CASE AUTHOR	CASE JUDGE
caseBelongsTo	CASE	CASE_TYPE

3. Evaluation

This section will present the preliminary evaluation of the proposed approach. This evaluation focused on two aspects of the knowledge base construction process: named entity extraction and relation extraction of Indian Court Judgement documents.

We have evaluated both the rule-based Named Entity Recognition and Relation Extraction components of the proposed approach. As there was no academic benchmark for the type

Table 4
Datatype properties used to annotate relations between entities and values

Relation Labels	Entity Label	Values
hasCaseNumber	CASE	string
hasCaseName	CASE	string
hasPartiesName	PARTIES	string
hasCourtName	COURTS	string
hasCourtOfficialsName	COURT_OFFICIALS	string

of documents in our use case, we have manually created a gold standard with a sample of documents with the help of human annotators who are domain experts.

3.1. Gold Standard Creation

There were no existing academic benchmarks for testing the entity and relation extraction on the Indian court judgment documents. Thus, we decided to create a gold standard for evaluating the performance of our proposed approach to entity extractions and relation extraction. We randomly selected five documents from the corpus and annotated them for gold entities and relations with the help of domain experts. The NyOn ontology [12] guided the gold entity types and gold relations that were annotated. Two annotators annotated each document to ensure the inter-annotator agreement, and any conflicts were resolved using a third domain expert. The annotated documents contained 363 named entities of 10 different class types and 154 triples of 6 relation types. We plan to increase the size of the gold standard for future work.

Metrics: As evaluation metrics, we have used commonly used precision-recall measures. We have a set of gold annotations and machine-generated output from our rule-based pipeline for both entities and relations. Table 5 shows some example output of rule-based machine NER, including correctly and incorrectly identified entities. In this context, the precision was calculated as the percentage of the machine-generated entities/relations that were correct according to the gold annotations and recall as the percentage of total gold entities/relations identified in the machine output.

3.2. Named Entity Recognition

Table 6 shows the results of NER by individual documents level, and Table 7 shows the result for each class type level. We notice that our rules identify entities with high precision but suffer from recall issues. Furthermore, we have noticed that there is a wide variation of the recall values across different documents as well as different entity types. Some types, such as dates higher recall, while others, such as court decisions or law, have a lower recall. Based on these results, we are performing error analysis and working on improving the recall of our rules to capture most of the relevant entities.

Table 5
An example snippet of rule-based NER output

Entity text span	Entity Type	Correct
ASSTT. GEN. MANAGER CENTRAL BANK OF INDIA ETC.	PARTY_TYPE	✔
PETITIONER	PARTY	✔
COMMISSIONER MUNICIPAL CORPORATION AHMEDABAD ETC. ETC.	PARTY_TYPE	✔
RESPONDENT	PARTY	✔
09/05/1995	DATE	✔
BENCH	BENCH	✔
B.P. JEEVAN REDDY	PARTY_TYPE	✔
SUJATA V. MANOHAR	PARTY_TYPE	✔
Bombay Rents, Hotel and Lodging House Rates Control Act, 1944	LAW	✔
Gujarat High Court	COURT	✔
K. BENCH	PARTY_TYPE	✖
M.P. Act and that the said non-obstante clause makes all the difference. Dewan Daulat Rai Kapoor arose under the Punjab Municipal Act, 1911	LAW	✖

Table 6
NER Evaluation by Document

Document No.	Total Gold Entities	Identified Entities	Correct Entities	Precision	Recall	F1
1592579	84	36	36	1	0.428	0.599
1592674	124	45	39	0.866	0.314	0.461
1592725	80	15	14	0.937	0.175	0.295
1592769	56	26	26	1	0.464	0.634
1592785	19	15	15	1	0.78	0.876

3.3. Relation Extraction

Table 8 shows the results of relation extraction by individual documents level, and Table 9 shows the result by each relation type level. In contrast to entity extraction, here we see that both precision and recall have been affected by our rules. The results are expected because relation extraction is a more complex task than entity extraction, and entity extraction recall issues are propagated to relation extraction. Furthermore, there are several relation types, such as *hasDecision* and *hasJudge*, for which we have not implemented the rules.In future work, we plan to perform error analysis to improve the current set of rules and implement new rules to cover the relation types that are not currently covered.

Table 7
NER Evaluation by Entity Type

Entity Type	Total Gold Entities	Identified Entities	Correct Entities	Precision	Recall	F1
LAW	85	22	16	0.7	0.188	0.296
COURT	69	26	26	1	0.3768	0.547
PARTICIPANT_TYPE	69	16	16	1	0.23	0.374
PARTICIPANT	58	26	26	1	0.44	0.611
DATE	50	50	50	1	1	1
BENCH	10	2	2	1	0.2	0.333
COURT_DECISION	10	1	1	1	0.1	0.182
DOCUMENTS	4	0	0	0	0	0
JURISDICTION	2	1	1	1	0.5	0.667

Table 8
Relation Extraction Evaluation by Document

Document No.	Total Gold Triples	Identified Triples	Correct Triples	Precision	Recall	F1
1592579	16	9	7	0.78	0.44	0.56
1592674	63	35	19	0.54	0.30	0.39
1592725	36	8	6	0.75	0.17	0.28
1592769	26	9	6	0.67	0.23	0.34
1592785	13	7	6	0.86	0.46	0.60

Table 9
Relation Extraction Evaluation by Relation Type

Relation Type	Total Gold Triples	Identified Triples	Correct Triples	Precision	Recall	F1
hasPartyName	30	25	23	0.92	0.76	0.83
hasCourtName	15	15	10	0.66	0.66	0.66
hasLaw	44	20	12	0.60	0.28	0.38
hasPartyType	29	15	13	0.87	0.45	0.59
hasDecision	10	0	0	0	0	0
hasCourtOfficial	10	8	5	0.62	0.5	0.55

4. Related Work

Information extraction approaches are emerging and are making their space in various domains, whether biomedical, social-media, legal, etc. These approaches can play an auspicious role in building an enriched knowledge base, specifically for legal domains in maintaining the huge volume of legal information scattered on various portals. Information extraction approaches can also provide a way for developing various applications like question-answering systems for the legal domain, judgment prediction, dispute resolution, etc. The most commonly used approaches for information extraction are; first finding the named entities present in the unstructured data and then finding the relation between those entities that have been identified. Named Entity

Recognition (NER) is somewhat challenging in the legal domain due to various legal terms, abbreviations, references made to the Acts and Laws, etc. Then after identifying these domain-specific entities, relation extraction (RE) is also of great importance as it identifies the relations between any two entities, which leads to triple construction and later knowledge graph creation.

Various review and survey articles on NER and RE focus on standard datasets, and even if most of the research work is done in the legal domain, the available dataset is not useful for the Indian context. We come across articles that focus on information extraction from unstructured text in the context of legal documents that do not cover the Indian context mostly. It is also discussed by Fernàndez-Cañellas et. al. [13] that Natural Language Processing (NLP) alone cannot guarantee the validity of the facts to be populated in the knowledge graph and data validation methods also need to be taken into account. In the legal context, this might mean the validity of the legal facts about judgments, cases, and stakeholders in the Knowledge Graphs (KG).

The relationship extraction depends highly on the syntactic and semantic analysis of sentences in a rule-based approach. Dragoni et al. [14] discussed how to combine the NLP approaches for rules extractions in the legal domain. The author uses the Deontic Lightweight Ontology called normonto, which represents and models the legal concepts and specifies the lexicons used for legal expressions like permission, prohibition, and obligation. Thomas and Sangeetha in [15] discuss rule-based entity extraction represented as regular expressions to detect entity mentions in Indian judicial texts using specific patterns or trigger words. The GATE tool and Java Annotation Patterns Engine (JAPE) grammar rules can support rule-based extraction; however, basic entities must be identified first. Andrew and Tannier in [16] take a hybrid approach by using a statistical Conditional Random Field (CRF) model and legal domain-specific JAPE rules in GATE gazetteer to annotate their dataset. Eftimov et. al. [17] proposes the rule-based approach for NER identification for evidence-based dieatry recommendations. Poudyal et. al. [18] also proposes a rule-based approach for NER and Relation Extraction. The author made use of the C programming language for rules extraction.

To sum up, as the paper focuses on the rule-based approach for named entity recognition and relation extraction for legal court judgment documents, we discussed various research work done in this area and how we can enhance the NER and Relation extraction for the legal domain by writing JAPE rules in context to India Judicial System. There exist many differences in the existing rule-based approaches, like [15] does not talk about the jurisdiction entity, and the paper focuses only on the Criminal Cases. [14] take the entity that expressions only legal expressions and not all the information related to the case document. [18] considers only entities related to the plaintiff, court, court staff, and decision. The authors did not consider other entities like petitioner, case jurisdiction, documents, evidence, etc. In this paper, we focus on extracting many such entities that are not considered in the works presented till now and making the information extraction comprehensive for the knowledge graph creation. Currently, we have focused only on the rule-based approach, but as the next step, we will use the Machine Learning and Deep Learning models for NER and Relation Extraction for the legal court decision documents.

5. Conclusion

In this paper, we have presented a pipeline for constructing Knowledge Graphs from the Indian Supreme Court Decisions corpus. The pipeline consists of data preprocessing, entity extraction, relation extraction, and triple construction. The generated triples are represented in RDF using the NyOn ontology and are stored in a triple store. The Knowledge Graph can be queried using SPARQL and can be used to build downstream applications such as Knowledge Base Question Answering with complex questions such as the ones that require aggregations or multi-hop reasoning (which are not possible with simple document retrieval or keyword search). In addition, the generated Knowledge Graph can become a useful resource for other AI tasks such as judgment predictions, case clustering, and classification.

The results are reasonably good but still have room for improvement. We plan to improve our rules to increase recall in entity recognition and both precision and recall in relation extraction in future work. We are also planning to explore neural approaches based on semi-supervised learning by using distant supervision data or using the output of the rule-based system as training data. Furthermore, we are planning to publish the generated data as Linked Data with public access and develop a semantic web portal to expose the knowledge base to various use cases.

Acknowledgments

This work is supported by the IHUB-ANUBHUTI-IIITD FOUNDATION set up under the NM-ICPS scheme of the Department of Science and Technology, India

References

[1] A. Elnaggar, C. Gebendorfer, I. Glaser, F. Matthes, Multi-task deep learning for legal document translation, summarization and multi-label classification, in: Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference, 2018, pp. 9–15.

[2] O.-M. Sulea, M. Zampieri, S. Malmasi, M. Vela, L. P. Dinu, J. Van Genabith, Exploring the use of text classification in the legal domain, arXiv preprint arXiv:1710.09306 (2017).

[3] N. Ramrakhiyani, S. Pawar, G. K. Palshikar, A system for classification of propositions of the indian supreme court judgements, in: Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation, 2013, pp. 1–4.

[4] K. D. Ashley, Artificial intelligence and legal analytics: new tools for law practice in the digital age, Cambridge University Press, 2017.

[5] M. Medvedeva, M. Vols, M. Wieling, Using machine learning to predict decisions of the european court of human rights, Artificial Intelligence and Law 28 (2020) 237–266.

[6] G. McElvain, G. Sanchez, S. Matthews, D. Teo, F. Pompili, T. Custis, Westsearch plus: A non-factoid question-answering system for the legal domain, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1361–1364.

[7] C. Hoppe, D. Pelkmann, N. Migenda, D. Hötte, W. Schenck, Towards intelligent legal advisors for document retrieval and question-answering in german legal documents, in: 2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), IEEE, 2021, pp. 29–32.

[8] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. d. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al., Knowledge Graphs, Synthesis Lectures on Data, Semantics, and Knowledge 12 (2021) 1–257.

[9] W. Huang, D. Hu, Z. Deng, J. Nie, Named entity recognition for chinese judgment documents based on bilstm and crf, EURASIP Journal on Image and Video Processing 2020 (2020). doi:10.1186/s13640-020-00539-x.

[10] E. Leitner, G. Rehm, J. Moreno-Schneider, Fine-grained Named Entity Recognition in Legal Documents, in: M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, Y. Sure-Vetter (Eds.), Semantic Systems. The Power of AI and Knowledge Graphs. Proceedings of the 15th International Conference (SEMANTiCS 2019), number 11702 in Lecture Notes in Computer Science, Springer, Karlsruhe, Germany, 2019, pp. 272–287. 10/11 September 2019.

[11] V. Malik, R. Sanjay, S. K. Nigam, K. Ghosh, S. Guha, A. Bhattacharya, A. Modi, Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation, in: Proceedings of the 2021 Conference on The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), Association for Computational Linguistics, Bangkok, Thailand (Online), 2021.

[12] S. Jain, P. Harde, N. Mihindukulsooriya, NyOn, 2022. URL: https://github.com/semintelligence/NyOn.

[13] D. Fernàndez-Cañellas, J. Marco Rimmek, J. Espadaler, B. Garolera, A. Barja, M. Codina, M. Sastre, X. Giro-i Nieto, J. C. Riveiro, E. Bou-Balust, Enhancing online knowledge graph population with semantic knowledge, in: International Semantic Web Conference, Springer, 2020, pp. 183–200.

[14] M. Dragoni, S. Villata, W. Rizzi, G. Governatori, Combining Natural Language Processing Approaches for Rule Extraction from Legal Documents: AICOL International Workshops 2015-2017: AICOL-VI@JURIX 2015, AICOL-VII@EKAW 2016, AICOL-VIII@JURIX 2016, AICOL-IX@ICAIL 2017, and AICOL-X@JURIX 2017, Revised Selected Papers, volume 10791, 2018, pp. 287–300. doi:10.1007/978-3-030-00178-0_19.

[15] A. Thomas, S. Sangeetha, An innovative hybrid approach for extracting named entities from unstructured text data, Computational Intelligence 35 (2019) 799–826.

[16] J. J. Andrew, Automatic extraction of entities and relation from legal documents, in: Proceedings of the Seventh Named Entities Workshop, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1–8. URL: https://aclanthology.org/W18-2401. doi:10.18653/v1/W18-2401.

[17] T. Eftimov, B. Seljak, P. Korošec, A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations, PLoS ONE 12 (2017). doi:10.1371/journal.pone.0179488.

[18] P. Poudyal, P. Quaresma, An hybrid approach for legal information extraction, Frontiers in Artificial Intelligence and Applications 250 (2012) 115–118. doi:10.3233/

978-1-61499-167-0-115.