LegalViz: Legal Text Visualization by Text To Diagram Generation

Anonymous

Abstract

Legal documents including judgments, court orders, government ordinances, professional papers, and textbooks of judicial examinations require highly sophisticated legal knowledge for understanding. To disclose expert knowledge for non-experts, we explore the problem of visualizing legal texts with easy-to-understand diagrams and propose a novel dataset of LegalViz with 23 languages and 5,580 cases of legal document and visualization pairs, using the DOT graph description language of Graphviz. LegalViz provides a simple diagram from a complicated legal corpus identifying legal entities, rules, statements, and transactions at a glance, that are important in each judgment. In addition, we provide a new evaluation approach for the legal diagram visualization by considering the graph and text similarities. We conducted empirical studies on few-shot and finetuning large language models for generating legal diagrams and evaluated them with the graph and text evaluation metrics by each model in 23 languages and confirmed the effectiveness of our dataset.

1 Introduction

Natural Language Processing (NLP) of the legal domain receives increasing attention (Niklaus et al., 2023) as the steep development of Large Language Model (Brown et al., 2020; OpenAI, 2023) (LLM) and their highly scored achievements of traditional NLP tasks. At an early stage of legal NLP, there are several research applying traditional NLP tasks on legal documents, such as Named Entity Recognition (Angelidis et al., 2018; Luz de Araujo et al., 2018; Pais et al., 2021; de Gibert Bonet et al., 2022), summarization (Elaraby and Litman, 2022; Aumiller et al., 2022), classification (Chalkidis et al., 2019) and text segmentation (Aumiller et al., 2021). These studies, however, often process the surface of legal articles, lacking in-depth analyses of the legal interpretation of the documents.

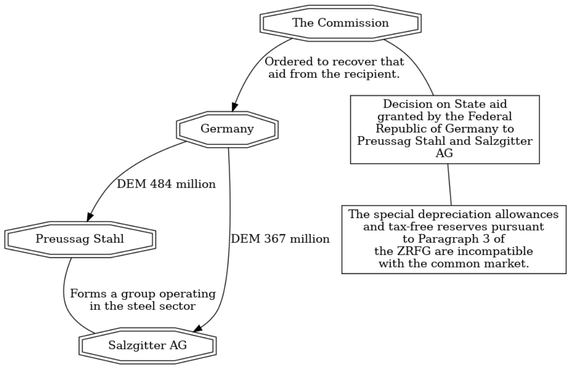

Figure 1: Annotated legal text visualization drawn by Graphviz.

Legal documents are often written in a strict format and include specific terminologies as discussed by Zhong et al. (2020); Chalkidis et al. (2020, 2022a). Legal experts often interpret articles considering not only the surface wording of the legal documents but also the objective and purpose of such articles, the legal interest of that law system, or even the legal custom of the rules. Therefore it is not sufficient to only consider the surface wording of the texts. Some notable studies are focusing on capturing those structural legal meanings, such as learning judgment facts and results (Niklaus et al., 2021), the fairness of law (Chalkidis et al., 2022b), and using the facts and attributes to predict charges (Hu et al., 2018). To study an in-depth analysis of legal interpretation, we conducted annotations to capture the requirements to interpret legal norms for experts such as legal statements applicants and defendant made, legal rules they rely on, legal entities, and transactions they related to, which experts use for final judicial conclusion.

On the other hand, for business companies investing in other countries and entrepreneurs starting new businesses in unfamiliar fields, there are numerous demands that non-legal experts also desire to grasp the meaning of the legal rules and court decisions that are related to their businesses, properties, and employment. To meet these demands, visualization of legal concepts is used in, for example, textbooks of judicial examination, university classrooms, or TV news to offer easy-to-interpret visual and conceptual understandings of legal materials for non-experts. Figure 1 is an example of such a legal diagram explaining the case in which the Commission ordered Germany to recover the aid in the principle of the common market and Germany made recovery requests. This figure can explain complex legal relations at a glance without reading the original article.

In this study, we explore an automatic visualization model with LLM providing legal diagrams, which recognizes legal rules concerned in the case, legal entities capable of exercising rights, legal transactions, and statements, from professional legal documents. To achieve this goal, we introduced a novel dataset, LegalViz, including 5,508 diagrams of DOT language code used in Graphviz and professional legal document pairs. Legal documents are collected from open source EU legislation materials of EUR-LEX, to let models comprehend legal systems in 23 different languages of EU countries to utilize in both professional and industrial domains. To the best of our knowledge, this is the first work to visualize legal documents with the help of the large language model.[1]

Our contributions to this study are as follows:

1. We introduce a novel dataset of LegalViz, which establishes a new task of generating diagram visualizations from legal documents, covering 23 languages from EUR-LEX.

2. We proposed an evaluation method to assess scores of the legal visualization, taking into account both diagram visualization quality and sentences of graph nodes and relations.

3. We conducted extensive empirical studies on LegalViz and observed the effectiveness of our dataset both quantitatively and qualitatively.

2 Related Work

We can categorize the applications of natural language processing in the legal domain into several core areas (Katz et al., 2023); namely, information extraction, classification, summarization, judgment prediction, and resources and benchmarks.

[1]Our dataset is available at ANONYMIZEDURL

Legal information extraction. Information extraction (IE) in the legal domain can be crucial for other higher-level tasks like classification or summarization. Named Entity Recognition (NER) is a fundamental information extraction task that has been developed for several languages, including Greek (Angelidis et al., 2018), Brazilian (Luz de Araujo et al., 2018), Romanian (Pais et al., 2021), and Spanish (de Gibert Bonet et al., 2022). Those NER approaches extract mainly the same objects as those in non-legal domains. Some efforts try to extract legal entities from court documents (II et al., 2021). Once NER identified entities, Relation Extraction in the legal domain (Chalkidis et al., 2021b) takes this information further by identifying and classifying the relationships between these entities, such as facts and allegedly violated articles, specific articles and paragraphs, and case references, as well as relevant facts and allegations.

Legal classification. The classification task of legal texts has been proposed with a focus on practical applications. For example, to enhance the interpretation of complex legal information, multi-label classification of legal texts assigns multiple conceptual class labels to words appearing in legal sentences (Chalkidis et al., 2019). Other applications include multi-labeled provision classification (Tuggener et al., 2020) or legal document classification (Chalkidis et al., 2021a), classifications in Greek legal domain (Papaloukas et al., 2021). Notably, FairLex (Chalkidis et al., 2022b) aims to ensure the fair application of the law by classifying attributes such as age, gender, region, and state.

Legal summarization. As a more complex and application-oriented task, legal summarization is also prominent in the field, which aims to generate a summary of legal sentences. Existing summarization studies address Canadian legal cases (Elaraby and Litman, 2022), EU legislations (Aumiller et al., 2022).

Judgment prediction. Judgment prediction is the task of predicting the outcomes of legal cases based on the given facts. Previous studies provide judgment data from various courts, including decisions from the Supreme Court of the United States (Katz et al., 2017) and the European Court of Human Rights (Medvedeva et al., 2020; Kaur and Bozic, 2019). Additionally, judgment prediction research has covered Switzerland (Niklaus et al., 2021), Chinna (Ye et al., 2018), including criminal law (Chen et al., 2019; Xiao et al., 2018), and asylum decisions (Chen and Eagel, 2017; Dunn et al.,

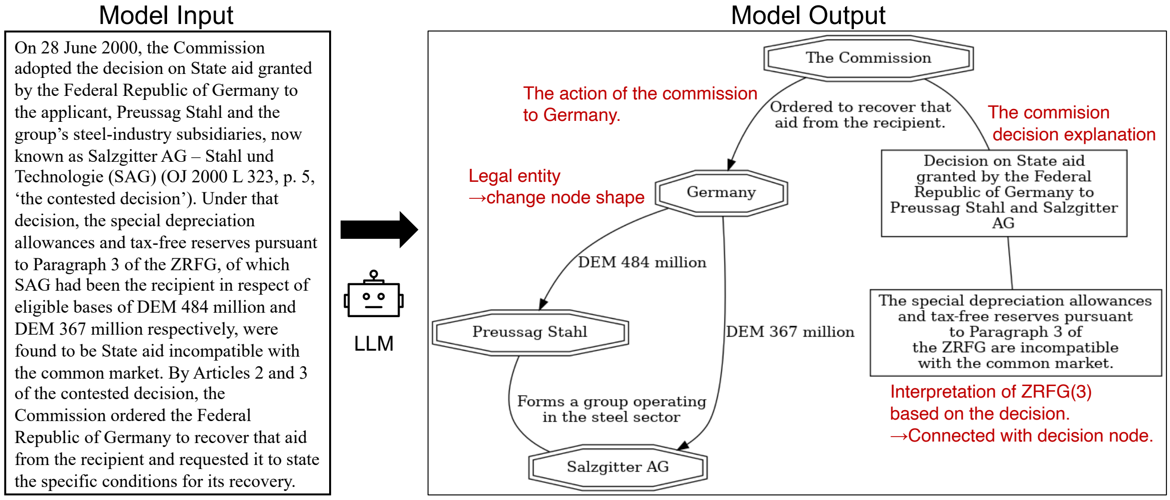

Figure 2: Legal text from EUR-LEX (left) to the resulting legal graph (right). Red texts present the auxiliary requirements for Graphviz visualization.

2017).

Legal resources and benchmarks. A range of datasets and their benchmarks have been proposed for legal NLP tasks, including English Tax Law (Holzenberger et al., 2020), European Legislation and the European Court of Human Rights (Chalkidis et al., 2019), Corporate and Contract Law (Hendrycks et al., 2021; Tuggener et al., 2020), Supreme Court cases and US court cases (Zheng et al., 2021), Germany legal cases (Urchs. et al., 2021), a mixture of Korean legal text summarization, prediction and classification (Hwang et al., 2024), refugee cases (Barale et al., 2023). There are also multilingual and multi-legal domain cases such as a multilingual corpus of English, German, Italian, Polish (Drawzeski et al., 2021), LEXGLUE (Chalkidis et al., 2022a) covering six predictive tasks over five datasets made of English from the US, EU, and Council of Europe, Lexfiles (Chalkidis et al., 2023), a comprehensive dataset of comprised of US, UK, Canada, India, European Court of Human Rights, and Lextreme (Niklaus et al., 2023) covers wide-range of tasks and countries among EU nations.

Text to graph generation. Following the iconic successions of the GPT models, it has become known that GPT models can generate not only contextual texts and program codes but also visualization codes (Bubeck et al., 2023). It is also soon known that LLMs, not limited to GPTs, can also generate the graph languages, and the datasets and methods for visualization code generations have been created, such as the TiKZ dataset (Belouadi et al., 2024) and diagram generation with refinements and diffusion process (Zala et al., 2023).

Our work proposes a novel application of text-to-graph generation in the legal domain, aimed at providing non-legal experts with a simple and clear understanding of professional legal text at a glance. Additionally, we introduced more detailed legal annotations than existing research, offering in-depth insights into the recognition of legal entities, their rights, the rules supporting legal statements, transactions between legal entities, and summaries of facts necessary for judicial judgments.

3 Dataset

3.1 Task Definition

We introduce a novel task to automatically visualize legal text with the DOT language of Graphviz. The task input is a legal text that composes both legal entities and/or rules that can form graph nodes and legal transactions and/or important facts valuable to note for judicial determination that can form graph relations. The task purpose is to produce a diagram that is coded in the DOT language to illustrate legal relationships among input texts. Figure 2 illustrates the overview of our proposed task input and output that comprises the following six aspects.

Legal entity extraction. To draw a graph from legal judgments, we first extract legal entities such as applicants and respondents of judgment, courts, creditors, debtors, criminal suspects, or companies and employees. Extracted entities are drawn as specific shapes (octagons). In contrast to extracting grammatical general nouns, proper nouns, or objects, we aim to extract persons or organizations capable of exercising legal rights and engaging in transactions.

Legal relationship extraction. Legal relationships encompass various elements, including the exercise of legal rights from one to another, legally significant transactions, the interrelations between legal statements made by entities and the underlying norms that support them, and relationships defined under law such as employment, contractual agreements, marriage, and family relationships. Extracted relationships are represented as the edge of a diagram with various lines. For graph construction, we detect and categorize the aforementioned legal relations between legal entities and predict their relation labels.

Legal source extraction. For a "legal source" extraction, we extract the rules applied or referred to in the judgments from the input text. This includes constitutions, statutes, ordinances, and case law. To draw the legal relationship diagram, these extracted rules are drawn in a specific shape (trapezium) and connected to the nodes applying the rules.

Legal statement extraction and summarization. To make legal texts more compact and understandable, we extract legal statements, detailed explanations of transactions, and factual descriptions of the case notable for the final judgment to summarize. Adding these summaries to diagrams makes non-experts grasp the facts important for final judgments at a glance.

Legal transaction extraction. We extract legal transactions between each entity such as purchases, notifications, and any actions exercising rights. By drawing these transactions in diagrams, we can identify the important actions for legal results and determine which entity performed those actions.

Structural legal understanding and explanation. By connecting the extracted elements above into one diagram, we can obtain the same legal interpretation view as the courts making judicial decisions. Legal professionals identify the rules applicable to each case and which legal actions are made by what character of legal entities are noteworthy for judicial interpretations. Therefore, we conducted annotations on identifying rules, legal entities, and transactions as well, that are used for judicial interpretation to introduce legal conclusions.

3.2 Legal Diagram Formalism

Here we define several rules to express legal relations within the DOT language grammar.

Graph node rules. Legal entities are represented by nodes (vertices) in DOT languages with the shape of double octagons except legally deceased persons who are presented in the shape of ellipses. Legal norms that are effective in the present case are represented by graph nodes with trapezium shapes.

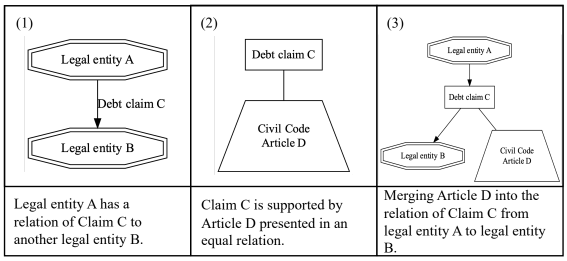

Figure 3: Annotation rule when adding explanation to graph relations.

Graph edge rules. Legal transactions and the explanatory relationships between legal entities are represented by directed edges. The family or marital relationships established under civil law are represented by an undirected bold edge. The legal rights that cannot be exercised are represented by dashed edges. Dotted edges denote relationships of the legal succession between legal entities. To illustrate the equivalent relationship between diagram nodes, undirected edges are used to connect entities and their status explanations, rules and statements, legal transactions, and their explanations.

We also note that legal relations can also be represented by graph nodes when legal relations have some relations with other entities. Figure 3 explains how to draw graphs when additional description is required for graph relations. In Graphviz, we cannot draw lines directly to the graph relations. Hence we change graph labels to nodes and connect to other nodes for adding explanation. Further details of the DOT language grammar for representations of legal entity relations and an actual dataset example are provided in Appendix B & D.

3.3 Dataset Creation

Collection of legal document. To construct the legal graph dataset, we collected legal documents as follows. (i) We collected legal documents from the EUR-LEX website[2], which offers public access to judgments, orders, opinions, and rules of EU countries over 22 languages. These judgments from 2006 to 2019, available in translations across 23 languages, were primarily sourced to capture

[2] https://eur-lex.europa.eu

the latest legal trends. (ii) We then extracted the factual sections of the judgments that contain legal facts to be expressed in the graph. (iii) Finally, we obtained the corresponding sections of legal documents in the remaining 22 languages to ensure consistency across translations.

Graphviz annotation. We have manually annotated Graphviz code visualization from the legal documents by an annotator with expertise in the legal domain. (i) We broke down long judgment cases into short paragraphs so that DOT language can draw diagrams in units that are easily understandable at a glance. (ii) We extracted the legal entities and rules as nodes of the diagram, legal transactions as relations within the diagram, and the summary of statements and explanations as normal nodes. (iii) We have created a Graphviz diagram to represent the extracted relations, using variations in node shape and relations, following the rules of node shape and relation variations given in Section 3.2.

Translation of Graphviz annotation. To cover the European Union's official languages present at the time the judgment was written, we translated our English annotation to other languages as follows. (i) We first used GPT-4 to extract the legal words and sentences from the provided English sentences, aiming to save as many terms as possible from the EU's officially translated variations of judgments. (ii) We then apply the translation of GPT-4 to such sentences if the extraction task fails. (iii) We manually checked the previously translated sentences and retranslated them using DeepL and the Azure GPT API if any translation errors were found. The prompts used in the translation process are described in Appendix C.

3.4 Dataset statistics

We build a total of 5,580 pairs of legal texts and graphs, encompassing 23 language variations and 250 unique legal texts. The constructed legal graph consists of 15,497 nodes and 60,890 relations. Table 1 shows dataset statistics by each data split. We also summarize the average word length, number of characters in legal sentences, and character length of Graphviz code for each language in Table 2.

Split	# Instances	# Nodes	# Relations
Train	3,280	8,965	37,687
Validation	1,150	3,404	11,213
Test	1,150	3,128	11,990
Total	5,580	15,497	60,890

Table 1: Dataset splits.

Lang.	ISO	L_{word}	L_{char}	L_{code}
All	-	113.9	675.4	642.4
Bulgarian	BG	119.6	662.9	648.9
Spanish	ES	139.7	720,2	648.2
Czech	CS	106.1	606.0	633.4
Danish	DA	115.1	669.1	644.5
German	DE	114.1	718.4	630.6
Estonian	ET	87.2	613.9	635.8
Greek	EL	126.8	732.7	649.0
English	EN	129.3	662.9	633.5
French	FR	135.1	708.7	640.2
Croatian	HR	107.1	603.6	646.0
Italian	IT	129.5	741.3	641.0
Latvian	LV	97.7	623.7	637.4
Lithuanian	LT	98.5	640.6	640.1
Hungarian	HU	100.8	700.3	645.4
Maltese	MT	104.6	741.6	651.9
Dutch	NL	128.7	720.7	641.0
Polish	PL	112.0	691.0	647.3
Portuguese	PT	131.5	685.4	646.9
Romanian	RO	124.7	710.2	654.2
Slovak	SK	104.6	608.1	633.6
Slovene	SL	109.9	601.7	635.7
Finnish	FI	81.2	681.2	649.6
Swedish	SV	114.6	674.5	643.4

Table 2: Dataset statistics. L_{word} and L_{char} are length of legal text. L_{code} is character length of Graphviz code.

4 Evaluation

Our goal is to visualize legal entities' relationships to promote understanding of complex legal documents. We compare the two Graphviz codes. One approach directly compares two graph codes using textual metrics such as the BLEU score, while the other is a completely image-based approach where we compare two visualized graphs using image-based metrics. The former approach ignores the fact that numerous different visualization codes can represent identical graphs and cannot evaluate whether the predicted code is meaningful in the context of the DOT language. The latter approach ignores the details of textual structures.

4.1 Similarity of two graphs with texts

To compare the matching of both the graph and textual representations of two graphs, ground-truth and predicted, we simultaneously calculate the graph-based similarity and the textual similarity of the nodes for evaluation. Formally, let $\mathcal{G}_r$ and $\mathcal{G}_h$ be the reference and hypothesis graphs. Each graph is composed of a set of edges E and nodes $\mathbf{v}$. An edge $e \in E$ that connects a starting node v_s to an

Model	Validation G	Validation G-N	Validation G-N-E	Test G	Test G-N	Test G-N-E
Few-shot						
Llama3 8B	20.61	1.42	0.90	19.17	1.88	1.04
Llama3 8B Inst.	21.69	1.81	1.22	19.15	1.62	0.83
CodeLlama 7B	10.68	0.29	0.15	10.79	0.33	0.09
CodeLlama 7B Inst.	15.46	0.57	0.28	11.83	0.51	0.24
CodeLlama 13B	11.07	0.50	0.29	10.92	0.57	0.28
CodeLlama 13B Inst.	14.88	0.77	0.49	11.85	0.69	0.31
GPT-3.5-Turbo	24.03	3.46	2.12	18.80	2.53	1.49
GPT-4	27.30	3.89	2.76	21.87	3.32	1.68
Finetuning						
Llama3 8B	25.29	2.29	2.18	21.20	1.25	1.19
Llama3 8B Inst.	26.44	2.83	2.63	22.72	1.38	1.27
CodeLlama 7B	29.32	4.72	3.89	24.24	2.77	2.16
CodeLlama 7B Inst.	**30.53**	**5.80**	4.91	**26.70**	3.38	2.64
CodeLlama 13B	29.77	4.84	4.13	25.00	2.93	2.54
CodeLlama 13B Inst.	30.04	5.67	**5.12**	25.94	**4.04**	**3.45**

Table 3: Scores of the legal text visualization. **G**, **G-N** and **G-N-E** denote `Graph`, `Graph&Node` and `Graph&Node&Edge` respectively. The highest scores of each column are in bold.

end node v_e is represented by a tuple $e = [v_s, v_e, l]$, where l is a label of an edge. Nodes always include non-empty texts, while edge-label texts can be blank for edges without labels.

Graph code validation. First, we examine whether the generated code forms a valid graph $\mathcal{G}_h$ in terms of the DOT language. This is done by simply processing with the pydot library[3].

Nodes alignment by bipartite matching. Second, we extract nodes $\{v_h\}$ from $\mathcal{G}_h$ and align them with nodes from the reference graph: $\{v_r\}$ from $\mathcal{G}_r$ using the similarity of the texts in nodes. For this node alignment, we apply the bipartite matching problem to the sets of nodes $\{v_h\}$ and $\{v_r\}$, using the matching score function $s(v_r, v_h)$, which is computed from the BLEU scores of the text included in the reference and hypothesis nodes:

$$s(v_r, v_h) = \mathrm{BLEU}(v_r, v_h)$$

where the BLEU score is computed upon the texts of nodes. Given the scores between all reference and hypothesis nodes, we apply a bipartite matching solver in NetworkX[4] for aligning nodes of reference and hypothesis graphs.

Graph, node, edge-label evaluation. After we determined the node alignment, we performed three levels of evaluation of two graphs with textual labels. `Graph` is the F1 metrics of the matched edges after the node alignment. This metric is for the similarity measurement of the entire graph structure, ignoring the textual differences of nodes and edges after the alignment. `Graph&Node` is the metric where we use the BLEU score for the aligned nodes to penalize the cases where the two graphs have the same edges while the texts of the aligned nodes are different. Therefore the `Graph&Node` metric is sensitive to the difference of node texts compared with the `Graph` metric. Similarly, `Graph&Node&Edge` is a metric that considers node and edge text similarity in terms of the BLEU score. The details of computing these metrics are explained in Appendix E.

[3] https://github.com/pydot/pydot

[4] https://networkx.org/

5 Experiments

We evaluate the ability to visualize graphs from legal sentences with LegalViz. This involves representing legal entities as graph nodes, depicting legal actions, and rights as relations, and illustrating the legal basis of statements as graph nodes that link to other nodes.

5.1 Experimental settings

We conduct the DOT language code generation experiments with the publicly available Llama family models and GPT APIs via Microsoft Azure. For Llama family models, we experimented with the models specialized for code generation of CodeLlama and the recently released Llama-3 models. Specially we used `CodeLlama-7B` and `CodeLlama-7B-Instruct`, `CodeLlama-13B`, and `CodeLlama-13B-Instruct`, and Llama3 models of `Meta-Llama-3-8B` and `Meta-Llama-3-8B-Instruct`. Our experimental settings are two holds: few-shot generation and finetuning of the publicly available models. In few-shot experiments, we notice not only the GPT models but only publicly available Llama models are capable of producing valid DOT language codes without finetuning to some extent. We follow the supervised finetuning of Hugging Face with the detailed finetuning parameters in Appendix F. In evaluation, we generate ten different Graphviz code predictions for each model. We examine each prediction by the order of the probability of the generated sequences and evaluate the first prediction that forms a valid Graphviz code.

5.2 Result

Overall results. First, we conduct the few-shot and finetuning experiments with LegalViz dataset. Table 3 presents the experimental results of each models evaluated by `Graph`, `Graph&Node`, and `Graph&Node&Edge` metrics explained in Section 4. In the first look, we notice that our

Model	BG	ES	CS	DA	DE	ET	EL	EN	FR	HR	IT	LV	LT	HU	MT	NL	PL	PT	RO	SK	SL	FI	SV
Few-shot / Test / Graph																							
CodeLlama 13B Instruct	6.48	15.42	13.89	12.58	18.40	9.74	4.35	16.46	13.09	11.04	13.02	7.46	9.54	12.66	9.43	13.22	10.94	9.82	14.21	12.93	16.22	7.13	14.61
GPT-3.5-Turbo	13.27	26.24	13.35	16.77	23.84	17.96	16.64	17.53	19.83	17.42	20.77	18.23	17.85	18.44	13.99	18.61	19.40	21.62	15.88	16.80	22.07	21.03	24.85
GPT-4	23.50	19.90	18.44	23.77	21.69	21.14	22.18	24.34	22.41	23.79	17.93	24.66	24.35	20.28	18.06	19.46	17.15	19.59	22.27	19.46	22.48	27.86	28.25
Few-shot / Test / Graph&Node																							
CodeLlama 13B Instruct	0.00	1.02	0.00	0.39	0.90	0.56	1.22	1.96	0.85	1.23	1.24	0.33	0.84	0.57	0.31	0.00	0.98	0.61	1.42	0.68	0.60	0.00	0.17
GPT-3.5-Turbo	3.37	7.47	0.96	1.29	5.48	2.31	1.47	1.73	1.91	0.92	3.69	1.51	1.97	1.24	1.39	2.02	1.90	2.98	3.18	3.45	1.71	2.20	3.96
GPT-4	3.13	2.65	2.46	3.39	5.98	1.69	1.64	2.95	2.75	2.11	3.66	5.31	3.39	2.55	2.11	3.41	1.30	3.09	4.60	3.61	4.16	1.87	8.45
Finetuning / Test / Graph&Node&Edge																							
CodeLlama 13B Instruct	0.00	0.91	0.00	0.38	0.86	0.56	0.17	1.27	0.26	0.70	0.12	0.32	0.27	0.00	0.00	0.00	0.61	0.08	0.30	0.17	0.20	0.00	0.00
GPT-3.5-Turbo	1.78	5.42	0.51	0.88	3.52	1.19	0.74	1.27	1.04	0.35	2.21	0.65	0.33	0.63	0.79	1.43	1.68	1.11	1.92	1.86	1.41	0.67	2.87
GPT-4	0.54	1.63	0.93	2.09	4.58	0.87	0.31	1.21	0.32	1.10	1.25	3.33	1.77	0.45	0.62	2.74	0.69	0.56	2.10	0.95	2.18	0.88	7.58
Finetuning / Test / Graph																							
Llama3 3 8B Instruct	24.86	32.23	25.87	22.88	24.06	17.61	26.45	30.69	22.07	20.53	25.38	20.46	17.78	20.80	21.39	21.61	21.75	22.49	20.87	15.50	19.48	19.91	17.54
CodeLlama 7B Instruct	23.72	33.67	24.27	33.47	28.07	24.22	9.62	39.27	29.28	29.26	30.67	26.71	27.91	27.82	23.19	25.39	22.85	29.92	27.33	24.72	24.13	26.16	22.51
CodeLlama 13B Instruct	24.26	32.73	25.83	30.06	28.66	21.72	15.76	33.35	23.73	31.15	33.73	18.88	19.67	25.21	18.62	25.51	24.40	30.92	33.52	19.17	31.44	22.19	26.15
Finetuning / Test / Graph&Node																							
Llama3 8B Instruct	1.28	3.02	0.48	2.56	0.64	1.01	1.76	3.37	1.22	1.22	1.46	1.79	0.23	1.08	0.39	1.11	2.52	0.68	2.51	0.18	1.38	1.42	0.51
CodeLlama 7B Instruct	1.19	7.60	1.91	4.52	4.70	0.73	0.00	9.63	3.79	1.46	5.30	3.79	2.50	3.19	2.38	1.88	4.94	4.31	3.29	3.74	2.24	1.53	3.11
CodeLlama 13B Instruct	3.95	9.64	1.70	6.28	4.80	1.61	1.77	7.24	5.94	4.14	7.53	1.21	3.30	2.02	2.53	2.55	3.61	7.17	4.54	2.05	2.77	2.22	4.35
Finetuning / Test / Graph&Node&Edge																							
Llama3 8B Instruct	1.28	3.00	0.48	2.56	0.64	1.01	1.76	3.37	1.22	1.22	0.78	1.79	0.23	0.94	0.39	0.86	2.02	0.68	2.27	0.18	0.71	1.40	0.51
CodeLlama 7B Instruct	0.79	5.25	0.98	3.98	3.89	0.35	0.00	8.16	2.98	0.94	4.58	3.79	2.02	2.64	0.75	1.53	4.51	3.44	2.22	1.12	2.22	1.53	3.11
CodeLlama 13B Instruct	3.85	8.81	0.94	5.22	4.41	1.61	1.77	7.24	5.05	3.70	5.67	1.21	2.54	0.68	1.80	2.32	2.98	5.93	3.74	1.70	2.40	1.42	4.35

Table 4: Scores by 23 languages in EUR-LEX.

	Validation		Test	
Model	**Top1**	**Top10**	**Top1**	**Top10**
Few-shot				
Llama3 8B	42.17	93.83	37.65	89.30
Llama3 8B Instruct	47.83	98.43	47.13	97.30
CodeLlama 7B	18.35	86.96	16.78	85.22
CodeLlama 7B Instruct	43.30	91.91	37.65	89.39
CodeLlama 13B	18.09	84.96	17.30	85.04
CodeLlama 13B Instruct	38.26	89.74	33.39	88.70
GPT-3.5-Turbo	96.70	96.78	94.17	94.26
GPT-4	**98.87**	**98.96**	**99.04**	**99.13**
Finetuning				
Llama3 8B	74.96	97.13	68.09	93.74
Llama3 8B Instruct	84.43	98.61	80.09	95.13
CodeLlama 7B	86.52	98.00	80.70	94.70
CodeLlama 7B Instruct	88.61	96.26	81.74	93.74
CodeLlama 13B	88.09	96.52	81.39	93.83
CodeLlama 13B Instruct	85.57	96.09	75.83	91.13

Table 5: Success rate of creating valid graphs in top-1 and top-10 generated results. The highest scores of each columns are highlighted.

finetuned models outperformed few-shot counterparts and even GPT models, which are assumed to be larger than the Llama models, suggesting the effectiveness of our dataset for finetuning. Also, CodeLlama-13B-Instruct took the highest scores on `Graph&Node` in the test set, `Graph&Node&Edge` in the validation and test set. We also noticed that instruct-tuned models perform better than their base models, which can reflect the complexity of our task.

For the evaluation metric of `Graph`, all finetuned models perform close to GPT models, suggesting that the structure of the graphs can be grasped by GPT models without further training. However, comparing them in `Graph&Node` and `Graph&Node&Edge`, finetuned models performed better than few-shot models. This suggests that predicting detailed texts in graphs requires further tunings with LegalViz.

Scores by languages. Table 4 presents the results of models by all 23 languages in EUR-LEX. Among these languages, models perform relatively weakly in languages that have relatively fewer resources (Chalkidis et al., 2021a), such as Maltese, Latvian, Estonian, Lithuanian, and Slovene. For languages that have relatively more resources such as English and French, models tend to have high scores. This tendency is especially observed in the results of few-shot settings of Llama while this tendency becomes weaker in the finetuned models, suggesting the effectiveness of our training dataset covering 23 languages.

From a linguistic point of view, Hungarian and Finnish, belonging to the same Uralic language group, have low scores in each model. This may reflect their linguistic difference from other languages. Similarly, for the Romance language group, e.g., Romanian, French, Spanish, Italian, and Portuguese, models have moderate performances, seemingly better than those of the Uralic language group and languages that also have fewer resources than those of English and French.

Valid graph generation. We are also surprised that all models can produce valid Graphviz codes in most cases. Table 5 presents the success rate of forming valid graphs in terms of the DOT language of Graphviz. As explained in the experimental setting, we generated ten different instances. Here "Top1" is the success rate of forming a valid graph for the first instance and "Top10" is the success rate that at least one out of ten instances forms a valid graph. GPT models are most accurate to generate valid DOT language codes in all models while Llama3 8B can generate a valid DOT

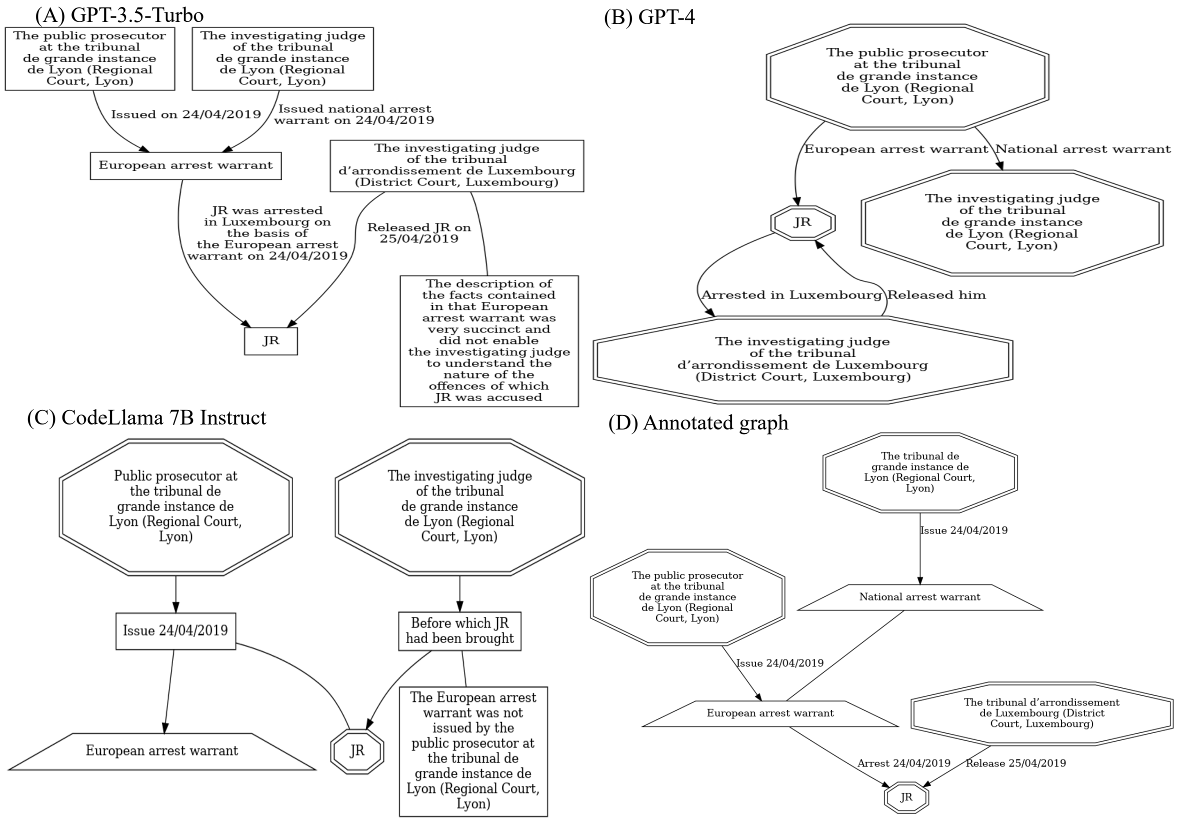

Figure 4: Qualitative analysis of diagrams drawn by Graphviz code. Each figures are generated by GPT-3.5-Turbo, GPT-4, CodeLlama 7B Instruct, and an annotated diagram.

language code in ten generations in the few-shot setting, suggesting that GPT models are *generalists* of generating graph codes. When finetuned, they become comparable with GPT models for generating valid codes, and indeed they exceed GPT models for generating legal diagrams as we have already seen in Table 3, suggesting that the finetuned models are *specialists* in the legal domain.

We also further discuss several generation experiments to survey which legal knowledge is effective in the generation in Appendix A.

6 Qualitative Analysis

Finally, we have conducted a qualitative analysis of few-shot GPT-3.5-Turbo, GPT-4, and CodeLlama 7B Instruct as CodeLlama-7B-Instruct scored a relatively high score on the F1 score comparison. Figure 4 presents the result of each graph generated by English input. Legal document is in Appendix G. Here, GPT-3.5-Turbo and GPT-4 failed to draw some nodes as legal entities with a double octagon shape and norm as a trapezium shape while CodeLlama 7B Instruct successfully illustrates them accordingly. The quality of the generated graphs was better in English and French while the generated graphs in languages with relatively fewer resources often include more errors than in English and French. For example, in languages including Bulgarian, Greek, Dutch, Danish, models can mistakenly generate two different nodes with very similar texts that are indeed the same node in the annotated graph, causing the structural errors of the entire graph. They sometimes even fail the coherent generation in one language, switching to another language during generation. The improvement of the generations in wide languages is the next step of future study.

7 Conclusion

We have proposed LegalViz, the first manually annotated dataset to visualize legal text with DOT language Graphviz and introduced a novel evaluation method taking into account both diagram visualization quality and sentences of graph nodes and relations We also observed the effectiveness of our dataset by conducting experiments in few-shot and finetuning models, comparing results by models, results by 23 languages, results of graph success rates, and qualitative analysis.

Limitation

LegalViz contains the same number of instances in 23 languages of EUR-LEX. However, this doesn't mean that the models with fintuned or few-shot have the same ability to treat all 23 languages equally. Especially models face difficulties in fewer language resources as we experimented. We cannot offer any warranty for using our dataset and models for real usages such as legal advice. We also consider that our dataset should be used with appropriate supervision by experts. This can be a *potential risk* when our dataset is misused. We assume that results of automatic visualizations by models are still different from the annotated visualizations in most cases, suggesting the current limitation of the generation.

Ethic Statements

The annotation material of this dataset is publicly available EU legal materials including judgments and orders, which do not include personal or sensitive information, with the exception of trivial information presented by consent, e.g., the names of the active presidents of the European Parliament, European Council, or other official administration bodies. The copyright for the editorial content of this website, the summaries of EU legislation, and the consolidated texts, which are owned by the EU, is licensed under the Creative Commons Attribution 4.0 International license.[5]

[5]https://eur-lex.europa.eu/content/legal-notice/legal-notice.html

References

Iosif Angelidis, Ilias Chalkidis, and Manolis Koubarakis. 2018. Named entity recognition, linking and generation for greek legislation. In *International Conference on Legal Knowledge and Information Systems*.

Dennis Aumiller, Satya Almasian, Sebastian Lackner, and Michael Gertz. 2021. Structural text segmentation of legal documents. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 2–11, New York, NY, USA. Association for Computing Machinery.

Dennis Aumiller, Ashish Chouhan, and Michael Gertz. 2022. EUR-lex-sum: A multi- and cross-lingual dataset for long-form summarization in the legal domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7626–7639, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Claire Barale, Mark Klaisoongnoen, Pasquale Minervini, Michael Rovatsos, and Nehal Bhuta. 2023. AsyLex: A dataset for legal language processing of refugee claims. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 244–257, Singapore. Association for Computational Linguistics.

Jonas Belouadi, Anne Lauscher, and Steffen Eger. 2024. Automatikz: Text-guided synthesis of scientific vector graphics with tikz. In *International Conference on Learning Representations (ICLR)*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, John A. Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuan-Fang Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, abs/2303.12712.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2019. Extreme multi-label legal text classification: A case study in EU legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 78–87, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021a. MultiEURLEX - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

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

Ilias Chalkidis, Nicolas Garneau, Catalina Goanta, Daniel Katz, and Anders Søgaard. 2023. LeXFiles and LegalLAMA: Facilitating English multinational legal language model development. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15513–15535, Toronto, Canada. Association for Computational Linguistics.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. 2022a. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330, Dublin, Ireland. Association for Computational Linguistics.

Ilias Chalkidis, Tommaso Pasini, Sheng Zhang, Letizia Tomada, Sebastian Schwemer, and Anders Søgaard. 2022b. FairLex: A multilingual benchmark for evaluating fairness in legal text processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4389–4406, Dublin, Ireland. Association for Computational Linguistics.

Daniel L. Chen and Jess Eagel. 2017. Can machine learning help predict the outcome of asylum adjudications? In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, page 237–240, New York, NY, USA. Association for Computing Machinery.

Huajie Chen, Deng Cai, Wei Dai, Zehui Dai, and Yadong Ding. 2019. Charge-based prison term prediction with deep gating network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6362–6367, Hong Kong, China. Association for Computational Linguistics.

Ona de Gibert Bonet, Aitor García Pablos, Montse Cuadros, and Maite Melero. 2022. Spanish datasets for sensitive entity detection in the legal domain. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3751–3760, Marseille, France. European Language Resources Association.

Kasper Drawzeski, Andrea Galassi, Agnieszka Jablonowska, Francesca Lagioia, Marco Lippi, Hans Wolfgang Micklitz, Giovanni Sartor, Giacomo Tagiuri, and Paolo Torroni. 2021. A corpus for multilingual analysis of online terms of service. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 1–8, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matt Dunn, Levent Sagun, Hale Şirin, and Daniel Chen. 2017. Early predictability of asylum court decisions. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, page 233–236, New York, NY, USA. Association for Computing Machinery.

Mohamed Elaraby and Diane Litman. 2022. ArgLegalSumm: Improving abstractive summarization of legal documents with argument mining. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6187–6194, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. 2020. A dataset for statutory reasoning in tax law entailment and question answering. In *NLLP@KDD*.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 487–498, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Wonseok Hwang, Dongjun Lee, Kyoungyeon Cho, Hanuhl Lee, and Minjoon Seo. 2024. A multi-task benchmark for korean legal language understanding and judgement prediction. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Michael J. Bommarito II, Daniel Martin Katz, and Eric M. Detterman. 2021. *Chapter 11: LexNLP: Natural language processing and information extraction for legal and regulatory texts*. Edward Elgar Publishing, Cheltenham, UK.

Daniel Martin Katz, Michael J. Bommarito, II, and Josh Blackman. 2017. A general approach for predicting the behavior of the supreme court of the united states. *PLOS ONE*, 12(4):1–18.

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael James Bommarito. 2023. Natural language processing in the legal domain. *ArXiv*, abs/2302.12039.

Arshdeep Kaur and Bojan Bozic. 2019. Convolutional neural network-based automatic prediction of judgments of the european court of human rights. In *Irish Conference on Artificial Intelligence and Cognitive Science*.

Pedro Henrique Luz de Araujo, Teófilo E. de Campos, Renato R. R. de Oliveira, Matheus Stauffer, Samuel Couto, and Paulo Bermejo. 2018. Lener-br: A dataset for named entity recognition in brazilian legal text. In *Computational Processing of the Portuguese Language*, pages 313–323, Cham. Springer International Publishing.

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(2):237–266.

Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 19–35, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. 2023. LEXTREME: A multi-lingual and multi-task benchmark for the legal domain. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3016–3054, Singapore. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. Technical report.

Vasile Pais, Maria Mitrofan, Carol Luca Gasan, Vlad Coneschi, and Alexandru Ianov. 2021. Named entity recognition in the Romanian legal domain. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 9–18, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Christos Papaloukas, Ilias Chalkidis, Konstantinos Athinaios, Despina Pantazi, and Manolis Koubarakis. 2021. Multi-granular legal topic classification on Greek legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 63–75, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. 2020. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1235–1241, Marseille, France. European Language Resources Association.

Stefanie Urchs., Jelena Mitrović., and Michael Granitzer. 2021. Design and implementation of german legal decision corpora. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 515–521. INSTICC, SciTePress.

Chaojun Xiao, Haoxiang Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *ArXiv*, abs/1807.02478.

Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. 2018. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. 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 1854–1864, New Orleans, Louisiana. Association for Computational Linguistics.

Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. 2023. Diagrammergpt: Generating open-domain, open-platform diagrams via llm planning.

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*, ICAIL '21, page 159–168, New York, NY, USA. Association for Computing Machinery.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.

		Validation			Test		
Model	**#**	**Graph**	**Graph&Node**	**Graph& Node&Edge**	**Graph**	**Graph&Node**	**Graph& Node& Edge**
CodeLlama 7B	0	29.32	4.72	3.89	24.24	2.77	2.16
CodeLlama 7B	1	28.13	4.35	3.66	22.22	3.17	2.42
CodeLlama 7B	2	28.29	3.76	3.03	23.60	3.27	2.36
CodeLlama 7B	3	28.74	4.31	3.69	24.83	3.01	2.36
CodeLlama 7B Instruct	0	30.53	5.80	4.91	26.70	3.38	2.64
CodeLlama 7B Instruct	1	29.11	4.70	4.22	24.63	3.51	2.75
CodeLlama 7B Instruct	2	29.94	5.12	4.28	25.40	3.64	2.80
CodeLlama 7B Instruct	3	31.00	5.01	4.34	26.43	3.69	2.93
CodeLlama 13B	0	29.77	4.84	4.13	25.00	2.93	2.54
CodeLlama 13B	1	30.76	5.22	4.84	23.40	3.59	3.06
CodeLlama 13B	2	30.04	5.23	4.60	24.18	3.63	3.03
CodeLlama 13B	3	30.66	4.61	4.20	24.52	3.12	2.80
CodeLlama 13B Instruct	0	30.04	5.67	5.12	25.94	4.04	3.45
CodeLlama 13B Instruct	1	26.33	4.26	3.94	22.98	3.55	2.82
CodeLlama 13B Instruct	2	28.18	5.14	4.68	22.58	3.67	2.88
CodeLlama 13B Instruct	3	27.93	4.99	4.42	22.69	3.63	3.01
Llama3 8B	0	25.29	2.29	2.18	21.20	1.25	1.19
Llama3 8B	1	23.22	1.54	1.29	21.09	0.99	0.93
Llama3 8B	2	22.22	1.74	1.43	20.93	1.03	0.93
Llama3 8B	3	24.89	2.32	2.12	20.39	0.93	0.90
Llama3 8B Instruct	0	26.44	2.83	2.63	22.72	1.38	1.27
Llama3 8B Instruct	1	23.59	1.92	1.69	21.76	1.25	1.19
Llama3 8B Instruct	2	23.78	1.63	1.48	20.90	1.26	1.10
Llama3 8B Instruct	3	25.25	2.74	2.60	22.69	1.44	1.35

Table 6: F1 score results of three types different legal knowledges experimented with finetuned models. #0: given normal prompt. #1: added the name of all graph nodes as prompt input. #2: added legal entities as prompt input. #3: added legal norms as prompt input.

A Effect of legal knowledge

In this experiment, we added additional information to the prompts to let models know how legal information should represented as nodes or edges. Added information are the following three types: (1) which words would be generated as graph nodes including legal entities and rules, (2) which legal entities would be generated as graph nodes, and (3) which rules would be generated as graph nodes. The result is given in Table 6. Detailed prompts are given in Appendix C. As an overall result, experiment (3) tends to be more effective in increasing the score of `Graph`, `Graph&Node`, and `Graph&Node&Edge` generation in both validation and test than experiment (1) and (2). However, all experiments (1) - (3) adding legal knowledge to prompt had lower scores than normal prompts.

B Graphviz annotation rule

The following is an example of the Graphviz code annotation rules.

```
[shape=doubleoctagon]: Entities which are capable to act as legal entity.
[shape=trapezium]: Any kinds of rules which are legally effective, applied to the
    present case or supporting legal statements.
[style=dotted]: Relationship of succession between 2 entities.
[dir=none]: Equivalent relationship, agreements, or connecting detailed explanation
    of other nodes.
[dir=none, style=bold]: Marital relationships or family relationships which have
    been established under civil law.
[style=dashed]: Expressing a legal right that cannot be exercised or not existed.
[shape=ellipse]: Expressing a person who is legally deceased.
```

C Prompt

The prompt for LLMs used in training, generation and dataset creation is presented in Table 7.

Method	Prompt
Prompt used for train and generation	Using the DOT language of Graphviz, draw a graph to explain legal entity nodes, legal relationships, legal statements and legal basis of them from given text, written in {language} text. Use “shape=trapezium” to represent a legally effective material and use “shape=doubleoctagon” to represent a legal entity in Graphviz code with {language}. At any time, reply only with the graphviz code.
Prompt for extraction	From legal text below of {language} language, extract the same meaning word or sentence as given English word to language. Please output only extracted result. Legal text: {legal text} Word or sentence to extract:
Prompt for translation	Translate below words or text from English to {language} Text:
Effect of legal knowledge (1)	Using the DOT language of Graphviz, draw a graph to explain legal entity nodes, legal relationships, legal statements and legal basis of them from given text. Use the following nodes in the graph. Nodes: {extracted nodes} Legal text: {legal text} Graphviz Code:
Effect of legal knowledge (2)	Using the DOT language of Graphviz, draw a graph to explain legal entity nodes, legal relationships, legal statements and legal basis of them from given text. Use the following legal entity in the graph. Legal entities: {extracted entity} Legal text: {legal text} Graphviz Code:
Effect of legal knowledge (3)	Using the DOT language of Graphviz, draw a graph to explain legal entity nodes, legal relationships, legal statements and legal basis of them from given text. Use the following legal norms in the graph. Legal norms: {extracted rules} Legal text: {legal text} Graphviz Code:

Table 7: The prompts used in the experiment and data processing. {legal text}, {language}, {extracted nodes}, {extracted entity}, {extracted rules}, and {extracted labels} indicate the place to insert.

D Train dataset examples

Dataset Example (1)


```
{'ID': '45',
 'category': 'EU law',
 'diagram_number': '7',
 'case_name': 'Case T-207/02: Nicoletta Falcone v Commission of the\nEuropean
   Communities',
 'case_number': 'C2005/006/64',
 'document_url': 'https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:C200
   5/006/64&qid=1713891140330',
 'year': '2004',
 'text': 'In Case T-207/02: Nicoletta Falcone, a candidate in Competition COM/A/10/0
   1, represented by M. Condinanzi, against Commission of the European Communities
   (Agent: J. Currall, assisted by A. Dal Ferro, with an address for service in
   Luxembourg)     application for annulment of the decision of 2 May 2002 of the
   selection board in Competition COM/A/10/01 to exclude the applicant from the
   written tests on the ground that she did not obtain sufficient marks to be
   included among the 400 best candidates     the Court of First Instance (Second
   Chamber), composed of J. Pirrung, President, A.W.H. Meij and N. Forwood, Judges;
    H. Jung, Registrar, has given a judgment on 26 October 2004, in which it:',
 'Graphviz': 'digraph {\n    rankdir=LR;\n    node [shape=box];\n\n    "Nicoletta
   Falcone" -> "M. Condinanzi" [label="represent" dir=none];\n    "The Comission of
    the European Comminities" -> "Nicoletta Falcone" [label="application for
   annulment of the decision of 2 May 2002 of the selection board in Competition
   COM/A/10/01 to exclude the applicant from the written tests on the ground that
   she did not obtain sufficient marks to be included among the 400 best candidates
   "];\n}',
 'language': 'English'
}
```

E Details of evaluation metrics

Based on the F1-score, which is widely used in the NLP community and derives from the elements in confusion matrix, say true positive (TP), false negative (FN), and false positive (FP):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

In this paper, we developed three metrics: `Graph`, `Graph&Node` and `Graph&Node&Edge` based on F1 scores with different TP counts. Before computing these metrics, we preliminary extract the sets of nodes in reference $\{v_r\}$ and hypothesis $\{v_h\}$. We also determine their alignment computed by the BLEU score as explained in Section 4. This alignment is expressed in a function that aligns a hypothesis node to a reference node if their counterpart node is found in the alignment:

$$a(v_h) = \begin{cases} v_r & (\text{if } v_h \text{ has aligned node in } \{v_r\}) \\ \emptyset & (\text{if } v_h \text{ has no aligned node in } \{v_r\}) \end{cases} \tag{4}$$

The reference graph is composed of a set of edges E_r and the hypothesis graph is composed by a set of edges E_h. Here, E_r include an edge $e_r = [v_{s,r}, v_{e,r}, l_r]$ that is an edge spanning from $v_{s,r}$ to $v_{e,r}$ with text label l_r. Similarly, E_h include an edge $e_h = [v_{s,h}, v_{e,h}, l_h]$ that is an edge spanning from $v_{s,h}$ to $v_{e,h}$ with text label l_h.

`Graph` considers the matching of edge nodes in the reference. Using the alignment function $a(\cdot)$

$$f_{\text{Graph}}(e_h, e_r) = \begin{cases} 1 & (\text{if } a(v_{s,h}) = v_{s,r} \text{ and } a(v_{e,h}) = v_{e,r}\ ) \\ 0 & (\text{otherwise}) \end{cases} \tag{5}$$

that considers only the alignment of the start and end nodes, ignoring node and label texts. Hereby `Graph` is computed from the following:

$$\text{TP} = \sum_{e_h \in E_h, e_r \in E_r} f_{\text{Graph}}(e_h, e_r) \tag{6}$$

$$\text{FP} = |E_h| - \text{TP} \tag{7}$$

$$\text{FN} = |E_r| - \text{TP} \tag{8}$$

where $|\cdot|$ is the number of entities in a set.

`Graph&Node` relies on BLEU scores of two node texts using the node-match function

$$f_{\text{Graph\&Node}}(e_h, e_r) = \begin{cases} \text{BLEU}(v_{s,h}, v_{s,r}) \cdot \text{BLEU}(v_{e,h}, v_{e,r}) & (\text{if } a(v_{s,h}) = v_{s,r} \text{ and } a(v_{e,h}) = v_{e,r}\ ) \\ 0 & (\text{otherwise}) \end{cases} \tag{9}$$

that is penalized by the difference of the start and end node texts. TP, FP, and FN are counted in the same equations Eq.6-8 replacing f_{Graph} with $f_{\text{Graph\&Node}}$.

Finally, `Graph&Node&Edge` further relies on BLEU scores of two label texts in addition to node texts using the following function:

$$f_{\text{Graph\&Node\&Edge}}(e_h, e_r) = \begin{cases} \text{BLEU}(v_{s,h}, v_{s,r}) \cdot \text{BLEU}(v_{e,h}, v_{e,r}) \cdot \text{BLEU}(l_h, l_r) & \\ & (\text{if } a(v_{s,h}) = v_{s,r} \text{ and } a(v_{e,h}) = v_{e,r}\ ) \\ 0 & (\text{otherwise}) \end{cases} \tag{10}$$

. This is the most strict evaluation by penalizing the difference of the reference and hypothesis node text and edge labels. Note that in some cases edges do not have labels. In that case, we assume $\mathrm{BLEU}(l_h, l_r) = 1$ if $l_r = \emptyset$ and $l_r = \emptyset$, otherwise $\mathrm{BLEU}(l_h, l_r) = 0$. This means that if both reference and hypothesis graphs has no edge labels, `Graph&Node` and `Graph&Node&Edge` become the identical score.

We reported the micro-averaged F1 scores for all three metrics.

F Detailed experimental settings

For training of LLMs, we follow the default setting of Hugging Face supervised finetuning of the trl[6] library for the optimizers and schedulers. We use the mini-batch size of 32. We use the max token length of 4096 for training as we notice some languages, e.g., Greek, require longer tokens than other languages depending on Llama tokenizers. In finetuning, we use FP32 precision and all trainable parameters are updated. All Llama-family experiments are done on a single node with four NVIDIA A100 GPUs.

G Qualitative analysis input

The legal text used the qualitative analysis is the following:

> On 24 April 2019, the public prosecutor at the tribunal de grande instance de Lyon (Regional Court, Lyon) issued a European arrest warrant in connection with criminal proceedings in respect of JR, suspected of having been involved in offences linked to a criminal organisation. The warrant was issued pursuant to a national arrest warrant issued on the same day by the investigating judge of the tribunal de grande instance de Lyon (Regional Court, Lyon). On the same day, JR was arrested in Luxembourg on the basis of the European arrest warrant. However, on 25 April 2019, the investigating judge of the tribunal d'arrondissement de Luxembourg (District Court, Luxembourg) before which JR had been brought, released him after concluding that the description of the facts contained in that European arrest warrant was very succinct and did not enable the investigating judge to understand the nature of the offences of which JR was accused.

[6] https://github.com/huggingface/trl