Artificial Intelligence and Law
https://doi.org/10.1007/s10506-024-09425-7

ORIGINAL RESEARCH

A case study for automated attribute extraction from legal documents using large language models

Subinay Adhikary[1] · **Procheta Sen**[2] · **Dwaipayan Roy**[1] · **Kripabandhu Ghosh**[1]

Accepted: 24 October 2024

Abstract

The escalating number of pending cases is a growing concern worldwide. Recent advancements in digitization have opened up possibilities for leveraging artificial intelligence (AI) tools in the processing of legal documents. Adopting a structured representation for legal documents, as opposed to a mere bag-of-words flat text representation, can significantly enhance processing capabilities. With the aim of achieving this objective, we put forward a set of diverse attributes for criminal case proceedings. To enhance the effectiveness of automatically extracting these attributes from legal documents within a sequence labeling framework, we propose the utilization of a few-shot learning approach based on Large Language Models (LLMs). Moreover, we demonstrate the efficacy of the extracted attributes in downstream tasks, such as *legal judgment prediction and legal statute prediction*.

Keywords Large language model · Legal attribute · Sequence labeling · Weak supervison

✉ Procheta Sen
procheta.sen@liverpool.ac.uk

Subinay Adhikary
subinayadhikary612@gmail.com

Dwaipayan Roy
dwaipayan.roy@iiserkol.ac.in

Kripabandhu Ghosh
kripa.ghosh@gmail.com

[1] Indian Institute of Science Education and Research Kolkata, Mohanpur, India

[2] University of Liverpool (UOL), Liverpool, UK

Published online: 11 November 2024

1 Introduction

Intuitively, several pieces of information can be found in the legal case documents in the unstructured form, including details about the parties involved (*appellant and respondent*), the court in which the case is being appealed (e.g., *Supreme Court, High Court, District Court*), the motive of the incident, and the judgment itself. Since legal case documents often encompass lengthy and intricate sentences, making it challenging and time-consuming to thoroughly read and comprehend the entire content of a case document (Bhattacharya et al. 2022), extracting information from legal documents presents a formidable challenge to the research community. However, a structured representation of a document (shown in Fig. 1) is better (Munir and Sheraz Anjum 2018; Horrocks 2013) in terms of efficiency in processing any downstream task (e.g., *Prior case retrieval* (Bhattacharya et al. 2022), *Judgement prediction* (Malik et al. 2021), *etc.*). There exists several methods for extracting information from legal documents (e.g., *catchphrase extraction* (Mandal et al. 2021), *witness identification* (Ghosh et al. 2020), *etc.*). However, these techniques often fall short of adequately capturing the thematic essence of a case.

As we show in Fig. 2, we engaged in extensive consultations with legal practitioners, resulting in the identification of a solution: the extraction of fine-grained information (which we term as *concepts, attributes, or tags*). Our primary focus is centered on uncovering a methodology for extracting the thematic perspective from legal cases.

Fine-grained information: Typically, to obtain an initial understanding of crime case documents, Indian Penal Code (IPC) sections are commonly employed. For instance, if a given case document includes IPC sections 300 and 302 along with

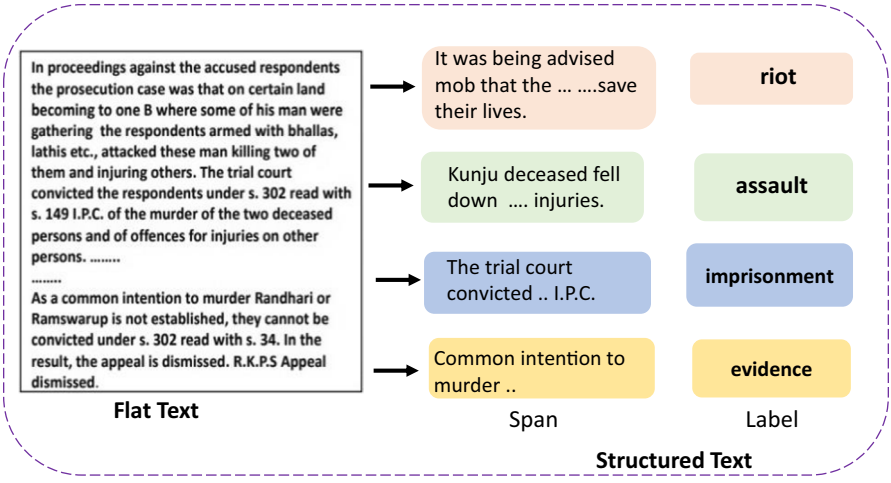

Fig. 1 Illustrates that a legal case document contains several information, such as case facts, arguments, and judgment, presented in a *flat text format*. Legal experts extract fine-grained concepts (e.g. riot, evidence, assault, etc.) from the case to represent it in a *structural format*

... that the dacoit standing at the main gate of the compound on the ground floor fired his gun which hit her mother on the forehead as a result of which she died on the spot. Usha Rani further deposed that thereafter when she peeped through the window the same person carrying the gun and standing near the main gate of the compound commanded her to go back and sit quietly otherwise she would also meet the same fate.This part of the evidence of Usha Rani was attacked by the learned counsel for the appellant by contending that after her mother was shot dead it was improbable that Usha Rani would have dared to peep outside endangering her own life and this part of the evidence has been introduced by the prosecution in order to show that the witness had seen the miscreants so that she may identify the miscreants at a later stage. We are unable to accept these submissions ...

Fig. 2 An excerpt from a case document from the Supreme Court of India collection, where the highlighted span has been annotated with the labels `witness_testimony`, indicates that the other party has contested the witness testimony presented in favor of the prosecution or the defense and also whether the court has agreed to such a challenge. It represents a piece of *fine-grained information*, that is highlighted by the span of text

other sections, it indicates a case of intentional murder, with the potential for the accused to face life imprisonment. Such information offers a high-level, coarse-grained perspective of the case document. However, our primary goal is to gather more profound insights from the documents, encompassing aspects such as the motive behind the crime, the influence of witnesses on the judgment, the rationale for the judgment, and more, as described in Fig. 2. These deeper insights contribute to the creation of concepts or attributes of the document, as illustrated in Sect. 3.

Our primary objective centers on annotating the case documents, and to initiate this annotation process, the construction of an thematic presentation of the document takes precedence. To achieve the thematic presentation, extensive consultations were conducted with legal experts (from West Bengal National University of Juridical Sciences) including criminal lawyers, and in-depth discussions transpired. To capture the more nuanced information from the legal documents, we delved extensively into the complexity of fine-grained information and finally, we came up with 7 different attributes or concepts as detailed in Sect. 3.

Law professionals were appointed to annotate 200 legal documents (significant annotations compared to other legal works in India, where (Bhattacharya et al. 2019) utilized 50 documents, (Bhattacharya et al. 2022) utilized 100 documents) with defined attributes as shown in Fig. 2. Having legal experts annotate a large number of documents can be a very time-consuming and expensive task. Consequently, we explore sequence labeling approaches using deep neural networks and large language models (LLMs) to effectively extract the concepts or attributes from legal documents, and that can reduce enormous manual effort.

Novelty of this work: Sequence labeling approaches are used for entity or attribute extraction for various domains. A limited number of manual annotations may not

be sufficient for training a sequence labeling model. As a result of this, we propose LLM based weak supervision approach to increase the number of training samples for the sequence labeling model. More precisely, we used fine-tuning and few-shot learning approaches to generate weakly supervised samples for training a sequence labeling model. Broadly speaking, the contributions of this paper are as follows.

1. We propose a new framework to present each legal document in terms of different important *concepts*, *attributes*, or *tags* (the term *concepts*,*attributes* and *tags* will be used interchangeably in this paper), as detailed in Sect. 3.
2. We propose a novel *weak supervision-based* approach to train a sequence labeling model that can be used to effectively extract concepts from legal documents,[1] as illustrated in Sect. 4.
3. We also demonstrate that structured representation enhances the performance of different downstream tasks namely *Judgement Prediction* and *Statute prediction*, as discussed in Sects. 5.2 and 5.3.

2 Related work

Existing literature related to our research scope can be broadly categorized into three areas: a) Legal Document Representation Techniques b) Entity Recognition and c) Large language model application in NLP. Each one of them is described as follows.

2.1 Legal document representation techniques

In legal cases, the documents often encompass lengthy and intricate sentences, making it challenging and time-consuming to thoroughly read and comprehend the entire content of a case document. To alleviate this extensive effort, researchers have emphasized the extraction of noun phrases known as *catchphrases* (Tran et al. 2018; Mandal et al. 2017) from the document. This approach aims to capture the key elements and central themes of the text. Additionally, *summarization* (Anand and Wagh 2022; Shukla et al. 2022; Bhattacharya et al. 2021) techniques have been proven to be effective in gaining a comprehensive understanding of the document by condensing its content into a concise summary. However, to get the *thematic view or topical representation* of the document, we need to extract fine-grained level information or concepts (as discussed above) from the document.

2.2 Entity recognition

Named Entity recognition [13] is a well established task in NLP. Entities (e.g., persons, organizations, countries) are defined as objects which have independent existence. Domain-specific entities (e.g. biomedical entity (Kalim et al. 2022)) are in

[1] The implementation of the proposed model is available at https://github.com/subinay494/Legal_Attributes_Extraction/.

general defined by domain experts through an ontology structure. Initially rule-based systems (Sari et al. 2010), and syntactical structure (Zhang et al. 2014) were used to automatically extract entities from a text. With the advancement of machine learning and deep learning techniques existing research has used conditional random fields, sequence-to-sequence models, and large language models for *Named entity recognition* (Curran and Clark 2003; Hammerton 2003). More specifically *legal name entities* extraction has been implemented using Conditional Random Fields (CRF by a sequence tagger, specifically the BiLSTM-CRF model). Named entity recognition is a task that involves automatically identifying text spans that could potentially represent named entities, such as the names of people, buildings, or institutions. In this research, we use the terms 'attribute extraction' and 'named entity extraction' interchangeably. Attributes refer to key concepts identified by legal experts (shown in Fig. 1), which are similar to entities defined in other domains.

Attribute extraction or entity extraction in the legal domain presents two primary challenges. First, there are no pre-existing, well-defined entity sets specific to the legal field. To address this, we consulted with legal experts to identify relevant entities. The second challenge is data availability. Annotations by legal professionals are expensive, resulting in a limited dataset (Bhattacharya et al. 2022). However, training an effective entity extraction pipeline typically requires a large amount of data (Wang et al. 2023). We primarily focus on this challenge in this work by using weak supervision techniques to train an entity extraction pipeline. The study in Naik et al. (2023) extracted named entities from legal documents using state-of-the-art named entity recognition approaches. Similarly, to extract the meta-information from the legal documents, the research community employed *Requirements Engineering* extensively (Sleimi et al. 2018; Kiyavitskaya et al. 2006).

2.3 LLM application in NLP

In recent literature Large Language Models (LLMs) have been successfully applied in a range of Natural Language Processing tasks like Machine Translation (Zhang et al. 2023), Summarisation (Liu et al. 2023), Entity Recognition [25]. The inception of LLMs started with Transformer architecture (Vaswani et al. 2017). Transformer architecture mostly used an attention mechanism and was successful in generalizing most of the tasks. Broadly speaking, LLMs are used in three different ways in NLP tasks a) Pre-trained models, b) Fine-Tuned models, and c) Prompt based applications. Pre-trained models are trained on a large amount of unannotated data through self-supervised training and can be applied on any kind of NLP task. Pre-trained models like BERT (Devlin et al. 2019), GPT-2 (Radford et al. 2019) have overperformed state-of-the-art baselines in NLP tasks (Wang et al. 2022). Fine-tuned models (Xu et al. 2021) are trained on a particular task and while fine-tuning the model is initialized with the pre-trained parameters. Existing research has shown that LLMs fine-tuned on specific downstream tasks perform better than training the model from scratch on the downstream tasks (Jensen and Plank 2022). With the

recent release of GPT–3.5,[2] LLAMA-2[3] and GPT-4[4] prompt base learning have also gained attention. In prompt learning (Liu et al. 2021), frozen LLMs are used with task specific prompts (i.e. set of keywords or tokens) to get better outputs for that particular task. Prompt-based learning is less memory-consuming than training a model from scratch or fine-tuning (Brown et al. 2020). In this work, we used both fine-tuning and prompt learning for automatically annotating legal documents with tags.

3 Automated attribute extraction

Legal document annotation framework: The corpus encompasses a total of 33545 case documents of the Supreme Court of India,[5] encompassing a diverse array of case topics, including but not limited to 'Criminal', 'Land and Property ', 'Constitutional', and 'Intellectual Property Rights', and more. We systematically extracted sections and acts from all the case documents using a well-established section extractor,[6] where it follows a rule-based mechanism to extract the statutes. The inclusion of criminal case documents in this study was deliberate, as a significant portion of the cases we encountered pertained to criminal matters. Notably, the Indian Penal Code (IPC) comprises 511 sections distributed across 23 chapters, and the majority of cases related to IPC 302 are consistent with our observations since capital punishment applies to this section. Particularly, those case documents are selected discussing IPC 299, IPC 300, IPC 304, and IPC 307,[7] etc. We propose seven important legal concepts or attributes corresponding to a criminal case document with the help of law experts. The different concepts and the corresponding descriptions are as follows.

1. *Expert witness Testimony (ExpWitTest):* If a legal document contains witness testimony from expert people like forensic and ballistic experts then we consider that the particular document has the concept 'Expert Witness Testimony' attribute present in it.
2. *Witness Testimony (WitTest)*: If a legal document mentions a witness collected from non-expert people (e.g., a common man who witnessed an event) then the document is tagged with 'Witness Testimony' tag.
3. *Assault:* If there is mention of someone hurt by a sharp weapon or gunshot then the corresponding legal document is tagged with 'Assault' tag.
4. *Riot:* A legal document is tagged with 'Riot' tag if there is mention of a disturbance of the peace by several persons, assembled and acting with a common intent in executing an unlawful enterprise in a violent and turbulent manner.

[2] https://platform.openai.com/docs/models/gpt-3-5.

[3] https://www.llama.com/llama2/.

[4] https://platform.openai.com/docs/models/gpt-4.

[5] https://main.sci.gov.in/.

[6] https://github.com/upalbhattacharya/indian-legal-dataset-preparation/tree/master/src/preprocess.

[7] The details of these IPC sections are briefly discussed in https://github.com/Law-AI21/llm_nlp.

5. *Homicide:* The tag 'Homicide' is associated with the concept where there has been an attempt to murder. The attempt may or may not be successful.
6. *Imprisonment:* A legal document is tagged with the 'Imprisonment' tag if there is a mention of a murderer who was sentenced to life imprisonment or rigorous imprisonment for a few years.
7. *Evidence:* Evidence (i.e. direct or circumstantial) or evidence inconsistency that affects arguments, then the corresponding legal document is tagged with the tag 'Evidence'.

In each document, law students labeled the text span (part of the sentence or the full sentence) corresponding to a concept present in that legal document using an annotation tool LeDA (Adhikary et al. 2023). Law practitioners (from West Bengal National University of Juridical Sciences) annotated 200 legal documents with gold standard annotations.

*Problem Definition*It is difficult to manually annotate a large number of legal documents, since annotations by legal experts are very expensive. However, to exploit the attributes corresponding to each case in a downstream task we would need a lot of annotated documents. As a result of this, we focus on the automatic annotation of legal documents with the attributes or concepts. We cast the automatic annotation problem as a sequence labeling problem (Rei and Søgaard 2018). In this context, each token in the input text is assigned a corresponding label. Sequence labeling has been extensively employed in NLP research to tackle various challenges (Fan et al. 2019). The advantage of casting the automatic annotation of a document into a sequence labeling problem is that along with all the attributes mentioned in a case, we can also obtain the highlighted sequence of tokens responsible for generating that attribute.

As described above, there are 7 important concepts or labels in a legal document. All the tokens in a legal document may not be useful for annotation. Consequently, we introduced an additional attribute named 'NoTag' to complement the aforementioned seven concepts (i.e. 8 in total).

3.1 Methodology

Here, we describe the working principle of a sequence labeling model. If x is an input sequence (i.e. $x = x_1, x_2, \ldots x_n$) and y is the desired output sequence (i.e. $y = y_1, y_2, \ldots y_{n'}$), then $n = n'$. The sequence labeling problem learns a mapping $H : X \xrightarrow{Y}$ using a state-of-the-art BiLSTM-CRF (Huang et al. 2015)model. The BiLSTM-CRF network essentially computes the maximum likelihood of each sequence label as follows.

$$P(y_{0,\ldots n}|r_{0\ldots n}) = \prod_{i=1}^{n} f(y_{i-1}, y_i, x) \tag{1}$$

In Eq. 1, $f(y_{i-1}, y_i, x)$ estimates the likelihood of an individual label (i.e. y_i) based on its neighboring sequence. The hidden state output vectors (r_i) obtained from a BiLSTM model are used to estimate the likelihood in $f(y_{i-1}, y_i, x)$.

3.2 Weak supervision framework

Acquiring a substantial volume of manually annotated gold standard data is a time-consuming and costly task. Consequently, we employed various weak supervision approaches to augment the number of training samples, ultimately aiming to enhance the accuracy of the sequence labeling model. We broadly categorize the weak supervision approaches into two categories. Each one of them is described as follows.

The detailed dataset statistics for the train and test data are given in Table 1.

Pseudo relevance feedback based weak supervision (PRWS): In information retrieval literature the top-K documents retrieved corresponding to a query are assumed as pseudo relevant documents corresponding to the query (Lavrenko and Croft 2001). The pseudo relevance documents are eventually used to improve the retrieval performance of the query. In the context of this problem set up, we consider the manually highlighted spans corresponding to the annotated documents as queries. The highlighted spans are then used to retrieve similar sentences from the collection of all legal case documents. We combine the top-K sentences corresponding to each highlighted span using the CombSUM (Fox and Shaw 1993) approach. Top k' sentences from the combined ranked list for a particular tag are considered as weak supervision examples for that particular attribute.

Large language modeling based weak supervision (LLMWS): One of the limitations of the PRWS approach mentioned above is that if there is not enough legal proceeding related to a particular tag in the legal document collection, then it will be challenging to come up with meaningful weakly supervised samples for the corresponding tag. Consequently, we also explore LLM based approaches to provide weakly supervised samples for a tag. LLMs are pretrained on a large dataset and can generate new samples for different kinds of tags. In LLM based approaches we broadly used fine-tuning and few-shot learning to generate training samples for a particular tag. We used GPT-2 (Radford et al. 2019) for fine-tuning and GPT–3.5 turbo, GPT-4, Llama-2 for few-shot learning approach. The gold standard annotations are used as few-shot examples or as training sets for fine-tuning an LLM. The reason for using two different versions of LLMs is to explore the performance difference between an open-sourced (i.e. GPT-2) and paid version (i.e. GPT–3.5) of LLMs. In spite of having good performance on most of the tasks, GPT–3.5 is also costly and there is a security issue involved with it. For example, in the context of cybersecurity GRC (Governance, Risk management, and Compliance), GPT models can have technical, legal, and ethical implications, especially for high-security clients with sensitive information(Rivas and Zhao 2023).

Figure 3 shows a schematic diagram of the training data generation process using a weak supervision approach and it also shows how a trained sequence labeling model eventually generates attribute sequences for any text.

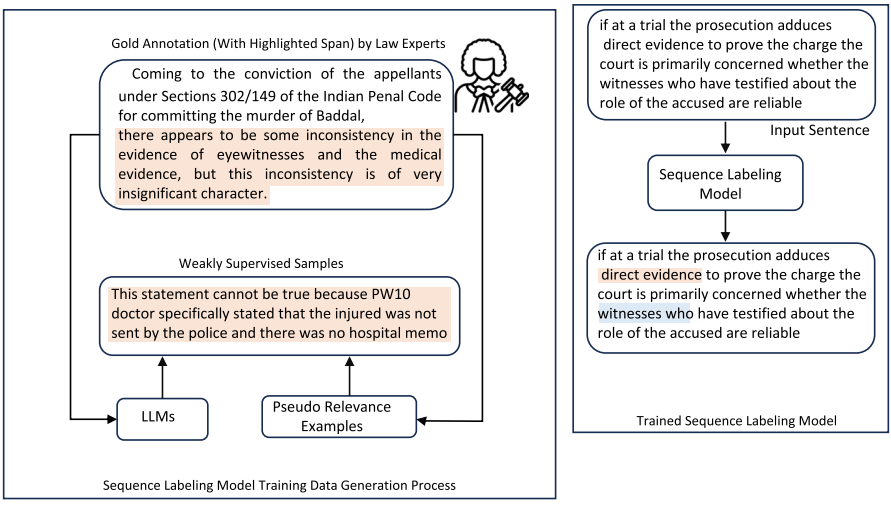

Fig. 3 Workflow diagram of the sequence labeling framework with weak supervision

4 Experiment setup

The objectives of the experiments are two fold. In the first set of experiments, we verified whether weak supervision can improve the baseline sequence labeling performance or not. In the second set of experiments, the objective is to show the effectiveness of the tags obtained from the sequence labeling approach in the downstream task. For the two different categories of experiments, we used different datasets.

Dataset and preprocessing: We used broadly two different kinds of dataset for all our experiments. The first dataset we used is the gold standard 200 legal documents annotated by law students (as described in Sect. 3). The details of the dataset is described in Table 1. The dataset is split into train and test set. Later weakly supervised samples were added in the training set. However, the evaluation is done only on gold standard annotation data.

The second type of dataset is used for pseudo relevance-based weak supervision approaches. Here, the source of crime-related data is the collection of legal case documents (i.e. 6854 documents) took place in the Supreme Court of India from 1958 to 2016. The case proceedings were crawled from a publicly available website.[8] The collection of court proceedings were indexed using Lucene 8.11.[9] As described in Sect. 3.2 for PRWS approach we retrieved sentences as weak supervision candidates. Consequently, the indexing unit in this problem setup is sentence. The index has only two fields: a) *DocId* and b) *Sentence*. For LLMWS category of approaches, no external data was required.

[8] http://www.liiofindia.org/in/cases/cen/INSC/

[9] https://lucene.apache.org/

Table 1 Down-scaled dataset details for automatic attribute extraction

Dataset		Tags						
Type	Statistics	ExpWittet	Wittest	Homicide	Assault	Imprisonment	Riot	Evidence
Train (- WS)	#sentences	31	35	50	39	50	32	24
Train (- WS)	#tokens	502	930	501	734	509	797	580
Train (+ WS)	#Sentences	31	35	50	39	50	37	34
Train (+ WS)	#tokens	502	930	501	734	509	915	762
Test	#sentences	372	162	78	48	34	20	10
Test	#tokens	7789	3841	1036	755	344	410	257

As a preprocessing step, each legal case document was split into sentences using sentence tokenizer from NLTK.[10] Then each highlighted sentence was split into tokens based on whitespace. While preparing the dataset for the training of the sequence labeling model, the words not highlighted within a sentence are tagged with `NoTag`. The sequence labeling approach implemented in Flair model[11] is used for all our experiments. Flair model considers each sentence within a legal document as a separate instance. Since `Riot` and `Evidence` tags have a comparatively low number of annotations (i.e. see Table 1), weak supervision examples were considered only for above mentioned tags.

For pseudo relevance based experiments, we initially explored top $5, 10, \ldots, 15$ as the value of k in top k documents and got the best result on the combination of k=5 for `riot` and k=10 for `Evidence`, as shown in Fig. 4. Consequently, for all the results reported in Table 2, we used $k = 5$ for *riot* and $k = 10$ for *Evidence*. In the LLMWS experiments, we primarily utilized four types of language models: GPT-2, GPT−3.5, LLAMA-2, and GPT-4. For GPT−3.5, GPT-4, and $LLAMA2_{pt}$, we employed prompts to generate samples, with Fig. 3 displaying the exact prompt used. To maintain consistency, the same prompt was applied across all three models. During generation, we randomly chose 10 clean examples. For the fine-tuned versions of GPT-2 and LLAMA-2, we utilized all annotated samples from the training set to fine-tune the model. The fine-tuned model was then used to generate potential candidates for different tags. For GPT−3.5 based experiments we fixed the temperature parameter to 0.8[12] after exploring all the values from 0 to 2 with an interval of 0.1. To finetune GPT-2, we used all the gold standard samples present for each tag in the train set as described in Table 1. To provide some statistical comparison of the text generator models, we calculated the Flesch-Kincaid readability score[13] of the generated samples (as shown in Table 3).

The different approaches used in our experiment setup can be broadly categorized into four groups. Each one of them are described as follows.

[10] https://www.nltk.org/api/nltk.tokenize.html

[11] https://pypi.org/project/flair/

[12] https://community.openai.com/t/cheat-sheet-mastering-temperature-and-top-p-in-chatgpt-api/172683

[13] https://pypi.org/project/py-readability-metrics/

Table 2 Automated Tag Extraction Results using Flair, Llama-2 model produces best overall accuracy

Method		Accuracy							
WS Approach	LLM	ExpWitTest	WitTest	Homicide	Assault	Imprisonment	Riot	Evidence	Overall
-WS	–	0.85	**0.50**	0.18	0.39	0.40	0.78	0.24	0.67
NC	–	0.21	0.05	0.11	0.04	0.02	0.03	0.01	–
PRWS	–	0.83	0.40	0.23	0.46	0.38	0.49	0.14	0.63
$PRWS_{ILBERT}$	–	0.83	0.40	0.23	0.46	0.38	0.59	0.14	0.63
$PRWS_{BERT}$	–	0.86	0.50	0.34	0.41	0.46	0.75	0.15	0.66
LLMWS	$GPT-3.5$	**0.88**	0.47	0.26	0.51	**0.57**	0.77	0.26	0.69
LLMWS	$GPT-2_{PT}$	0.86	0.49	0.24	0.50	0.56	**0.80**	0.18	0.66
LLMWS	$GPT-2_{FT}$	0.84	0.45	0.27	0.45	0.51	0.82	0.27	0.65
LLMWS	$Llama2-7bPT$	0.86	0.64	0.19	0.50	0.52	0.64	0.31	0.69
LLMWS	$Llama2-7bFT$	0.87	**0.72**	0.28	**0.53**	0.47	0.71	**0.37**	**0.72**
LLMWS	$GPT-4$	0.84	0.42	0.23	0.42	0.52	0.81	0.21	0.65
LLMWS+PRWS	$GPT-3.5$	0.82	0.50	0.27	0.45	0.49	0.76	0.26	0.65
LLMWS+PRWS	$GPT-2_{FT}$	**0.88**	0.46	0.15	0.57	0.51	0.51	0.24	0.67
LLMWS (SeqLabeler)	Llama2-7bFT	0.17	0.46	0.41	0.24	0.26	0.26	0.18	0.32

We bold the hightest F-score value of each attribute along with overall highest model accuracy

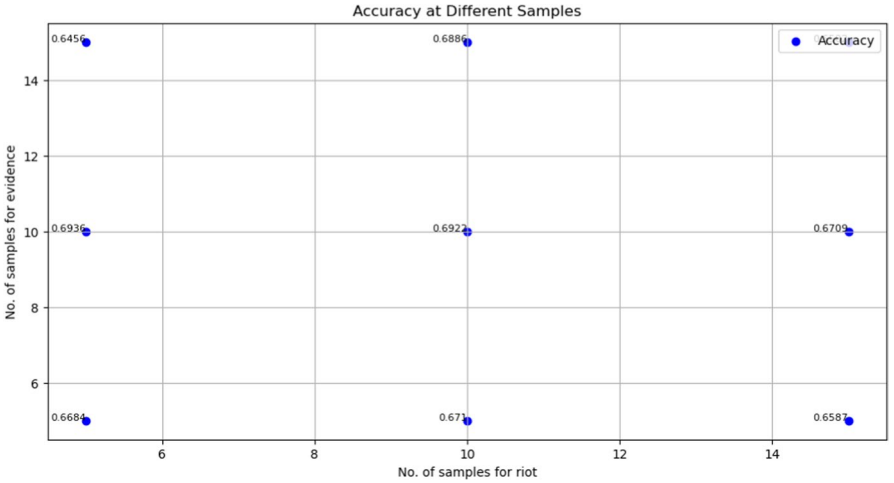

Fig. 4 This illustrates that how overall accuracy of the model varies with different combination of samples for the concepts `riot` and `evidence`, as utilized in validation set for pseudo relevance-based emperical study. For the combination of 5 samples for *riot* and 10 samples for *evidence* yeilds best performance, achieving an accuracy of 0.6936

WS: In this approach we do not use any kind of weak supervision and use only gold standard annotation data to finetune the pre-trained sequence labeling model in Flair.

NC: In this approach we use a simple naive base classifier to estimate the likelihood of different attributes given a token in a sentence. A token is assigned to an attribute having maximum likelihood. The reason for using NC as one of our baselines is to ensure whether automatic extraction of attributes from a legal document at all requires a sequence labeling setup.

PRWS: In this approach we use the pseudo relevance examples as a weak supervision technique to generate more training samples for fine-tuning the sequence labeling model. We explored three different variations of PRWS approach based on the re-ranking applied to the pseudo-relevance samples obtained from the retrieval model. Re-ranking using InLegalBERT (Paul et al. 2023) and BERT model are named as $PRWS_{InLegalLBERT}$ and $PRWS_{BERT}$. The objective of re-ranking

Table 3 Readability scores: GPT-2 generated text has the highest score, higher score indicates easier readability

	Pseudo-relavance	GPT-2	GPT−3.5
Readability score	33.3	74.6	27.6
Avg. no. of words per sentence	36	28	18
Word overlap	144	61	34
Similarity	.83	.74	.61

Prompt= " Your task is to generate 10 statements that closely resemble the statements given in the Examples delimited by {}. The statements in the example resemble a tag {tag name} . The generated statements should also resemble the same tag. Statements that imply some impending discrepancy in facts generally resemble {tag name}. Each statement must not have less than 25 words. Examples: {*examples*} "

Fig. 5 This prompt has been used to generate samples for the `riot` and `evidence` respectively

is to obtain better quality weak supervision samples compared to only a pseudo relevance-based approach.

LLMWS: In this category of approaches LLMs are used to generate new training samples. We tried three different variations of this model. In *LLMWS(GPT-2*$_{PT}$*)*, *LLMWS(Llama2-7b*$_{PT}$*)*, *LLMWS(Llama2-7b*$_{FT}$*)* and *LLMWS(GPT-2*$_{FT}$*)*, we used pre-trained and fine-tuned GPT-2 and Llama2-7b models to generate new samples respectively for finetuning the sequence labeling model. In *LLMWS(GPT-3.5)* and *LLMWS(GPT-4)*, few shot learning is used to generate weakly supervised samples from GPT-3.5-turbo model and GPT-4 model, as shown in the Fig. 5.

LLMWS+PRWS: In this category of approaches we used weakly supervised samples from both LLMs and pseudo relevance feedback to finetune the sequence labeling model. There are two different variantions of this approach based on which LLM was used to generate samples (i.e. PRWS+LLMWS$_{GPT-2_{FT}}$ and PRWS+LLMWS$_{GPT-3.5}$).

For all the approaches mentioned above, we used Flair as the sequence labeling model, as it is a state-of-the-art model. However, for comparison purposes, we also applied the best supervision approach to another sequence labeling model called Sequence Labeler (***LLMWS(SeqLabeler)***)[14] in Table 2.

5 Results

5.1 Automated legal entity extraction

The different observations from Table 2 are as follows. In terms of overall accuracy LLMWS$_{Llama-2}$ has performed the best. However, no single method has performed the best for all eight tags. Generally speaking, 'ExpWittest' has the best accuracy rate among all the different tags for all the methods reported in Table 2. One possible reason may be that 'Expwittest' has the highest number of gold annotations. PRWS$_{BERT}$ has performed better than PRWS. This shows that re-ranking has provided better weak supervision examples than standard pseudo-relevance results. The fact that GPT-3.5-turbo performs the best shows that the

[14] https://github.com/marekrei/sequence-labeler

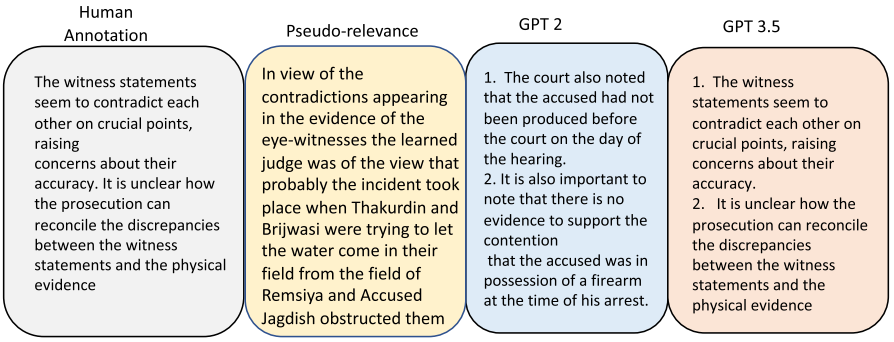

Fig. 6 Comparison of Evidence Samples from Human Annotation Vs. Weakly Supervised Samples Generated from LLMs (i.e. GPT-2 and GPT−3.5) and Pseudo Relevance Feedback

Table 4 Error analysis: It represents that weak supervision yields better performance than compared to absence of weak supervision. For the sentence, "**But then another mob came which according to PW1 consisted of 200–1200 by PW2 according to PW4 the mob consisted of 100 persons.**", the model predicts "wittest" (Predicted Label) instead of "riot" (True Label), when weak supervision is not used, which affects overall model performance

		Predicted label	
Word	True Label	Without_weak_supervision	With_weak_supervision
But	**riot**	wittest	**riot**
then	**riot**	wittest	**riot**
another	**riot**	wittest	**riot**
mob	**riot**	wittest	**riot**
came	**riot**	wittest	**riot**
which	**riot**	wittest	**riot**
according	**riot**	wittest	**riot**
to	**riot**	wittest	**riot**
PW1	**riot**	wittest	**riot**
consisted	**riot**	wittest	**riot**
of	**riot**	wittest	**riot**
200-1200	**riot**	wittest	**riot**
by	**riot**	wittest	**riot**
PW2	**riot**	wittest	**riot**
according	**riot**	wittest	**riot**
to	**riot**	wittest	**riot**
PW4	**riot**	wittest	**riot**
the	**riot**	wittest	**riot**
mob	**riot**	wittest	**riot**
consisted	**riot**	wittest	**riot**
of	**riot**	wittest	**riot**
100	**riot**	wittest	**riot**
persons	**riot**	wittest	**riot**

On the other side, adding LLMs generated samples with the existing data for the concept `riot`, i.e., applying weak supervision, improves model's overall accuracy

Table 5 Error analysis: It Illustrates that weak supervision affects model performance. For the sentence, "**Dr. PauLs testimony thus creates some doubt regarding the reliability of the prosecution evidence that Joseph had received injury with a wooden spear at the hand of accused No 6.**", the model predicts "wittest" (Predicted Label) instead of "evidence" (True Label) in the absence weak supervision, that drops overall model performance

		Predicted label	
Word	True label	Without_weak_supervision	With_weak_supervision
Dr.	**evidence**	wittest	**evidence**
Pauls	**evidence**	wittest	**evidence**
testimony	**evidence**	wittest	**evidence**
thus	**evidence**	**evidence**	**evidence**
creates	**evidence**	**evidence**	**evidence**
some	**evidence**	**evidence**	**evidence**
doubt	**evidence**	**evidence**	**evidence**
regarding	**evidence**	**evidence**	**evidence**
the	**evidence**	**evidence**	**evidence**
reliability	**evidence**	wittest	**evidence**
of	**evidence**	wittest	**evidence**
the	**evidence**	wittest	**evidence**
prosecution	**evidence**	wittest	**evidence**
evidence	**evidence**	wittest	**evidence**
that	**evidence**	wittest	**evidence**
Joseph	**evidence**	wittest	**evidence**
had	**evidence**	wittest	**evidence**
received	**evidence**	wittest	**evidence**
injury	**evidence**	wittest	**evidence**
with	**evidence**	wittest	**evidence**
a	**evidence**	wittest	**evidence**
spear	**evidence**	wittest	**evidence**
at	**evidence**	wittest	**evidence**
the	**evidence**	wittest	**evidence**
hand	**evidence**	wittest	**evidence**
of	**evidence**	wittest	**evidence**
accused	**evidence**	wittest	**evidence**
No	**evidence**	wittest	**evidence**
6	**evidence**	wittest	**evidence**

On the other hand, adding LLMs generated samples with the annotated data for the concept `evidence`, i.e., applying weak supervision, enhances model's overall accuracy

few-shot learning approach in LLM can be used to solve the issue of not having enough gold standard annotations in the legal domain. The accuracy for `NoTag` was more than 80% for all the approaches. Consequently, we didn't report it in Table 2.

Table 2 shows a comparative analysis of the weakly supervised samples generated from PRWS and LLM based approaches. Given that Llama-2 was the highest performing model in Table 2, it is likely that Llama-2 was generating more diverse

content comared to other models (i.e. GPT-2 and PRWS). Figure 6 shows the samples generated from different approaches.

Tables 4 and 5 present an error analysis, illustrating how the use of "weak supervision" aids the model in accurately predicting the word label, thereby enhancing overall model accuracy. As we previously discussed in Sect. 4, we employ LLMs to generate training samples for *riot* and *evidence*. This "weak supervision" approach enhances the performance of `riot` and `evidence`, as shown in Table 2.

After completion of fine-tuning the model, we focus on showcasing the effectiveness of automated extracted text segments in different downstream tasks such as *Judgement Prediction* (Strickson and De La Iglesia 2020) and *Statute Prediction* (Vats et al. 2023). Extracted text segments contain specific information about the case documents such as relevant legal provisions, precedents, and legal reasoning, all of which can contribute to predicting the likely outcome. That motivates us to utilize these segments in *judgment prediction* and *stature prediction*, where specific information or patterns that align with reasoning play an important role (Vats et al. 2023; Paul et al. 2020).

5.2 Judgement prediction

To showcase the efficacy of the structured representation of legal documents, we delved into a downstream task namely legal judgment prediction. We cast judgment prediction as a binary classification problem. The two classes (i.e. class 0 and class 1) correspond to the two scenarios where judgment was made in favor or against the appeal. We used the dataset published in Malik et al. (2021) for our experiment purpose. The dataset[15] contains 5, 082 documents for training and 94 crime-related case documents for testing. We explored both BERT-uncased and InLegalBert embeddings to obtain a semantic representation corresponding to a legal document.

To investigate the effectiveness of the legal document attributes, we explored both 'Text+Tag' and 'Text+span' approaches in Table 6. For the 'Text+Tag' version, we appended all the extracted attributes of each test document to its corresponding content. Similarly for the 'Text+Span' version, we appended the highlighted texts (as shown in Fig. 8) corresponding to each tag to the content of the document. Table 6 shows that for both BERT and InLegalBERT variations the 'Text+Tag' and 'Text+Span' versions performed better compared to using only the text versions. It is also interesting to observe that the 'Text+Span' version is performing better compared to 'Text+Tag' version. Furthermore, it is noteworthy that 89.71% of the spans originate from beyond the last 512 tokens (i.e., Which was used for Text Only versions) across all 94 documents. This observation underscores the additional information contributed by these spans, leading to better performance in classification (i.e., Additional 21 documents were correctly classified in 'Text+Span' version compared to Text only version).

We employ Large Languege Models (LLMs) for the judgement prediction. Figure 7 demonstrates that we follow in-context learning (ICL) approach (Brown et al.

[15] https://github.com/Exploration-Lab/CJPE

Prompt= "**Given a Supreme Court of India case proceeding enclosed in angle brackets ⟨⟩, your task is to predict the decision or verdict of the case (for the appellant).**
Prediction: Given a case proceeding, the task is to predict decision 0 or 1, where label 1 corresponds to the acceptance of the appeal or petition of the appellant or petitioner and label 0 corresponds to the rejection of the appeal or petition of the appellant or petitioner.

Context:
case_proceeding: "F. NARIMAN, J. Leave granted. ..."
Prediction: 1
case_proceeding: "This appeal by special leave is .."
Prediction: 0

Instructions:
Learn from the above given two examples and perform the task for the following case proceeding. **Give the output predicted case decision as either 0 or 1.**
case_proceeding: ⟨ case_procedding ⟩ "

Fig. 7 Prompt Template Overview: For the Judgement Prediction, we provide case documents as the examples of both appeal acceptance and rejection in the prompt. Finally, the label for the test document is predicted by Language Models (LLMs) based on the provided **Instructions**

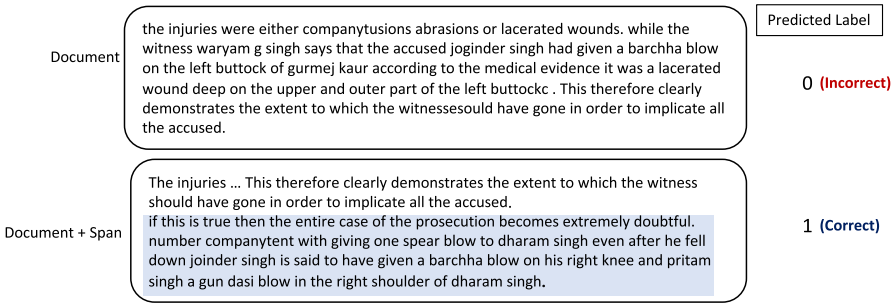

Fig. 8 The incorporation of spans along with text enriches the information content, consequently leading to a reduction in misclassifications

2020), utilizing one case document for each class (accept and reject) from the training set as the examples, i.e. few-shot learning. In Table 6, we show that 'Text+span' yields best performance for both the LLMs. Hence, we can infer that the highlighted sections corresponding to each tag hold greater significance in representing legal documents compared to the tag names themselves.

Table 6 In the Judgement Prediction Results, the addition of extracted spans with text contributes more fine-grained information, which increases overall accuracy

Method		Metric						
		Acc (Class 0)			Acc (Class 1)			
Embedding type	Input format	Precision	Recall	F1-score	Precision	Recall	F1-score	Overall Acc
BERT	Text	0.50	0.34	0.40	0.54	0.70	0.60	0.53
BERT	Text+Tag	0.54	0.52	0.52	0.59	0.62	0.60	0.56
BERT	Text+Span	0.61	0.25	0.35	0.56	0.86	0.67	**0.57**
InLegalBERT	Text	0.57	0.36	0.44	0.57	0.76	0.65	0.57
InLegalBERT	Text + Tag	0.63	0.31	0.41	0.58	0.84	0.68	0.60
InLegalBERT	Text +Span	**0.75**	0.40	0.52	0.62	**0.88**	**0.72**	**0.66**
Llama3-70b	Text	0.65	0.60	0.63	0.58	0.64	0.61	0.62
Llama3-70b	Text + Tag	0.67	0.62	0.65	0.60	0.66	0.63	0.64
Llama3-70b	Text +Span	0.69	**0.70**	**0.69**	**0.65**	0.64	0.64	**0.67**
Gemini 1.0 Pro	Text	0.31	0.10	0.15	0.42	0.75	0.54	0.40
Gemini 1.0 Pro	Text + Tag	0.40	0.16	0.23	0.43	0.73	0.54	0.43
Gemini 1.0 Pro	Text+span	0.48	0.20	0.28	0.45	0.75	0.56	**0.46**

We bold the highest value of precision, recall, and F-score for each class regardless of the method. Additionaly, we bold hightest overall accuracy for each method.

Prompt= " Given a part of Supreme Court of India case proceeding enclosed in angle brackets ⟨ ⟩

List of Statutes

Statute=[

"Statute #1": "Statute #1 Description"

"Statute #2": "Statute #2 Description"

.....

]

Instructions:

Ensure that the statutes generated as responses are from Statute. Avoid generating fabricated or invalid tags. **Your task is to identify statues from the statute for this legal document in the list form.**

Fact_statement: ⟨ fact ⟩ "

Fig. 9 Prompt Template Overview: For the Statute Prediction, each statute is outlined in **List of Statutes**. The labels for the test document are predicted by Language Models (LLMs) based on the provided **Instructions**

5.3 Statute prediction

Table 7 Addition of extracted spans with text impacts more fine-grained information, which yields better performance in statute prediction

	IPC Section-wise Accuracy						
Model	Input Format	148	300	302	304	307	**Avg**
GPT–3.5-turbo	Text	0.23	0.0	0.94	0.50	0.46	0.42
GPT–3.5-turbo	Span	0.28	0.0	0.87	0.63	0.42	0.44
GPT–3.5-turbo	Text + Span	0.20	0.0	0.96	0.58	0.63	**0.46**
GPT-4	Text	0.85	0.43	0.95	0.58	0.52	0.65
GPT-4	Span	0.86	0.46	0.96	0.63	0.55	0.69
GPT-4	Text + Span	0.89	0.48	0.96	0.86	0.55	**0.75**
Llama3-70b	Text	0.63	0.1	0.73	0.14	0.60	0.58
Llama3-70b	Span	0.64	0.02	0.94	0.30	0.56	0.60
Llama3-70b	Text + Span	0.59	0.1	0.93	0.28	0.58	**0.61**
Gemini 1.0 Pro	Text	0.27	0.14	0.90	0.27	0.32	0.50
Gemini 1.0 Pro	Span	0.86	0.40	0.97	0.76	0.53	0.75
Gemini 1.0 Pro	Text + Span	0.85	0.42	0.96	0.79	0.54	**0.76**

We bold highest avg value for each model, demonstrating that "Text+Span" produces the best outcomes

We cast the statute prediction problem as a multi-label classification problem and the facts of the case were employed as the basis for analysis. We have collected a dataset from Podder and Bhattacharya (2020) for this experiment. For the training phase a total of 200 documents were utilized for Indian Penal Code (IPC) 148, 300, 302, 304, and 307.[16] The reason for choosing only the five statutes mentioned above for the classifier is that the attributes proposed in Sect. 3 apply to only criminal case documents.

In this experiment, we considered those 200 documents as the testing set, and statute descriptions were utilized in the prompt, as shown in Fig. 9. By analyzing the impact of predicted tags on statute prediction, this experiment aimed to shed light on the effectiveness and significance of incorporating additional information in multi-label classification tasks. The experiment results presented in Table 7 shows that the variant 'Span' and 'Text +Span' overperforms the 'Text' only version.

6 Conclusion and future work

In this work, we propose a weak supervision-based sequence labeling approach that leverages LLMs to generate samples for different predefined attributes for a legal case document. Experimental results show that among black box LLMs, LLAMA-2 performs the best for generating new samples for riot and evidence. We also show the effectiveness of extracted attributes in two downstream tasks namely *judgment prediction* and *statute prediction*. Notably, our results show that

[16] The details of these IPC sections are briefly discussed in https://github.com/subinay494/Legal_Attributes_Extraction/.

augmenting the highlighted spans of the corresponding attributes in legal case documents improves the performance of the downstream tasks.

The proposed approach works mainly on criminal case documents based on Indian Law. In future, we plan to extend the setup proposed in this paper across different domains (e.g. civil cases, intellectual property law cases etc.) of law and also to the law of different countries.

Acknowledgements We express our gratitute to the AI4ICPS Innovation Hub for supporting our research by awarding Chanakya Ph.D. Fellowship to the first author.

References

Adhikary S, Roy D, Ganguly D, Kumar Guha S, Ghosh K (2023) Leda: a system for legal data annotation. Front Art Intel Appl 379:367–370

Anand D, Wagh R (2022) Effective deep learning approaches for summarization of legal texts. J King Saud Univ- Computer Inf Sci 34(5):2141–2150. https://doi.org/10.1016/j.jksuci.2019.11.015

Bhattacharya P, Ghosh K, Pal A, Ghosh S (2022) Legal case document similarity: you need both network and text. Inf Proc Manage 59(6):103069

Bhattacharya P, Poddar S, Rudra K, Ghosh K, Ghosh S (2021) Incorporating domain knowledge for extractive summarization of legal case documents. Proc. of ICAIL

Bhattacharya P, Paul S, Ghosh K, Ghosh S, Wyner A (2019) Identification of rhetorical roles of sentences in indian legal judgments. In: Araszkiewicz, M., Rodríguez-Doncel, V. (eds) Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901

Curran JR, Clark S (2003) Language independent ner using a maximum entropy tagger. In: Daelemans, W., Osborne, M. (eds) Proceedings of CoNLL-2003. Edmonton, Canada

Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). Association for Computational Linguistics,

Fan Z, Wu Z, Dai X-Y, Huang S, Chen J (2019) Target-oriented opinion words extraction with target-fused neural sequence labeling. In: Proc. of NAACL 2019, pp. 2509–2518

Fox EA, Shaw JA (1993) Combination of multiple searches. In: Text Retrieval Conference

Ghosh K, Pawar S, Palshikar G, Bhattacharyya P, Varma V (2020) Retrieval of prior court cases using witness testimonies. IOS Press, Legal Knowledge and Information Systems

Hammerton J (2003) Named entity recognition with long short-term memory. In: Daelemans, W., Osborne, M. (eds.) Proceedings of CoNLL-2003. Edmonton, Canada

Horrocks I (2013) What Are Ontologies Good For? Springer, Evolution of semantic systems

Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF Models for Sequence Tagging

Jensen KN, Plank B (2022) Fine-tuning vs from scratch: Do vision & language models have similar capabilities on out-of-distribution visual question answering? Proc. of LREC

Kalim WB, Mercer RE (2022) Method entity extraction from biomedical texts. In: Proc. of the 29th International Conference on Computational Linguistics, pp. 2357–2362

Kiyavitskaya N, Zeni N, Mich L, Cordy JR, Mylopoulos J (2006) Text mining through semi automatic semantic annotation. In: Practical Aspects of Knowledge Management: 6th International Conference, PAKM 2006, Vienna, Austria, November 30-December1, 2006. Proceedings 6, pp. 143–154. Springer

Lavrenko V, Croft WB (2001) Relevance based language models. In: Proc. of SIGIR, pp. 120–127

Liu Y, Fabbri AR, Liu P, Radev D, Cohan A (2023) On Learning to Summarize with Large Language Models as References

Liu P, Yuan W, Fu J, Jiang Z, Hayashi H, Neubig G (2021) Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. CoRR

Malik V, Sanjay R, Nigam SK, Ghosh K, Guha SK, Bhattacharya A, Modi A (2021) ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. Proc ACL, pp 4046–4062

Mandal A, Ghosh K, Ghosh S, Mandal S (2021) A sequence labeling model for catchphrase identification from legal case documents. Artificial Intelligence and Law, pp 1–34

Mandal A, Ghosh K, Pal A, Ghosh S (2017) Automatic catchphrase identification from legal court case documents. In: Proc. of CIKM

Munir K, Sheraz Anjum M (2018) The use of ontologies for effective knowledge modelling and information retrieval. Appl Comput Inf 14(2):116–126

Naik V, Patel P, Kannan R (2023) Legal entity extraction: an experimental study of ner approach for legal documents. Int J Adv Computer Sci Appl. https://doi.org/10.14569/IJACSA.2023.0140389

Paul S, Mandal A, Goyal P, Ghosh S (2023) Pre-trained language models for the legal domain: a case study on indian law. In: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, pp. 187–196

Paul S, Goyal P, Ghosh S (2020) Automatic charge identification from facts: A few sentence-level charge annotations is all you need. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 1011–1022

Podder RS, Bhattacharya P (2020) Unsupervised legal concept extraction from indian case documents using statutes. In: Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation, pp. 62–65

Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I et al (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):9

Rei M, Søgaard A (2018) Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. Proc.NAACL, pp. 293–302

Rivas P, Zhao L (2023) Marketing with chatgpt: Navigating the ethical terrain of gpt-based chatbot technology. AI 4(2):375–384

Sari Y, Hassan MF, Zamin N (2010) Rule-based pattern extractor and named entity recognition: A hybrid approach. In: 2010 International Symposium on Information Technology, vol. 2, pp. 563–568

Shukla A, Bhattacharya P, Poddar S, Mukherjee R, Ghosh K, Goyal P, Ghosh S (2022) Legal case document summarization: Extractive and abstractive methods and their evaluation. In: Proc. of the 2nd Conference of the Asia-Pacific Chapter of the ACL and the 12th International Joint Conference on Natural Language Processing, Online only

Sleimi A, Sannier N, Sabetzadeh M, Briand L, Dann J (2018) Automated extraction of semantic legal metadata using natural language processing. In: 2018 IEEE 26th International Requirements Engineering Conference (RE), pp. 124–135. IEEE

Strickson B, De La Iglesia B (2020) Legal judgement prediction for uk courts. In: Proceedings of the 3rd International Conference on Information Science and Systems, pp. 204–209

Tran VD, Nguyen ML, Satoh K (2018) Automatic catchphrase extraction from legal case documents via scoring using deep neural networks. CoRR

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Lu, Polosukhin I (2017) Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) NEURIPS

Vats S, Zope A, De S, Sharma A, Bhattacharya U, Nigam S, Guha S, Rudra K, Ghosh K (2023) Llms–the good, the bad or the indispensable?: A use case on legal statute prediction and legal judgment

prediction on indian court cases. In: Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 12451–12474

Wang Y, Wang Y, Sun Z, Li Y, Hu S, Ye Y (2023) Deep purified feature mining model for joint named entity recognition and relation extraction. Inf Proce Manage 60(6):103511

Wang S, Sun X, Li X, Ouyang R, Wu F, Zhang T, Li J, Wang G GPT-NER: Named Entity Recognition Via Large Language Models. arXiv preprint arXiv:2304.10428

Wang H, Li J, Wu H, Hovy E, Sun Y (2022) Pre-trained language models and their applications. Engineering

Xu R, Luo F, Zhang Z, Tan C, Chang B, Huang S, Huang F (2021) Raise a child in large language model: Towards effective and generalizable fine-tuning. Online and Punta Cana, Dominican Republic, Proc. of EMNLP

Zhang X, Li D, Wu X (2014) Parsing named entity as syntactic structure. In: Interspeech. Fifteenth Annual Conference of the International Speech Communication Association

Zhang B, Haddow B, Birch A (2023) Prompting Large Language Model for Machine Translation: A Case Study