LeGen: End-to-end Legal Information Extraction using Generative Models

Anonymous ACL submission

Abstract

Despite the rapid growth in access to digital devices, the new users of the devices, especially in developing countries like India, are not able to access information on their rights and entitlements, jobs and livelihood, healthcare, education, etc. as the information is in the form of very long, complex sentences and heavy in legal parlance. Open information extraction techniques can be used to convert unstructured legal text into triples of the form ⟨`subject`, `relation`, `object`⟩ in a domain-independent manner. However, the legal text is long and complex which calls for extracting structure beyond triples, also called complex information extraction. This paper proposes a generative approach to perform complex information extraction from legal statements. We achieve this by encoding legal statements as trees to capture their complex structure and semantics. This end-to-end modeling reduces the propagation of errors across complicated pipelines. We experimented with multiple generative architectures to conclude that our proposed approach reports up to 14.7 % gain on an Indian Legal benchmark and is competitive on open information extraction benchmarks.

1 Introduction

The number of people with access to smartphones and other computing devices is on a constant rise. Some data sources point to there being a 71% reach of smartphones in 2023 [39, 17]. This should lead to greater access to information and data for a large part of the population, however we observe that the new users of digital devices (often called the Next Billion Users) [16] is not able to leverage the access to devices to access information about their rights and entitlements, jobs, livelihood, health or education. One primary reason for this phenomenon is that information in these domains, if they exist, exist in textual formats, in legal parlance, with long and complex sentence structures [1]. Understanding the textual information and taking action on them puts substantial cognitive load on the new users, who often do not have the educational training and agency to consume and act on the information [20].

NLP techniques can assist in structuring and organizing legal data to enable automatic search and retrieval [11, 43]. Open information extraction (OIE) techniques [23, 38, 13] can be used to extract structured information such as triples of the form ⟨`subject, relation, object`⟩ from a sentence in a domain-independent manner. However, legal text poses unique challenges - Legal sentences and documents are lengthy with complex inter-clausal relationships between them [8]. Existing OIE techniques are unable to return the best results on legal sentences. For instance, the output of OpenIE6 [23] on *If over 50 percent of a company's workers take concerted casual leave, it will be treated as a strike* are 2 triples - $i)$ ⟨`it, will be treated, as a strike`⟩, $ii)$ ⟨`over 50 percent of a company's workers, take concerted, casual leave`⟩. The model fails to identify that a `condition` connects the two extractions. Apart from `condition`, clauses can have relations such as `contrast` or `disjunction`, etc (Table 1) among them. Identifying such relations is important to design systems that empower users interpret complex legal information.

The problem of extracting structure beyond triples is handled by a relatively new area of research known as complex information extraction [26]. However, most of these techniques [32, 33] involve multiple-step pipelines for identifying clauses and relationships between them that propagate errors. They also lack language understanding and generalization capabilities. This paper proposes *LeGen*, an end-to-end generative approach for complex information extraction from legal sentences. Generative architectures, such as T5 [35], BART [25], or GPT [34] have been very successful in understanding text and generalization. By encoding legal sentences as a discourse tree

Sentence	Clauses	Relations	Relations among Clauses
If over 50 percent of a company's workers take concerted casual leave, it will be treated as a strike	1) Over 50 percent of a company's workers take concerted casual leave 2) It will be treated as a strike	CONDITION	$R_{CONDITION}$ (Over 50 percent of a company's workers take concerted casual leave, It will be treated as a strike)
A non-resident can open an NPS account, but the account will be closed if the citizenship status of the NRI has been changed.	1) A non-resident can open an NPS account 2) The account will be closed 3) The citizenship status of the NRI has been changed	CONTRAST, CONDITION	$R_{CONTRAST}$(A non-resident can open an NPS account, $R_{CONDITION}$(The account will be closed, The citizenship status of the NRI has been changed))
If balance amount in the account of a deceased is higher than 150,000 then the nominee or legal heir has to prove the identity to claim the amount	1) Balance amount in the account of a deceased is higher than 150,000 then 2) The nominee has to prove the identity to claim the amount 3) Legal heir has to prove the identity to claim the amount	CONDITION, DISJUNCTION	$R_{CONDITION}$(Balance amount in the account of a deceased is higher than 150,000 then, $R_{DISJUNCTION}$(The nominee has to prove the identity to claim the amount, Legal heir has to prove the identity to claim the amount))

Table 1: Examples of clauses and relations CAUSE, CONDITION, CONTRAST, and DISJUNCTION among clauses

[32], (Section 4.1) we use BART and T5 architectures to capture both the structure and semantics of a complex sentence more accurately. Such end-to-end modeling reduces the propagation of errors across multiple steps. Our salient contributions are:

1. We introduce the problem of information extraction for Indian Law
2. We introduce the idea of using complex information extraction for legal statements
3. We propose *LeGen*, an end-to-end generative approach for legal information extraction using a novel tree-based encoding technique
4. We release a new benchmark for legal information extraction, curated from Indian Law statements
5. We report substantial gain over Graphene [32], a state-of-the-art complex information extraction technique on the Indian Legal benchmark.
6. We show *LeGen*'s flexibility by training it as an OIE task, and conclude that it is competitive on an OIE benchmark.

Our paper is organized as follows. In Section 2, we discuss work related to legal, complex, and open information extraction. We formally describe the problem in Section 3 and introduce *LeGen* in Section 4. We discuss our experiments and results in Section 5 and 6 and discuss future work in Section 7. The limitations of our approach are described in Section 8.

2 Related Work

2.1 Legal Information Extraction

As mentioned in [11], NLP or machine learning can be applied to legal research for multiple tasks not limited to finding information relevant to a legal decision [2, 30, 6], contract review (checking that a contract is complete and avoids risk) [7, 24], legal entity recognition [4], generating legal document – includes legal systems that generate legal documents by filling the blanks in the already existing templates and another kind in which, based on set of questions asked by the system, a tailored or custom made legal document is produced, and providing legal advice using QA system [1]. Such contributions have been made to both Indian and non-Indian legal systems.

In India, various efforts have been made to automate the judicial pipeline. The SemEval task [31] introduced 3 problems to be tackled on the ILDC corpus [27]. – i) legal named entity recognition [21] performs named entity recognition on the ILDC corpus, ii) rhetorical role prediction structures legal transcripts into rhetorical roles [22] and iii) court case judgment prediction proposes using AI-based techniques to automate course case judgments. However, to the best of our knowledge, accurately extracting structure from unstructured legal sentences in the Indian Legal domain has not been studied.

Among the datasets, there is the Chinese legal dataset LEVEN [40] which detects legal events (charge-related events including general events in legal documents), the Indian Legal dataset, ILDC [27] containing Supreme Court cases annotated with court decisions which can be used for predicting justice and explanation, CaseHold [42] dataset comprising of multiple choice questions to identify the relevant cases, CUAD [19], an annotated legal data set for contract review and various others.

As mentioned above, these data sets and research's primary focus is understanding the court cases, judgments, prediction tasks, or segmentation. Our work focuses on extracting structural information from complex legal sentences.

2.2 Open Information Extraction

Open Information Extraction uses an independent paradigm to extract the information as a triple, $\langle$`subject, relation, object`$\rangle$. Banko et al.,[41] introduced the concept of Open Information Extraction and proposed Text Runner. Following this, many rule-based systems were developed like REVERB [13] and OpenIE5 [36]. Moving from rule-based system, we have RNNOIE [1] [38] which uses a neural-based approach to open information extraction and is trained by extracting non-neural systems.

The state-of-the-art in Open Information Extraction, OpenIE6 [2] uses iterative grid labeling with BERT architecture to generate triples from input sentences. It combines the results from the three models (coordination model, OIE model, and Allennlp models) to generate triples from input sentences.

2.3 Complex Information Extraction

Many OIE systems have been developed which cater to identifying triples in a complex sentence [26] like OLLIE [37], MinIE [15], ClausIE [12], StuffIE [33] and Graphene [5].

ClausIE [3] , MinIE [4] and OLLIE [5] uses a linguistic-based approach to information extraction. OLLIE open information system uses a set of pre-defined templates and rules to identify the relation present in the sentence. MinIE also uses a linguistic approach to extract information with a difference that enhances the output by adding other semantic information like polarity, modality, attribution, and quantities. StuffIE [33][6], another open information system that aims to extract complex information which is referred to as facets in this work, uses syntactical dependency to tag facets or relations in the sentence. Graphene [32] [7] uses 39 handcrafted rules to construct a discourse tree and then obtain the triples from the sub-sentences of the input sentences. These techniques are either rule-based or use a pipeline of techniques to extract the structure of a complex sentence. To the best of our knowledge, ours is the first attempt at using generative neural architectures to model complex information extraction.

3 Problem Definition

We use the sentences from Table 1 for demonstration. We denote them by $\mathcal{S}$. Our goal is to identify from $\mathcal{S}$:

[1] https://github.com/gabrielStanovsky/supervised-oie
[2] https://github.com/dair-iitd/openie6
[3] https://gate.d5.mpi-inf.mpg.de/ClausIEGate/ClausIEGate/
[4] https://github.com/uma-pi1/minie
[5] https://github.com/knowitall/ollie
[6] https://gitlab.inf.unibz.it/rprasojo/stuffie
[7] https://github.com/Lambda-3/Graphene

- A set C of all clauses in $\mathcal{S}$. A clause refers to an indivisible, atomic sentence in $\mathcal{S}$. C = {"it will be treated as a strike", "over 50 percent of a company's workers, take concerted, casual leave"} for the first sentence in Table 1.

- A set $COMP$ of complex sentences that are obtained either by i) combining N clauses using an N-ary relation, or, ii) by combining subsets of C and $COMP$ using N-ary relation.

- A set R of N-ary relations that relate N clauses or complex sentences and generate a new complex sentence. In other words, R_{r_i}: $\{C \cup COMP\}^N \longrightarrow COMP$, where $R_{r_i} \in R$. For $\mathcal{S}$, $R = \{R_{\texttt{condition}}\}$. The output of $R_{\texttt{condition}}$("it will be treated as a strike", "over 50 percent of a company's workers, take concerted, casual leave") is $\mathcal{S}$.

Three properties that should be satisfied by C, $COMP$ and R are:

- **Correct**: Every $c \in C$, $c' \in COMP$ and $r \in R$ should convey the same meaning as expressed in $\mathcal{S}$

- **Non-redundant**: C, R, and $COMP$ should not contain repeated information

- **Complete**: All information conveyed in the sentence should be expressed by C, R, and $COMP$

4 LeGen

We propose $LeGen$, an end-to-end generative model to perform complex information extraction from legal sentences. $LeGen$ is based on the idea of discourse trees which are defined in the next subsection. We model it as a generation task, that outputs discourse trees for a sentence.

The Discourse tree as proposed in Graphene [5, 32] employs a top-down approach to break longer text into smaller parts in contrast to the bottom-up approach employed for RST trees. Simplified sentences can not be decided beforehand because they're not consistent and may need changes (like rephrasing) depending on their specific sentence structures. An example of Discourse Tree structure is shown in Figure 1 (left). The leaf nodes are the clauses (defined in Section 3, 'Balance amount in the account of a deceased is higher than 150,000 then','The nominee has to prove the identity to claim the amount .' and 'Legal heir has to prove the identity to claim the amount .') . Each non-leaf node represents a complex sentence formed by combining the clauses represented by its children nodes. They are combined using the relation label on the non-leaf node, (SUB/-CONDITION, CO/DISJUNCTION). Relations in a discourse tree fall under two categories: co-ordinations and sub-ordinations.

4.1 Discourse Tree

Discourse Tree originated from Rhetorical Structure Theory (RST) [28]. RST identifies the hierarchical structure of the text and the rhetorical relations between the text parts. Rhetorical relations are split into classes of coordinates and subordinates and can be mapped to the span of text or words.

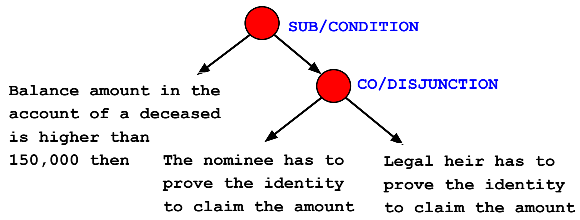
Co-ordinations. Coordinating sentences are a type of sentence structure in which two or more independent clauses are joined together using coordinating conjunctions. These clauses are typically of equal importance, and they are combined to create a more complex and informative sentence. Coordinating conjunctions are 'and',' or' and 'but'.

Sub-ordinations. Subordination sentences are a type of sentence structure in which one main or independent clause is combined with one or more subordinate or dependent clauses. These clauses are linked together to form a single sentence, with the main clause expressing a complete thought, while the subordinate clauses provide additional information, clarification, or context. Some of the subordinations are 'while', 'because', 'if', 'whenever', 'since' etc.

4.2 Generating Discourse Trees

Any existing rule-based approach can be used to generate the discourse trees for sentences. Currently, Graphene [32] generates discourse trees with good precision and recall. Graphene uses a set of 39 hand-crafted rules to identify 19 relations [5]. However, on analyzing these rules, we observed redundancies and inconsistencies. i) For instance, it is very difficult to distinguish between `BACKGROUND`, `ELABORATION`, or `EXPLANATION` relations. ii) the rules proposed for identifying `TEMPORAL_BEFORE` and `TEMPORAL_AFTER` relations from the text are not accurate. iii) Does not identify the date and named entities correctly . To ad-

If balance amount in the account of a deceased is higher than ₹150,000 then the nominee or legal heir has to prove the identity to claim the amount.




```
SUB/CONDITION('Balance amount in the
account of a deceased is higher than
150,000 then .', CO/DISJUNCTION('The
nominee has to prove the identity to
claim the amount .','Legal heir has to
prove the identity to claim the amount
.'))
```

Figure 1: Discourse tree for an example law sentence (on the left). Corresponding linear encoding of the Discourse tree (on the right). SUB and CO refer to subordination and coordination, respectively.

dress i) and ii), we merged BACKGROUND, ELABORATION, and EXPLANATION into ELABORATION. We converted TEMPORAL_BEFORE and TEMPORAL_AFTER into a single TEMPORAL relation. We didn't address iii), but we show in Section 6 that *LeGen* is robust to these issues. The final list of relations that were kept is in the Appendix.

4.3 Encoding of Discourse Tree

Figure 1 shows how we convert a discourse tree of any sentence into a sequence encoding. This allows complex information extraction to be simplified by expressing discourse trees as a sequence of text. We view it as a language translation task where the output language is the tree encoding. In the context of a translation task, teacher forcing utilizes pairs of text written in two different languages by influencing the generated text based on the provided input. During the training process, the encoder processes text in one language, while the decoder processes text in the other language and predicts the next token for each position. In our method, we convert an original input sentence, which includes clauses and their relationships, into a discourse tree that explicitly denotes those relationships.

We encode the discourse tree by doing a pre-order traversal of the tree. Algorithm 1 discusses our steps.

Algorithm 1 Generating encoding $\mathcal{E}$ for a Discourse Tree T.

Input: Discourse Tree $\mathcal{T}$ with root *root*
Output: Encoding, $\mathcal{E}$
Append '$root.label($' to $\mathcal{E}$
foreach *child of root in* $\mathcal{T}$ **do**
  if *child is a leaf* **then**
    Append '$child.label,$' to $\mathcal{E}$
  end
  else
    Generate encoding $\mathcal{E}'$ of Discourse Sub-Tree with *child* as root
    Append $\mathcal{E}'$ to $\mathcal{E}$
  end
end
Append ')' to $\mathcal{E}$
return $\mathcal{E}$

5 Experiments

5.1 Datasets

5.1.1 Training

We trained *LeGen* using 17k sentences from Penn Tree Bank [29] dataset. We perform our experiments on 32x2 cores AMD EPYC 7532, 1 TB of memory, and 8x A100 SXM4 80GB GPU systems. We train the models using BART-base (139 M), BART-small (70.5 M), T5-base (246 M), and T5-small (77M) architectures. BART trained faster (2 hours on small and 2.5 hours on base). T5 took considerably longer time (3 hours for small and 4 hours for base). We train it separately for 2 tasks:

Task 1: Identifying Sub-ordinations and Co-ordinations. We encoded every sentence into a discourse tree structure as described in Section 4. We trained BART [25] and T5 [1] models for 30 epochs using cross-entropy loss with a learning rate of $e^{-}5$. We trained our models on 3 seeds and report averaged results.

Task 2: Identifying Co-ordinations. In order to test *LeGen*'s flexibility, we also separately trained it as a coordinate boundary detection task [36]. The purpose of this study was to test the competency of BART and T5 models in splitting sentences over state-of-the-art non generative techniques like OpenIE6. We converted the OpenIE6 coordinate boundary labels into a discourse tree and generated its encoding. The non-leaf nodes in this tree represented only the coordination relation. We kept the same hyperparameters that we used for the subordination task and obtained the best results for batch size 3. We trained our model on 3 seeds and report averaged results (Section 6).

5.1.2 Test

1) Indian Legal Dataset. There are Indian Legal datasets such as the ILDC [27] for legal named entity recognition, rhetorical role identification, and court judgment prediction tasks from court transcripts. There are non-Indian legal datasets such as ECtHR [9] or Pile of Law [18] used to build pre-trained language models for law. However, we are unaware of any datasets that annotate individual legal sentences for information extraction. We closed this research gap by creating an Indian Legal Benchmark for information extraction by including 107 sentences from Wiki [8] on Labour Law [9].

2) Penn Tree Bank. Penn Tree Bank [29] consists of sentences from articles in the Wall Street Journal. It is annotated with coordinate boundaries (‘and’, ‘or’, ‘but’, comma-separated list) and the text spans it connects. This test set containing 985 sentences was used to evaluate $LeGen$’s flexibility in identifying co-ordinations.

5.2 Metrics

5.2.1 Metrics for Task 1

While discourse trees have been used to improve downstream tasks such as text classification [14] or open information extraction [32], we are unaware of any metric used to evaluate them directly. So, we evaluate the trees based on: i) structure of the tree and ii) content of the tree, i.e. the relation labels. For both, we performed a human evaluation since there can be more than one correct tree for a sentence.

Tree Structure Evaluation (TSE). We employed a strict evaluation technique, i.e. it was marked as correct only if all the 3 requirements cited in Section 3 were satisfied.

- Every node in the tree was correctly split. For instance, a tree that splits sentence on a nondistributive coordination like ‘between" – "*The term ’industry’ infuses a contractual relationship between the employer and the employee*" into "*The term ‘ industry ’ infuses a contractual relationship between the employer*" and "*The term ‘ industry ’ infuses a contractual relationship between the employee*" will be marked as incorrect.

- Tree does not contain multiple nodes with the same information

- All information in the sentence was conveyed in the tree.

TSE reports the percentage of sentences that generated correct trees.

Tree Content Evaluation (TCE). To evaluate the content of the tree, we asked the annotators to mark each relation in the tree as correct/incorrect. The annotators were briefed about the different relations in the test set. A relation was marked wrong if it could have been expressed using some other relation or if it connected incorrect clauses.

5.2.2 Metrics for Task 2

We employed a **mapping-based approach** proposed in CalmIE [36] to compare the clauses generated by our technique with the gold set. For every conjunctive sentence, we evaluate it by matching its collection of system-generated clauses with the reference set. This involves establishing the most optimal one-to-one correspondence between the clauses in both sets. Subsequently, precision is determined for each mapping by calculating the ratio of shared words to the total words in the generated sentence, while recall is calculated as the ratio of shared words to the total words in the reference sentence.

Let $G = \{G_1, G_2, G_3 \ldots\}$ be gold/reference clauses each represented as a bag of words model, i.e. $G_i = \{G_i^{a1}, G_i^{a2}, G_i^{a3} \ldots\}$ where each G_i^{aj} denotes a token in a clause. Similarly let $T = \{T_1, T_2, T_3 \ldots\}$ be clauses generated by a model where $T_i = \{T_i^{a1}, T_i^{a2}, T_i^{a3} \ldots\}$. CalmIE performs matching in a greedy fashion, however, this type of matching is not optimal and might change based on the order in which greedy matching is performed. So, we perform matching to get the global maximum. This problem of finding the global optimum from a distance or similarity matrix can be treated as a linear sum assignment problem [10]. We match clauses from Gold Set G and Predicted

[8] https://en.wikipedia.org/wiki/Main_Page
[9] https://en.wikipedia.org/wiki/Indian_labour_law

Set T to maximize the F1 score. The F1 score will be computed using precision and recall metrics.

$$p = precision(G_i, T_j) = \frac{|G_i \cap T_i|}{|T_i|} \tag{1}$$

$$r = recall(G_i, T_j) = \frac{|G_i \cap T_i|}{|G_i|} \tag{2}$$

$$f1(G_i, T_j) = \frac{2pr}{p + r} \tag{3}$$

Let $m(.)$ be matching function such that G_i matches with $T_{m(i)}$ and conversely $G_{m(j)}$ matches with T_j. If $|G| \neq |T|$, then only $k = min(|G|, |T|)$ matches are possible. Thus in such cases, $m(i)$ will not return valid value for all i and $precision(G_i, T_{m(i)})$ and $recall(G_i, T_{m(i)})$ will be zero.

$$\begin{aligned} p_{example} &= precision(G, T) \\ &= \frac{1}{|T|} \sum_{i=1}^{|T|} precision(G_{m(i)}, T_i) \end{aligned} \tag{4}$$

$$\begin{aligned} r_{example} &= recall(G, T) \\ &= \frac{1}{|G|} \sum_{i=1}^{|G|} precision(G_i, T_{m(i)}) \end{aligned} \tag{5}$$

$$f1_{example} = (G, T) = \frac{2 p_{example} r_{example}}{p_{example} + r_{example}} \tag{6}$$

Please note that (4) to (6) represent scores for only one example in the test set.

5.3 Baselines

Graphene. We used Graphene [32] as the competing technique for Task 1.

OpenIE6. We used the Coordination Boundary Detection Model released with OpenIE6 as our baseline for Task 2.

6 Results

6.1 Task 1

We asked 2 annotators (authors of the paper) to evaluate the trees. Each tree was evaluated by 1 annotator according to the metrics described in Section 5.2.1.

Inter-annotator Agreement. We sampled 50% of the sentences annotated by Annotator 1 and asked Annotator 2 to evaluate them. We obtained a Cohen's Kappa agreement value of 86.3, indicating near-perfect agreement [3].

Table 2 shows the **TSE**, **TCE**, and the number of clauses and relations generated in the discourse trees by each of these 3 techniques. It is clear that

	TSE	TCE	#Relations and Clauses (c, r)
Graphene	0. 6168	<u>0.9242</u>	(247, 377)
T5 - BASE	**0.7076**	**0.9618**	(191, 349)
BART BASE	<u>0.6977</u>	0.9210	(183, 281)

Table 2: TSE and TCE results of Graphene, T5, and BART, averaged over 3 seeds. The best values are in bold. Second best are undelined.

Input	Clauses generated by Graphene	Clauses generated by T5 BASE
The Factories Act 1948 and the Shops and Establishment Act 1960 mandate 15 working days of fully paid vacation leave each year to each employee with an additional 7 fully paid sick days.	1) This was with an additional 7 fully paid 2) This was to each employee 3) The Factories leave each year sick days 4) Act 1948 mandate 15 working days of fully paid vacation The Factories 5) The Shops and Establishment Act 1960 mandate 15 working days of fully paid vacation The Factories	1) This was to each employee with an additional 7 fully paid sick days 2) The Factories Act 1948 mandate 15 working days of fully paid vacation leave each year 3) The Shops and Establishment Act 1960 mandate 15 working days of fully paid vacation leave each year.

Table 3: Examples showing the superiority of generative architectures in identifying correct clauses. Their strength also lies in accurate detection of named entities

the generative approach for discourse tree creation outperforms Graphene. T5-Base performs the best and beats Graphene by 9 pts with a TSE score of 70%. BART-Base hallucinates more and the reason for its underperformance is the generation of terms not present in the original sentence. Graphene underperforms on sentences where domain-specific named entities such as statutes, laws, or case names are present, e.g. *Shops and Establishment Act 1960* or *The Factories Act 1948* (Table 3). Graphene also cannot identify nondistributive coordination like 'between' and splits sentences on them. All these issues are handled very well by generative models even though they were trained on Graphene's output.

While evaluating for **TCE**, we took into consideration the fact that there could be multiple ways of representing sentences with different relations. There are situations, where models are able to split the sentences but unable to identify the relations and BART has made spelling mistakes in identifying the relation. Although such scenarios were rare in T5, we came across them in Graphene and BART.

Model		OpenIE	T5-small	T5-base	BART-small	BART-base
Mapping based Approach	**Precision**	**0.9803**	0.9647	0.9747	0.8215	0.8369
	Recall	**0.9845**	0.9544	0.9730	0.7391	0.7574
	F1-score	**0.9816**	0.9571	0.9726	0.7682	0.7859

Table 4: Mapping based Scores for OpenIE6, T5, and BART averaged over 3 seeds. The best values are in bold. The second best is underlined.

Level	Mapping Based Approach	OpenIE	T5-base	T5-small	BART-base	BART-small	Count
Level 0	Precision	**0.9796**	0.9632	0.9182	0.9755	0.9714	163
	Recall	**0.9816**	0.9632	0.9182	0.9755	0.9714	
	F1 Score	**0.9816**	0.9632	0.9182	0.9755	0.9714	
Level 1	Precision	**0.9856**	0.9800	0.9789	0.8240	0.8126	716
	Recall	**0.9866**	0.9773	0.9669	0.7418	0.7287	
	F1 Score	**0.9856**	0.9781	0.9717	0.7720	0.7580	
Level 2	Precision	0.9465	**0.9518**	0.9428	0.7287	0.6789	98
	Recall	**0.9737**	0.9685	0.9348	0.5790	0.4900	
	F1 Score	0.9564	**0.9567**	0.9365	0.6321	0.5611	
Level 3	Precision	0.9354	**0.9607**	0.9144	0.5454	0.6330	6
	Recall	**0.9914**	0.8823	0.8178	0.3574	0.3227	
	F1 Score	**0.9606**	0.9168	0.8536	0.4252	0.4155	
Level 4	Precision	0.7975	**0.9100**	0.8848	0.7666	0.6772	2
	Recall	**1.0000**	0.8950	0.8183	0.3480	0.3216	
	F1 Score	0.8814	**0.9008**	0.8416	0.4432	0.4334	

Table 5: Level-wise scores aggregated across 3 seeds. The best values are in bold. The second best is underlined.

6.2 Task 2

Table 4 shows our results. We obtained competent results from the T5-base against OpenIE6. The slight drop in the performance of T5-Base could be attributed to ambiguous labels in the Penn Tree Bank dataset. For instance, one split in the gold for "*He retired as senior vice president, finance and administration, and chief financial officer of the company Oct. 1*" is "*He retired as senior vice president, finance Oct. 1*", while T5 generates "*He retired as senior vice president, finance, of the company Oct. 1*". T5 generates a better split but it gets penalized because this is not captured in gold.

BART did not perform well as it hallucinated while generating the output where it used words that are not in the input. BART was also unable to split all elements of comma-separated lists. The same problem was observed for T5-small which improved with T5-base.

We also evaluated the performance of our model against sentences with different levels of complexity. Conjunctive sentences are likely to have multiple conjunctions and thus produce complicated coordination tree structures with greater height. We evaluated models for sentences with different coordination tree heights in the gold set (Table 5). In the test and train set, at level 0, we have 163 and 2426 sentences, level 1 has 716 and 12958, level 2 has 98 and 1716, level 3 has 6 and 153, level 4 has 2 and 26 and level 5 has 0 and 1 sentences. Level 0 indicates that a sentence cannot be split into simpler sentences. The model will generate `NONE` as output for these sentences. We see a similar trend with OpenIE6 slightly outperforming the generative approach. One reason for this is the presence of ambiguous labels in the test set for hierarchies with multiple levels. On such sentences, even though T5 generates a better split, it is still penalized. BART does well on identifying sentences that should not be split, however, it hallucinates when sentences become more complex.

7 Conclusion

We proposed an end-to-end generative legal information extraction technique that can improve the understanding of long and complex legal sentences. We model this as complex information extraction. We achieved this by learning the discourse tree of the sentence using generative models like T5 and BART. We outperformed Graphene, a state-of-the-art complex information extraction technique on an Indian Legal Benchmark, and achieved competitive results on the task of the coordinate boundary detection technique. We plan to extend the generative-based complex information extraction for rhetorical role prediction and extend support for Indian languages.

8 Limitations

- Generative models are prone to hallucinations.
- Systems running these models should have the computational capacity to process large models like T5 or BART.

References

[1] Abdelrahman Abdallah, Bhawna Piryani, and Adam Jatowt. 2023. Exploring the state of the art in legal qa systems. *arXiv preprint arXiv:2304.06623*.

[2] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ computer science*, 2:e93.

[3] Nicole J-M Blackman and John J Koval. 2000. Interval estimation for cohen's kappa as a measure of agreement. *Statistics in medicine*, 19(5):723–741.

[4] Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, and Serena Villata. 2017. Legal nerc with ontologies, wikipedia and curriculum learning. In *15th European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 254–259.

[5] Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018. Graphene: Semantically-linked propositions in open information extraction. *arXiv preprint arXiv:1807.11276*.

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

[7] Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2017. Extracting contract elements. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 19–28.

[8] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

[9] 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. *arXiv preprint arXiv:2103.13084*.

[10] David F Crouse. 2016. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

[11] Robert Dale. 2019. Law and word order: Nlp in legal tech. *Natural Language Engineering*, 25(1):211–217.

[12] Luciano Del Corro and Rainer Gemulla. 2013. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366.

[13] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, et al. 2011. Open information extraction: The second generation. In *Twenty-Second International Joint Conference on Artificial Intelligence*. Citeseer.

[14] Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2019. Evaluating discourse in structured text representations. *CoRR*, abs/1906.01472.

[15] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. Minie: minimizing facts in open information extraction. Association for Computational Linguistics.

[16] Google. 2023. Next billion users. `https://blog.google/technology/next-billion-users/`.

[17] Meghna Gupta, Devansh Mehta, Anandita Punj, and Indrani Medhi Thies. 2022. Sophistication with limitation: Understanding smartphone usage by emergent users in india. In *ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS)*, pages 386–400.

[18] Peter Henderson, Mark Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel Ho. 2022. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Advances in Neural Information Processing Systems*, 35:29217–29234.

[19] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *arXiv preprint arXiv:2103.06268*.

[20] Anirudha Joshi. 2013. Technology adoption by'emergent'users: the user-usage model. In *Proceedings of the 11th Asia Pacific conference on computer human interaction*, pages 28–38.

[21] Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named entity recognition in indian court judgments.

[22] Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for automatic structuring of legal documents. *arXiv preprint arXiv:2201.13125*.

[23] Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Soumen Chakrabarti, et al. 2020. Openie6: Iterative grid labeling and coordination analysis for open information extraction. *arXiv preprint arXiv:2010.03147*.

[24] Spyretta Leivaditi, Julien Rossi, and Evangelos Kanoulas. 2020. A benchmark for lease contract review. *arXiv preprint arXiv:2010.10386*.

[25] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

[26] Mechket Emna Mahouachi and Fabian M Suchanek. 2020. Extracting complex information from natural language text: A survey. In *CIKM (Workshops)*.

[27] 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*.

[28] William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

[29] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

[30] Masha Medvedeva, Michel Vols, and Martijn Wieling. 2020. Using machine learning to predict decisions of the european court of human rights. *Artificial Intelligence and Law*, 28:237–266.

[31] Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Kumar Guha, Sachin Malhan, and Vivek Raghavan. 2023. Semeval 2023 task 6: Legaleval - understanding legal texts.

[32] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019. Transforming complex sentences into a semantic hierarchy. *arXiv preprint arXiv:1906.01038*.

[33] Radityo Eko Prasojo, Mouna Kacimi, and Werner Nutt. 2018. Stuffie: Semantic tagging of unlabeled facets using fine-grained information extraction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 467–476.

[34] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

[36] Swarnadeep Saha et al. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299.

[37] Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 523–534.

[38] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895.

[39] Shangliao Sun. 2023. Smartphone penetration rate in india from 2009 to 2023, with estimates until 2040. https://www.statista.com/statistics/1229799/india-smartphone-penetration-rate/#:~:text=In%202023%2C%20the%20penetration%20rate,end%20of%202020%20was%20Xiaomi.

[40] Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. Leven: A large-scale chinese legal event detection dataset. *arXiv preprint arXiv:2203.08556*.

[41] Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26.

[42] Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.

[43] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does nlp benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.

9 Appendix

9.1 Graphene Relations used for *LeGen* training

1. **SPATIAL** : This relation is used to denote the place of occurance of an event .

   Eg: The Inter-state Migrant Workmen Act 's purpose was to protect workers whose services are requisitioned outside their native states in India .

SUB/ELABORATION('The Inter-state Migrant Workmen Act 's purpose was to protect workers .', **SUB/SPATIAL**('This is in India .','Workers 's services are requisitioned outside their native states .'))

2. **ATTRIBUTION**: This relation is used when a statement is being made by some person or institution.

   Eg: But some militant SCI TV junk-holders say that 's not enough .

   SUB/ATTRIBUTION('This is what some militant SCI TV junk-holders say .',"s not enough .')

3. **CONTRAST**: This relation is indicated by the words "although" , "but" , "but now", "despite" , "even though" , "even when", "except when" , "however", "instead" , "rather", "still" , "though" , "thus", "until recently", "while" and "yet".

   Eg: This can have its purposes at times , but there 's no reason to cloud the importance and allure of Western concepts of freedom and justice .

   CO/CONTRAST(SUB/ELABORATION('This is at times .','This can have its purposes .'), 'There 's no reason to cloud the importance and allure of Western concepts of freedom and justice .')

   Eg2: No one has worked out the players ' average age , but most appear to be in their late 30s .
   CO/CONTRAST('No one has worked out the players ' average age .',' most appear to be in their late 30s . ')

4. **LIST** : This is used to indicate conjunctions ('and' or comma seperated words) between the sentences

   Eg: He believes in what he plays , **and** he plays superbly .
   CO/LIST('He believes in what he plays .','He plays superbly .')

5. **DISJUNCTION**: This is used to show the presence of 'OR' in the sentences.

   Eg: The carpet division had 1988 sales of $ 368.3 million , or almost 14 % of Armstrong 's $ 2.68 billion total revenue .

   CO/DISJUNCTION('The carpet division had 1988 sales of $ 368.3 million .','The carpet division had 1988 sales of almost 14 % of Armstrong 's $ 2.68 billion total revenue .')

6. **CAUSE**: Indicates the presence of the word - 'because' or 'since'.

   Eg: Jaguar 's own defenses against a hostile bid are weakened , analysts add , because fewer than 3 % of its shares are owned by employees and management .

   SUB/CAUSE('Jaguar 's own defenses against a hostile bid are weakened , analysts add .','Fewer than 3 % of its shares are owned by employees and management .')

7. **CONDITION**: When multiple sentences are connected by phrase 'if' 'in case','unless' and 'until', CONDITION relationship phrase is used to denote the connection between the sentences.

   Eg: Unless he closes the gap , Republicans risk losing not only the governorship but also the assembly next month .

   SUB/CONDITION('He closes the gap .','Republicans risk losing not only the governorship but also the assembly next month .')

8. **ELABORATION**: Identified by the presence of words such as "more provocatively","even before" ," for example","recently" ," so" ,"so far" ," where" ,"whereby" and "whether" .

   REGEX:

   ```
   ``since(\\W(.*?\\W)?)now"
   ```

   Eg: Not one thing in the house is **where** it is supposed to be , but the structure is fine .

   CO/CONTRAST(SUB/ELABORATION('Not one thing in the house is .','It is supposed to be .'), 'The structure is fine .')

9. **TEMPORAL** : Denotes the time or date of occurrence of the event.

   Eg: These days he hustles to house-painting jobs in his Chevy pickup before and after training with the Tropics .

   ```
   SUB/TEMPORAL('These days he hustles to house-painting jobs in his Chevy pickup before and after .','These days he is training with the Tropics .')
   ```

10. **PURPOSE**: This kind of relation is identified by the presence f words such as “for" or “to".

    Eg: But we can think of many reasons to stay out for the foreseeable future and well beyond .

    ```
    SUB/PURPOSE('But we can think of many reasons .','This is to stay out for the foreseeable future and well beyond .')
    ```