Artificial Intelligence and Law
https://doi.org/10.1007/s10506-025-09448-8

ORIGINAL RESEARCH

Leveraging LLMs for legal terms extraction with limited annotated data

Julien Breton[1,2] · Mokhtar Mokhtar Billami[2] · Max Chevalier[1] · Ha Thanh Nguyen[3] · Ken Satoh[3] · Cassia Trojahn[1] · May Myo Zin[3]

Accepted: 25 February 2025

Abstract

The legal industry is characterized by the presence of dense and complex documents, which necessitate automatic processing methods to manage and analyse large volumes of data. Traditional methods for extracting legal information depend heavily on substantial quantities of annotated data during the training phase. However, a question arises on how to extract information effectively in contexts that do not favour the utilization of annotated data. This study investigates the application of Large Language Models (LLMs) as a transformative solution for the extraction of legal terms, presenting a novel approach to overcome the constraints associated with the need for extensive annotated datasets. Our research delved into methods such as prompt-engineering and fine-tuning to enhance their performance. We evaluated and compared, to a rule-based and BERT systems, the performance of four LLMs: GPT-4, Miqu-1-70b, Mixtral-8x7b, and Mistral-7b, within the scope of limited annotated data availability. We implemented and assessed our methodologies using Luxembourg's traffic regulations as a case study. Our findings underscore the capacity of LLMs to successfully deal with legal terms extraction, emphasizing the benefits of one-shot and zero-shot learning capabilities in reducing reliance on annotated data by reaching 0.690 F1 Score. Moreover, our study sheds light on the optimal practices for employing LLMs in the processing of legal information, offering insights into the challenges and limitations, including issues related to terms boundary extraction.

Keywords Legal terms extraction · Limited annotated data · Large language models (LLMs) · One-shot learning · Fine-tuning · GPT-4

This work has been submitted for the Special Issue on Applications and Evaluation of Large Language Models in the Legal Domain. This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011014922 made by GENCI.

Extended author information available on the last page of the article

Published online: 14 March 2025

1 Introduction

The legal industry is characterized by an extensive volume of documents, including contracts, legislation, court rulings, and regulatory filings. These documents are dense, complex, and in constant evolution (Sassier and Lansoy 2008), making their analysis and processing both time-consuming and prone to human error. The automation of these processes, therefore, offers significant benefits. It not only accelerates the review and management of legal documents but also improves compliance, accuracy, and accessibility of legal information. Moreover, as laws and regulations frequently change, automated systems also ensure that analyses are up-to-date with the latest legal standards.

Term extraction in the legal domain involves the identification and categorization of terms that have legal significance. This task is crucial for several applications, including legal research, compliance monitoring, contract analysis, and case preparation. By automatically extracting terms, legal professionals can quickly locate relevant information, understand the relationships between different legal terms, and extract insights from large volumes of text.

In this study, our attention is directed towards the extraction of legal terms, which is the first step to reason on the legal rules. Upon acquiring rules that are amenable to computer processing, a multitude of applications have been delineated in the scientific literature, including, but not limited to, formal logic, case similarity or question-answering. However, domain-specific (e.g. legal) extraction presents unique challenges. Legal texts often contain complex sentence structures, domain-specific jargon, and ambiguous terms that can vary in meaning based on context. Figure 1 illustrates a real example of legal terms extraction from the Luxembourg's traffic law provided in article (Sleimi et al. 2018).

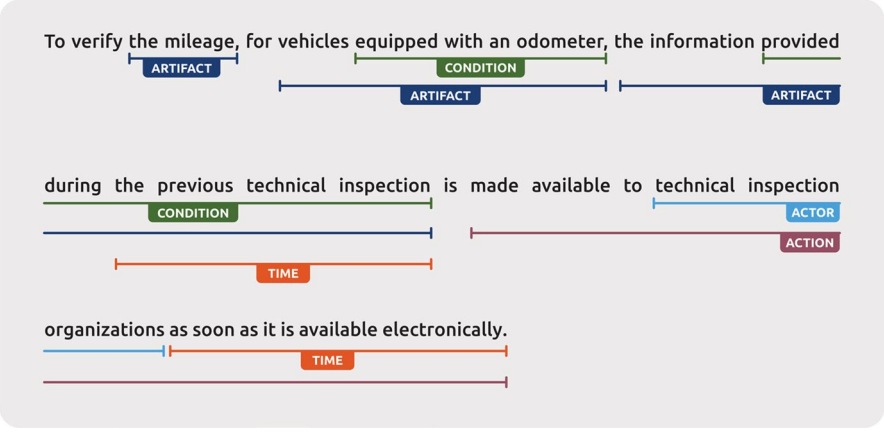

Fig. 1 Legal terms extractions from the following sentence: "To verify the mileage, for vehicles equipped with an odometer, the information provided during the previous technical inspection is made available to technical inspection organizations as soon as it is available electronically"

Various techniques have been developed and explored in the literature for terms extraction, including rule-based systems, Bi-directional Long Short-Term Memory (Bi-LSTM) networks (Huang et al. 2015), and BERT (Bidirectional Encoder Representations from Transformers) models (Devlin et al. 2018). Each of these methods has its own set of advantages and application contexts. Rule-based systems rely on a set of predefined rules and patterns identified by experts, making them highly interpretable but often less flexible. Bi-LSTM networks, a type of recurrent neural network, are adept at capturing the sequential nature of text data, making them useful for tasks involving context understanding over longer sequences. BERT models, leveraging transformer architectures, excel in understanding the context of words in a sentence by applying the attention mechanism (Vaswani 2017), which has significantly advanced the state of the art across a wide range of natural language processing tasks. However, a common challenge across all these techniques is their dependency on large amounts of annotated data during training. Annotated datasets are crucial for training these models to the specific tasks they need to perform, such as identifying and classifying terms within text. The creation of such datasets involves manually labelling text with the correct annotations, a process that is both time-consuming and resource-intensive. This dependency on extensive annotated data limits the scalability of deploying these models, especially in domains where annotated data is scarce or expensive to produce.

Large Language Models (LLMs) now present a compelling alternative through zero-shot and few-shot learning capabilities, effectively addressing the challenge of dependency on large volumes of annotated data for specific task training. These opportunities are enabled through the development of foundational models, which are constructed using extensive datasets and are poised for customization and application in specific tasks.

The following research questions (RQ) will drive our study:

- RQ1: Do LLMs present a viable solution to the challenges associated with the requirement for annotated data in legal information extraction?
- RQ2: What's the best strategy (prompt-engineering, fine-tuning) to maximize the performance of LLMs in legal terms extraction?

In the forthcoming section on related work, we will revisit the automation within the legal domain, focusing on the introduction of rule-based systems and the integration of deep learning technologies. Additionally, we will discuss the emergence and ascension of LLMs in recent years.

After this discussion, the methodology section will provide an in-depth analysis of the dataset and elaborate on the various strategies we have developed. In our article, we have opted to leverage four models for an information extraction task, described in Fig. 2, in limited annotated data context: GPT-4, Miqu-1-70b, Mixtral-8x7b and Mistral-7b. To obtained annotated legal terms and their matching legal concepts, we evaluate two strategies: Prompt-Engineering and Fine-Tuning. Subsequently, we included an evaluation using a state-of-the-art BERT model to benchmark and compare the effectiveness of these new approaches.

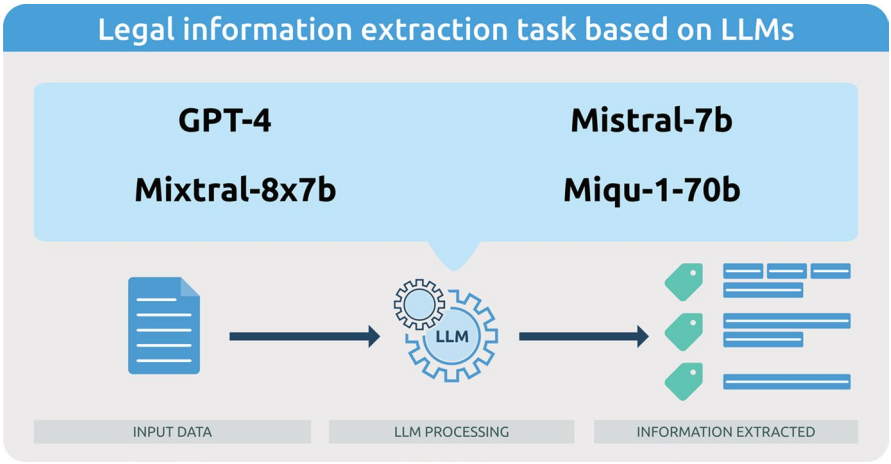

Fig. 2 Schematic overview of the legal terms extraction process introduced in our article

Prior to concluding this article, the evaluation section will furnish comprehensive results on the application of LLMs for legal terms extraction, while also providing the evaluation framework and addressing the limitations of the LLMs, such as the "boundary issue".

2 Related work

The interest within the scientific community in automating legal information extraction and reasoning reflects a targeted effort to improve the legal automated processing. This endeavour encompasses a diverse set of tasks, each targeting a specific aspect of legal information processing.

Firstly, case similarity involves the development of algorithms capable of identifying precedents and relevant cases based on legal facts, outcomes, and juridical principles, thereby facilitating a more informed legal decision-making process vmo narcha2021automated,moodley2019similarity. This task leverages natural language processing (NLP) techniques to analyse legal texts, extracting patterns and insights that can predict case outcomes or suggest relevant prior cases (Mandal et al. 2021). Xia et al. (2019) address this task by employing a Word2Vec model and conducting a comparative analysis with a Bag of Words (BOW) model that utilizes TF-IDF (Term Frequency-Inverse Document Frequency).

Secondly, reasoning within legal use cases represents a significant challenge that has prompted numerous applications and research efforts within the scientific community (Satoh et al. 2020; Mumford et al. 2022). Logical formulas represent one of these attempts to formalize legal reasoning into structured, computable formats. By encoding legal rules and principles into logical expressions, this approach aims to enable automated reasoning systems to infer conclusions from given legal facts, thus simulating aspects of human legal reasoning. Ferraro (2020) develop an integrated

pipeline for the extraction of legal information and the subsequent generation of logical formulas. Al-Ageili and Mouhoub (2022) developed a decision-making system within the domain of residential land use suitability, which is based on an ontology.

Question answering (QA) within the legal domain focuses on developing systems that can understand and respond to natural language queries about legal matters. This involves parsing complex legal documents and providing concise, accurate answers to specific questions, thereby making legal information more accessible to both legal professionals and the public. Martinez-Gil (2023) published a comprehensive survey related to QA systems, while works such as that by Do et al. (2017) introduced a system that incorporates SVM (Support Vector Machine) and CNN (Convolutional Neural Network) technologies and emphasized the significance of input features in achieving effective outcomes.

Finally, and related to our study, Information Extraction (IE) with Entity Recognition and Extraction (ER/EE) pertains to the automated identification and categorization of terms or entities within legal texts, such as parties involved, relevant dates, and legal concepts. For example, the anonymization of German financial documents, which depends on the identification of entities (Biesner 2022). Early entities extraction methods relied on predefined patterns and expert-crafted rules to identify terms in text (Sleimi et al. 2018; Farmakiotou 2000; Wang et al. 2020). While rule-based systems demonstrated impressive precision and recall within their designated datasets, they also posed challenges as they demand considerable human effort to encompass linguistic nuances, necessitate extensive technical expertise along with domain knowledge, and struggle with adapting to even slight modifications in the underlying data, resulting in an unwieldy proliferation of rules (Waltl et al. 2018).

Statistical models like HMMs (Hidden Markov Model) Eddy (1996) and CRFs (Conditional Random Fields) Lafferty (2001) utilize probabilistic models to capture the sequential dependencies between words and label sequences, offering more flexibility and generalization compared to rule-based methods (Zhang 2004; Alex et al. 2007). For example, Santosh et al. (2023) devised a system aimed at semantically annotating coherent text segments and assigning appropriate labels to them. However, they may require substantial feature engineering and labelled data for training, limiting their scalability and adaptability to new domains.

With the advent of deep learning techniques, particularly neural network architectures like recurrent neural networks (RNNs) Sherstinsky (2020), convolutional neural networks (CNNs) O'Shea (2015), and more recently, transformer-based models such as BERT and the attention mechanism, significant advancements have been made in terms extraction (Liu 2017; Korvigo et al. 2018; Shen 2022; Yan et al. 2019; Friedman et al. 2022). These models can effectively learn representations of words and their contextual information, enabling them to accurately identify and classify terms in a wide range of domains and languages. Nevertheless, they often require large amounts of annotated data and computational resources for training and inference. Transfer learning, particularly using pre-trained language models for Named Entity Recognition (NER), has emerged as a promising approach in entity extraction (Zin et al. 2023).

By leveraging pre-trained models like BERT, RoBERTa, BART, or T5, which have been trained on vast amounts of text data, researchers can better tackle these

tasks. However, there are still challenges and limitations to consider when applying transfer learning for terms extraction using pre-trained language models. While these models have been trained on vast amounts of text data, they may not capture the specific domain or language nuances of the target terms extraction task. Fine-tuning on domain-specific datasets can help, but it may still require a considerable amount of annotated data for effective adaptation. Acquiring high-quality annotated datasets can be expensive and time-consuming, especially for niche domains or languages with limited resources. Additionally, ensuring consistency and accuracy in annotations is essential for the effectiveness of the model. Moreover, fine-tuning large pre-trained models can be computationally intensive, requiring powerful hardware and significant memory resources. Training may take a long time, especially when dealing with large datasets or complex models.

Integrating Large Language Models (LLMs) into the term extraction presents a novel avenue for enhancing the performance. These models, built on the foundation of extensive pre-training over diverse and large datasets, offer a significant leap in understanding and processing natural language. LLMs are, by design, adept at grasping the subtleties of language and domain-specific nuances, potentially reducing the gap identified in earlier approaches that rely on standard pre-trained models. Several works have already reported advances through the use of LLMs in various domains (Savelka 2023; Blair-Stanek et al. 2023; Yang et al. 2020), even in medical annotation. Goel (2023) leveraged an LLM to conduct terms extraction tasks, specifically targeting terms such as dose, duration, frequency, and drug names; achieving an F1 score of approximately 0.8. This study not only underscores the LLM's capacity for high performance in extracting terms with significant accuracy but also highlights its utility in streamlining human-involved processes. By incorporating an LLM into a human-annotated workflow, the study demonstrated a remarkable reduction in annotation time, averaging a 58% decrease. This efficiency gain suggests that integrating LLMs can significantly enhance the productivity of human annotators while maintaining, or even improving, the quality of the extracted terms.

3 Methodology

This section details the methodology implemented to explore the research questions introduced earlier. The main objective of our study is to examine the capability of Large Language Models (LLMs) for legal terms extraction. As depicted in Fig. 2, utilizing LLMs necessitates legal data as input. Specifically, these data consist of raw sentences extracted, in our study, from the Luxembourg Traffic Law. Employing various techniques, the LLM is capable of identifying predefined legal terms. The concluding phase involves post-processing the outputs generated by the LLM to obtain data in JSON format.

To explain this whole process in depth, the following subsections will delve into detail. First, with the dataset, we will describe the semantic concepts used from the Sleimi et al. dataset (Sleimi et al. 2018). We will discuss the source of these data and their suitability for zero-shot and one-shot learning. Then, we will discuss the models utilized in this study: GPT-4, Miqui-1-70b, Mixtral-8x7b, and Mistral-7b. This

Table 1 Concept definitions from Sleimi et al. (2018) for the eight concepts used in our study

Concept	Definition
Action	The process of doing something
Actor	An entity that has the capability to act
Artifact	A human-made object involved in an action
Condition	A constraint stating the properties that must be met
Location	A place where an action is performed
Modality	A verb indicating the modality of the action (e.g may, must, shall)
Reference	A mention of other legal provision(s) or legal text(s) affecting the current provision
Time	The moment or duration associated with the occurrence of an action

choice is justified by comparing them based on the following criteria: the weights' status (closed weights versus open weights), the number of parameters, the required hardware, inference time, and training time. These criteria have been selected to transcend not only the scientific outcomes, but also to emphasize the potential for industrial application.

We then examine two strategies for performing legal terms extraction: Prompt Engineering and Fine-Tuning, giving all the details about our process. We will describe the post-precessing requirements in order to work with the LLMs output. The whole code is accessible in a public GitLab repository.[1]

3.1 The legal terms dataset

The dataset we use in our work has been introduced in 2018 by Sleimi et al. (2018). This study accomplishes the extraction of legal terms by employing rule-based methodologies. The authors report that their rule-based system is capable of achieving a precision score of 0.874 and a recall score of 0.855. However, achieving this level of performance necessitates significant time investment from experts in the annotation process and rule based pattern creation. Furthermore, in a subsequent paper (Sleimi 2021), they addressed the issue of limited generalizability by applying their approach to various legal codes, including the Code of Commerce, the Penal Code, the Code for Healthcare, the Labour Code, and the Code for the Environment. This application resulted in a significant reduction in precision, which declined by around 14.8%, indicating a significant decrease from the original domain to new domains.

In their paper (Sleimi et al. 2018), experts annotated 200 French selected statements from the Luxembourg Traffic Law and identified 1339 phrases. They focus on 14 legal concepts and publish the dataset[2]. However, in our article, we worked on a subpart of these concepts by using only 8 of them: Action, Actor, Object,

[1] *** Hidden during the double-blind review *** https://gitlab.irit.fr/ala/legal-concepts-extraction.

[2] https://sites.google.com/view/metax-re2018/.

Condition, Location, Modality, Reference and Time. The eight selected concepts are detailed in Table 1. This selection was guided by our observation that the other concepts were either underrepresented in the dataset, leading to a major flaw in the experiment, or did not align with our interpretation of their definitions. We prioritized high-level concepts that are applicable across various legal domains and document types, as this approach addresses the objective of been used in multiple legal subdomain and not only the traffic law.

Figure 3 details the distribution of the concepts. Given the low annotation counts for each concept in the training dataset (fewer than 250 per concept), we characterize this task as operating within a context of limited annotated data. Our methodology emphasizes this constraint further by not incorporating supplementary information from experts. In contrast to rule-based approaches, our method does not rely on predefined trigger words or syntactic patterns.

A distinctive feature of this dataset and the associated tasks is the occurrence of term overlap. Contrary to other research in legal terms recognition, such as the work by Leitner et al. (2019), our task is incompatible with the IOB2 (Inside-outside-beginning) format. This is because parts of a sentence may be annotated with two different concepts. For instance, in the Fig. 1, the segment "equipped with an odometer" could be annotated as a condition, while "vehicles equipped with an odometer" could be annotated as an artifact. In this example, "equipped with an odometer" is implicated in two distinct concepts, illustrating annotation overlapping. Therefore, this task is multi-class and multi-label.

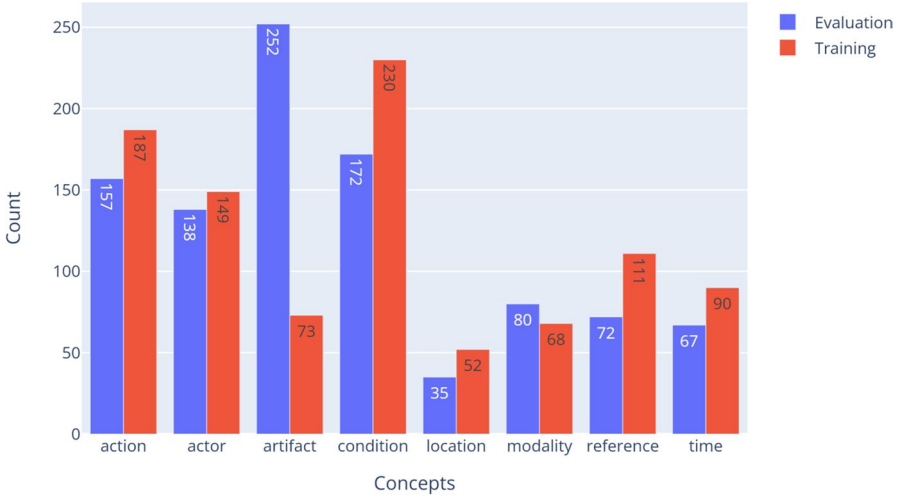

Fig. 3 Concepts distribution from the training and evaluation datasets (Sleimi et al. 2018). Our article employs eight legal concepts: action, actor, artifact, condition, location, modality, reference, and time

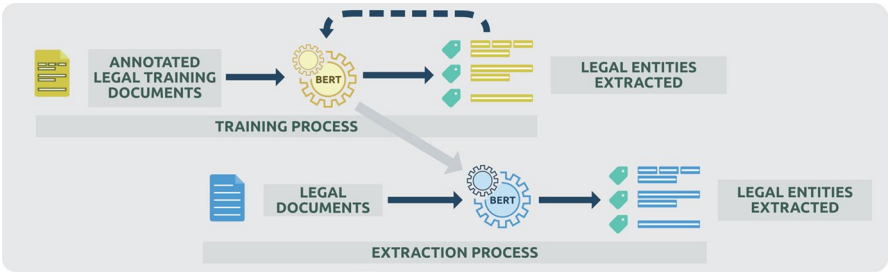

Fig. 4 Overall Legal-BERTerm process by (i) fine-tuning the model on the training dataset and (ii) using this fine-tuned model to perform the evaluation

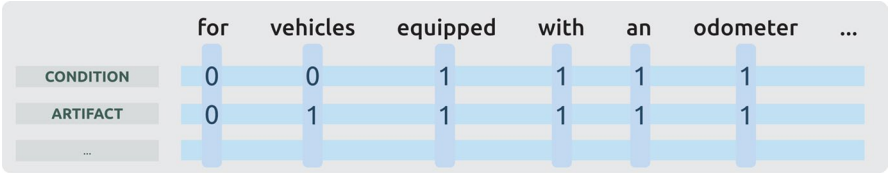

Fig. 5 Tokenizer matrix based on the short sentence: "for vehicules equipped with an odometer". Words activate legal concepts like Condition or Artifact with binary values (0 or 1)

3.2 Fine-tuned CamemBERT

To evaluate our approach with state-of-the-art systems, we introduce *legal-BERTerm*, a BERT model for the legal terms extraction task. This section provides a description of the implementation, highlighting the advantages and disadvantages of the architecture.

Due to the French nature of the dataset (detailed in Sect. 3.1), we employed a CamemBERT model, instead of the traditional BERT. We used Legal-CamemBERT-base (Louis et al. 2023), which has been fine-tuned on over 22,000 legal articles from Belgian legislation in French, making it better suited for processing and understanding French legal texts. This choice was driven by accurately capturing the nuances and intricacies of the French language present in our dataset, thereby enhancing overall performance and reliability.

Legal-BERTerm relies on expert-produced data corpora to function effectively. The dataset should be divided into two parts: one for training and the other one for the evaluation, as illustrated in Fig. 4. Traditional terms extraction with BERT relies on Inside-Outside-Beginning tagging, a common method in natural language processing for labelling sequences, particularly in tasks such as NER. However, in the legal terms extraction task, annotation overlaps exist in the document, illustrated in the example Fig. 1. The task is formally defined as a multi-label, multi-class classification problem, therefore, it necessitates to modify the internal architecture of the CamemBERT model.

The first modification pertains to the input matrix, used inside the PyTorch model, to allow overlapping classification. Figure 5 illustrate this modification,

Table 2 Weights' status, number of parameters and GPU used for the LLMs: GPT-4, Miqu-1-70b, Mixtral-8x7b and Mistral-7b

Model	Weights' status	Number of parameters	GPU used
GPT-4	Closed-weights	N/A	N/A
Miqu-1-70b	Open-weights	70B	A100 (80Go)
Mixtral-8x7b	Open-weights	12.9B used (46.7B total)	A100 (80Go)
Mistral-7b	Open-weights	7B	V100 (32Go)

by allowing multiple tokens in a sentence to be activated simultaneously, such as the word "odometer". The second modification involves the custom metric used to reward the model during the training process. We compute a macro F1 score, which is the average F1 score across all concepts. The final modification involves creating a custom Model Class using the Transformer library from Huggingface. This modification allows us to override the loss function, switching from CrossEntropyLoss[3] to BCEWithLogitsLoss,[4] which is a binary cross-entropy combined with a sigmoid function suitable for multi-label and multi-classes use cases. This custom architecture, along with the other experiments in this article, is available in our GitLab repository.[5]

As discussed, utilizing Legal-BERTerm model necessitates a sufficient amount of training data, 200 statements in this dataset (Sleimi et al. 2018). This architecture can reach great performance (detailed in Sect. 4.2.1), but must deal with data produced by experts through time-consuming processes. However, this method represents the up-to-date state-of-the-art and the recent advent of generative AI models has introduced new research avenues in zero-shot and few-shot learning. These models can perform legal terms extraction without the need for extensive data and expert intervention. This approach is examined in detail in the following section.

3.3 Large language models

To extract legal terms, we selected four LLMs with distinct architectures: GPT-4, Miqu-1-70b, Mixtral-8x7b, and Mistral-7b. Table 2 provides the Weights' status, which is the ability to users to use the model in local and to train it. We also give the number of parameters, which represent the capacity of LLMs to learn features. Finally, the GPU we used for inference and fine-tune of these models.

Since the advent of LLMs, GPT (from OpenAI[6]) models have emerged as a benchmark for information extraction tasks. However, it is crucial to acknowledge and compare their closed-weights model with open-weights ones. This is why we

[3] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.
[4] https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html.
[5] https://gitlab.irit.fr/ala/legal-concepts-extraction.
[6] https://openai.com/.

selected three open-weight models from the Mistral company[7]: Miqu-1-70b, Mixtral-8x7b and Mistral-7b. In our article, we also want to evaluate the impact of LLM size by comparing performance in relation to the number of parameters.

In addition to model performance, LLM application in industry frequently introduces new constraints, including time and cost considerations. Consequently, a smaller model, characterized by faster inference and fine-tuning times and requiring less powerful GPUs, would be more desirable. This perspective will be further elaborated and discussed in the evaluation section.

3.4 Extraction strategies

Utilizing LLMs presents primarily two alternatives for tailoring their behaviour to a specific task. Our article explores both strategies to evaluate their comparative performance. Prompt Engineering and Fine-Tuning represent two distinct approaches that modify the LLMs output. Subsequent sections will delineate the inner workings of these methods with the entire pipeline, from input data processing to generating usable outputs.

3.4.1 Prompt engineering pipeline

Prompt engineering is a subfield within artificial intelligence and human-computer interaction (HCI) focused on the design, analysis, and optimization of prompts, structured inputs or queries, to elicit desired responses or behaviours from AI systems, particularly those based on LLMs. This domain encompasses a range of activities, including the formulation of prompts to guide AI in legal terms extraction, thereby enhancing its performance.

Unlike conventional methods, prompt engineering does not modify the underlying architecture or weights of an AI model. Instead, it leverages the pre-existing knowledge and capabilities encoded within the model during its initial training phase, which typically involves exposure to vast datasets spanning diverse domains. By crafting prompts, the input queries given to the model can guide the AI to generate outputs that align more closely with specific user intentions or requirements.

Prompt engineering significantly lowers the barrier to accessing advanced AI capabilities, enabling users without extensive technical expertise in AI model development or the resources for comprehensive model training, like hardware or dataset, to utilize these technologies. Furthermore, it facilitates swift experimentation and iteration, allowing for changes to be tested and implemented in real time without impacting the model's fundamental functionality. Additionally, it creates opportunities for personalized AI interactions, as prompts can be customized for specific tasks, such as legal terms extraction.

Table 3 illustrates the structure of the pre-prompt: initially, a role is assigned to the LLM: "NLP expert", along with the task: "extracting terms from sentences".

[7] https://mistral.ai/.

Table 3 Zero-shot pre-prompt used before the target sentence to perform legal terms extraction. This pre-prompt has been translated into English for the purposes of this article

You're an NLP expert specializing in extracting terms from sentences. These terms include Action, Actor, Artifect, Condition, Location, Modality, Reference and Time.
These concepts take on the following definitions:
* Action: the act of doing something
* Actor: an entity with the capacity to act.
* Artifact: a man-made physical element involved in an action
* Condition: a constraint stating the properties that must be respected.
* Place: the place where an action takes place
*Modality: a verb indicating the modality of the action (e.g. may,must, etc.)
* Reference: mention of other legal provisions or texts affecting the current provision
* Time: the moment or duration associated with the performance of an action.
An element of the initial sentence can have several classifications, so you need to extract all possible classifications.
When analyzing texts, your answers should be formatted as JSONs, listing the identified concepts without elaboration or justification.
The JSON should take the following form:"Action": [],"Actor": [],"Object": [],"Condition": [],"Place": [],"Modality": [],"Reference": [],"Time": []
Only JSON should be used as the answer; no justification or explanation is accepted.
What's more, you must not reformulate the extracted elements - they must be identical to the word.

Indeed, the guidance provided in the OpenAI documentation[8] suggests that clearly defining the role and capabilities of a model, can significantly enhance the quality and relevance of the outputs generated. The second part entails a description of the eight concepts and definitions. This description represents the only necessary introduction of external knowledge, necessitating the involvement of an expert. Lastly, details regarding the output are provided, specifying that it should be in JSON format without further explanation and adhering to a non-rephrase constraint (the output text should not be altered or reworded in any way).

The adoption of this specific prompt structure emerged from an empirical process of iterative experimentation, aimed at pinpointing the configuration that consistently delivers superior outcomes. This approach was informed by a close adherence to best practices and guidelines recommended by OpenAI. Through successive trials, adjustments were made to refine the prompt's components, including the precise definition of the model's role, the task description, and the format of the expected output. This empirical methodology ensured the identification of a prompt architecture that meets the objectives and also aligns with the capabilities of the LLM.

The final prompt concatenates the pre-prompt with the sentence to process. This composite sentence is then provided to the large language model for processing. As noted in the dataset description, our sentences are derived from legal documents from Luxembourg, focusing on traffic law. Here is an example of an added sentence,

[8] https://platform.openai.com/docs/guides/prompt-engineering.

Table 4 One-shot pre-prompt addition used before the target sentence to perform legal terms extraction. This pre-prompt has been translated into English for the purposes of this article

For example, with the following sentence: "The owner or keeper of a road vehicle who considers a decision concerning the type-approval or registration of his vehicle to be ill-founded may refer the decision to the Minister, who, after requesting the SNCA's position, will confirm or amend it within two months of the appeal being lodged, accompanied by all the necessary documents and information.".
You must obtain: {"action": ["refer the decision to the Minister"], "actor": ["The owner or keeper of a road vehicle", "Minister"], "condition": ["accompanied by all the necessary documents and information", "who, after requesting the SNCA's position, will confirm or amend it within two months of the appeal being lodged, accompanied by all the necessary documents and information", "who considers a decision concerning the type-approval or registration of his vehicle", "concerning the type-approval or registration of his vehicle"], "modality": ["may"], "time": ["after requesting the SNCA's position", "within two months of the appeal being lodged"]}

after the pre-prompt, in order to extract terms: "When one or more major or critical defects or non-conformities are found on a Luxembourg-registered vehicle, the roadworthiness inspector may decide that the vehicle must undergo a full roadworthiness inspection within a given timeframe". This methodology is called "zero-shot" because it does not provide any examples to the LLM.

Conversely, in the one-shot approach, an example along with the desired output is included at the end of the pre-prompt, offering a direct illustration of the task to be performed by the model. You can find the pre-prompt's addition in the Table 4. With the objective to minimize the expert involvement and follow the limited annotated data constraint, the one-shot prompt only include one example, annotated by the expert. Section 5 will explore the benefits of adding more examples.

To execute the four models, we employ two NVIDIA GPUs during the inference stage: the A100 (80 GB) and the V100 (32 GB). To accommodate the models within the GPU memory constraints, we employ a 4-bit precision quantized model. The script files and the hyperparameter configurations are publicly available and can be accessed through a GitLab repository.[9] Specifically, we adjusted the temperature parameter to 0.5 in the configuration of the Large Language Model (LLM). This setting was chosen to strike a balance between coherence and creativity. Although a temperature setting of 0 is generally recommended for tasks requiring maximum determinism, this configuration led to degraded performance in our experiments. The model appeared overly constrained, limiting its ability to extract legal terms. As a result, we determined that allowing a certain degree of creativity was necessary, which informed our decision to use the current temperature setting of 0.5. This value aims to achieve balance, offering enough flexibility without limiting hallucination in the model's outputs.

To finalize this pipeline, a post-processing is employed to refine the LLM's output, ensuring it is computationally usable. Given that the LLM is not inherently trained for JSON output, instances of poorly formatted outputs may occur. To rectify

[9] *** Hidden during the double-blind review *** https://gitlab.irit.fr/ala/legal-concepts-extraction.

this issue, any content that does not conform to the JSON format is removed, and attempts are made to serialize the remaining data, thus enhancing its utility for computational applications.

In the next section, we will introduce an additional step following prompt engineering to enhance the potential outcomes. This technique, known as fine-tuning, is designed to complement the prompt-engineering approach by training the LLMs on a training dataset.

3.4.2 Fine-tuning pipeline

Fine-tuning refers to the process of adjusting a pre-trained model to better suit a specific task or dataset. This approach leverages a model that has already been trained on a large, general dataset, enabling it to learn general features, representations, or patterns. The fine-tuning stage involves continued training of the model on a smaller, task-specific dataset. This process allows the model to adapt its previously learned features to the nuances and specific characteristics of the target task or domain, thereby improving its performance on that task compared to training from scratch or using the pre-trained model without adjustment.

Sleimi et al. (2018) supplied both a training dataset and an evaluation dataset, as described in Fig. 3. Whereas the zero-shot approach does not utilize a training dataset and the one-shot approach employs only a single example, fine-tuning leverages the entire training dataset to refine the output. In practice, the fine-tuning approach using the zero-shot prompt mechanism with a training sentence. If the response is incorrect, the correct JSON is provided to the model as feedback. This process allow the LLM to learn and improve its ability to generate accurate responses based on the training data provided.

Every open-weight model, introduced before, has been fine-tuned on the training dataset with Hu (2021) (Low-Rank Adaptation) configuration. Low-Rank Adaptation provides a computationally efficient method for the fine-tuning of LLMs through minimal modifications to their pre-trained weights. This technique facilitates targeted adjustments, enabling the tailoring of models to specific tasks or datasets without the need for extensive retraining. By altering only a limited subset of parameters, LoRa maintains the overarching knowledge embedded within the model. The fine-tuning process utilizing LoRa is markedly quicker and requires fewer resources compared to conventional methods, thereby accelerating the cycle of iteration and development. We also set up a maximum of 6 epochs with a learning rate of 1×10^{-4} and the adamw_torch optimizer. The entire implementation is supported by the Transformers library. The LLM architecture remains unchanged for the three models, while LoRa specifically targets the Q, K, V, and O modules. All the hyperparameters are available in the GitLab repository.[10]

Following this fine-tuning phase, the next steps closely mirror the prompt engineering pipeline. The evaluation dataset, incorporating the zero-shot prompt, is processed through the fine-tuned LLMs.

[10] *** Hidden during the double-blind review *** https://gitlab.irit.fr/ala/legal-concepts-extraction.

4 Evaluation

Following the inference phase of the LLMs and subsequent post-processing, the output undergoes evaluation. To this end, the forthcoming section delineates the evaluation methodology and provides an in-depth analysis of the results.

The initial section addresses the evaluation methodology, wherein we introduce the concepts of Precision, Recall, and F1 Score as metrics. We will demonstrate the limitations inherent to these metrics when applied to LLMs in the context of information extraction tasks. A particular focus will be placed on the challenge of boundary discrepancies between annotation and inference. To circumvent these limitations, we propose an enhanced evaluation framework that incorporates strategies such as the use of a detailed analysis of True Positive instances and the Levenshtein distance (Yujian and Bo 2007). This approach is designed to yield a more accurate representation of the model's performance.

The final subsection is devoted to discussing the outcomes derived from the models, interpreted through the lens of the evaluation framework. In this section, we will review and interpret the performance metrics. Additionally, we will undertake a comparative analysis between the LLMs and the rule-based approach, paying special attention to the extent and nature of expert involvement required by both methodologies. This comparison aims to shed light on the relative advantages and limitations of each approach in the context of our study.

4.1 Evaluation methodology

In the assessment of information extraction models' performance, a range of metrics is routinely utilized to evaluate their precision and reliability. Recall, precision, F1 score, and F2 score are common metrics in delineating a model's performance. These measures are especially valuable in contexts where achieving an equilibrium between false positives and false negatives is critical for the intended task.

Recall, also known as the true positive rate, is quantified by the formula $\text{Recall} = \frac{TP}{TP+FN}$, highlighting the model's proficiency in identifying all pertinent instances. Precision, on the other hand, gauges the accuracy of positive predictions and is computed as $\text{Precision} = \frac{TP}{TP+FP}$. The F1 score amalgamates precision and recall through the equation $\text{F1} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$, offering a balanced measure. Meanwhile, the F2 score, which gives greater weight to recall, is determined by $\text{F2} = \frac{5 \times \text{Precision} \times \text{Recall}}{4 \times \text{Precision} + \text{Recall}}$. Collectively, these metrics furnish a detailed framework for evaluating the compromises between false positives and false negatives, a process essential for making well-informed decisions across varied application landscapes.

4.1.1 Boundary issue

We introduce an evaluation framework to address certain limitations inherent in traditional metrics. A primary concern arises from discrepancies in boundary

annotation between experts and LLMs. Consider the sentence: “The very old blue car has to pass the technical inspection". Here, an expert may annotate “very old blue car" as an artifact, whereas LLMs might only identify “blue car" as the relevant segment. Under conventional evaluation methodologies, the LLM’s annotation would be dismissed as a False Positive. However, we contend that such an annotation could still represent a valid response (i.e. True Positive).

To accommodate this perspective, we have refined the True Positive category into two distinct subcategories: a) the *perfect match*, the expert’s annotation is equal to the LLM’s annotation; b) the *partial match*, which acknowledges instances where the LLM correctly identifies the overarching concept (in this instance, artifact) and its annotation is a subset of, or is encompassed by, the expert’s annotation. In the results section, we will delineate the “true positive" category into two nuanced classifications: “perfect match" and “partial match." This distinction allows for a more granular analysis of the model’s performance by recognizing not only the instances of exact match between the model’s annotations and those of human experts, but also instances where the model’s annotations accurately capture a subset of the expert-identified concept. Table 5 presents examples to summarize our evaluation methodology.

To assess the “partial march" category, we will introduce an additional metric: the normalized Levenshtein distance. The Levenshtein distance, or edit distance, serves as a metric to gauge the similarity between two strings, by calculating the minimal number of single-character alterations needed to transform one string into the other. The goal is to determine the difference between the LLM’s annotation and the expert’s annotation. We will normalize this metric in order to compare it across all the partial match annotations and all the models.

This metric is particularly relevant in the legal domain, where precision is paramount, and even minor discrepancies can have significant implications. Legal texts are characterized by complex, lengthy sentences that often incorporate multiple clauses to convey layered, interconnected ideas. Unlike general texts, where sentences average around 20 words, legal documents like the French Labor Code can reach an average of 70 words per sentence. This extended length is not incidental; it reflects a deliberate structuring that allows for precise definitions, conditions, and exceptions to be embedded within a single sentence. Through this chaining of ideas, legal language achieves a high degree of specificity, often outlining multiple contingencies and nuanced interpretations within a single clause. However, this structure also means that even minor changes in wording or punctuation can drastically alter

Table 5 Examples for the evaluation methodology including: Perfect Match, Partial Match (with the Normalized Levenshtein Distance: NLD) and False Positive

Expert annotation	LLM annotation	Result
Very old blue car	Very old blue car	Perfect Match (True Positive)
Very old blue car	Blue car	Partial Match (True Positive) (NLD = 0.529)
Blue car	Very old blue car	Partial Match (True Positive) (NLD = 0.529)
Blue car	Red bus	False Positive

a sentence's meaning, highlighting the critical need for absolute accuracy in legal contexts. For instance, the insertion or omission of a comma can impact the interpretation of a legal clause, potentially changing rights, obligations, or interpretations of liability. By using the normalized Levenshtein distance, we can capture and quantify such subtle variations between annotations, which is essential for evaluating that automated models achieve the high level of accuracy required in legal contexts.

4.2 Results

4.2.1 Overall performance

Table 6 presents an overview of the models: GPT-4, Miqu-1-70b, Mixtral-8x7b, and Mistral-7b; detailing their configurations across different strategies, including prompt style and fine-tuning. Notably, GPT-4, configured with a one-shot prompt style and without fine-tuning, demonstrated superior performance, achieving an F1 score of approximately 0.690.

Figure 6 provides a comparison of the performance of the four models under their optimal F1 configurations. Within the evaluation dataset, which comprises 973 legal terms, GPT-4 accurately extracted 270 terms with an exact match. Furthermore, approximately 400 additional terms were recognized as "partial matches" of the annotations provided by experts. GPT-4 demonstrates superior accuracy, yielding the lowest number of errors in terms of both False Positives and False Negatives. It is observed that a smaller model size tends to be associated with an increased

Table 6 Precision, Recall, F1 and F2 results for legal terms extraction using four models: GPT-4, Miqu-1-70b, Mixtral-8x7b and Mistral-7b. GPT-4 achieves optimal performance across all evaluated metrics with the following configuration: one-shot prompt style and no fine-tuning

	Strategy		Result			
Model	Prompt style	Fine-tuning	Precision	Recall	F1	F2
Rule-based Sleimi et al. (2018)			0.972	0.958	0.965	0.961
CamemBERT			0.819	0.543	0.633	0.574
GPT-4	Zero-Shot	No	0.645	0.684	0.664	0.676
	One-Shot	No	**0.677**	**0.704**	**0.690**	**0.698**
Miqu-1-70b	Zero-Shot	No	0.567	0.637	0.600	0.622
	One-Shot	No	0.590	0.563	0.576	0.568
	Zero-Shot	Yes	0.573	0.628	0.600	0.616
Mixtral-8x7b	Zero-Shot	No	0.526	0.526	0.526	0.526
	One-Shot	No	0.629	0.353	0.453	0.387
	Zero-Shot	Yes	0.628	0.435	0.514	0.463
Mistral-7b	Zero-Shot	No	0.405	0.392	0.398	0.394
	One-Shot	No	0.539	0.098	0.165	0.117
	Zero-Shot	Yes	0.474	0.424	0.448	0.433

Values in bold correspond to best LLM performance

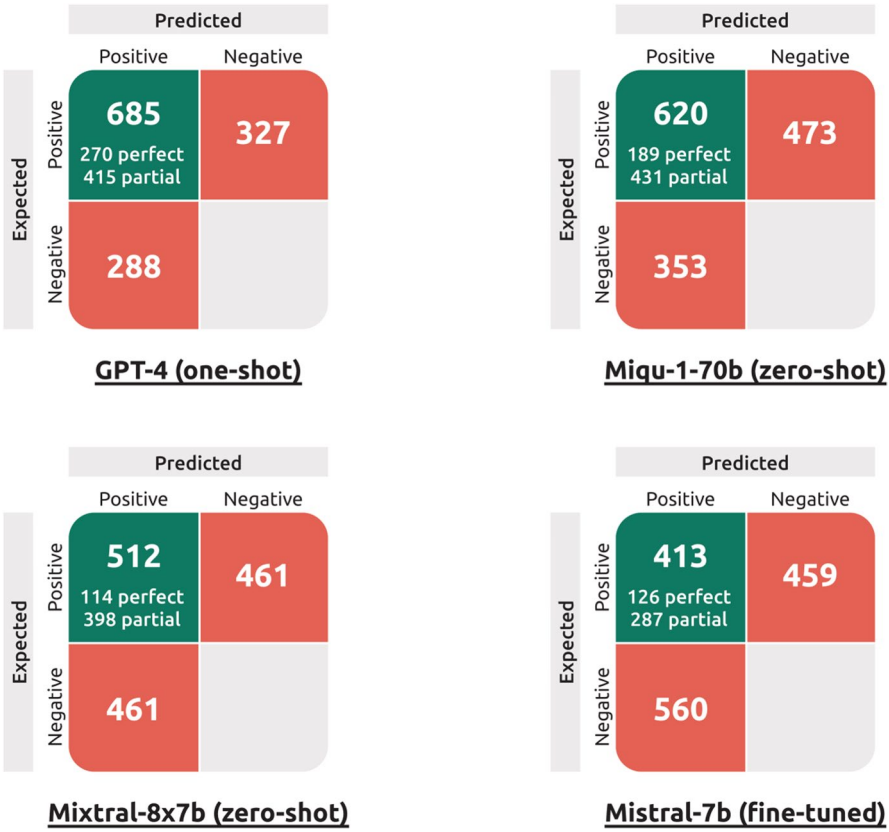

Fig. 6 Confusion matrixes with the best F1 configuration of our models. Sum of legal terms related to the Perfect Match, Partial Match, False Positive and False Negative categories

likelihood of generating false positives. In Sect. 4.2.2, we will conduct an in-depth analysis of the models' performance, focusing on fine-tuning time, inference time, and model size.

Figure 7 demonstrates a gradual increase in the F1 score as the allowance for partial matches in LLMs is varied. Specifically, in the absence of permitted partial matching, the performance of GPT-4 initiates at a level below 0.5. However, with an allowance for up to 60% variation in partial matches, GPT-4 achieves an F1 score of approximately 0.6. For reference, the rule-based authors reported an average Jaccard index about 0.46, which gives a very similar performance.

Figure 7 complements the insights provided by Table 6, highlighting that model performance is not binary and is significantly influenced by the choice of threshold applied to variations in the Levenshtein distance. As the threshold approaches 1, there is an increased risk of incorporating errors, underscoring the delicate balance required in setting this parameter to optimize performance without compromising accuracy.

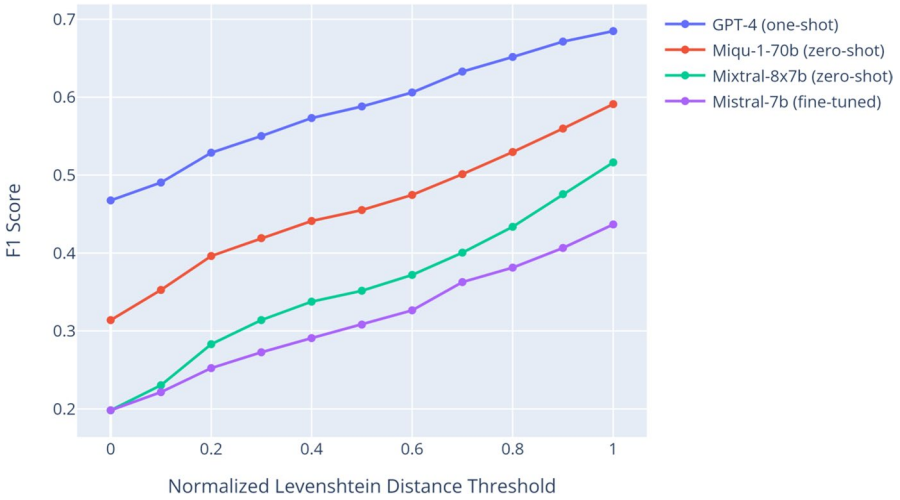

Fig. 7 Impact of Normalized Levenshtein Distance Threshold (NLDT) on F1 score. An NLDT of O allows no variation between the annotations made by experts and the partial matches from LLMs. By permitting a sentence variation of approximately 60%, GPT-4 can achieve an F1 score of around 0.6

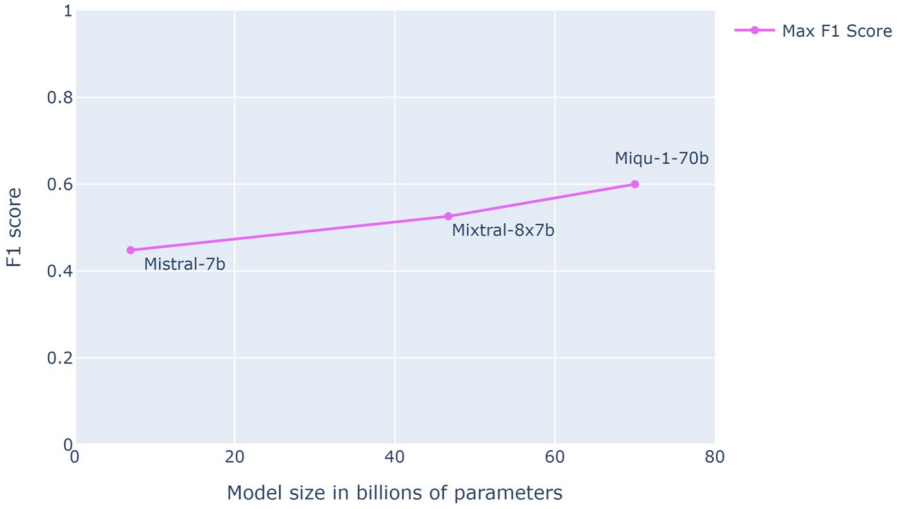

Fig. 8 Max F1 score per model on legal terms extraction related to the size in billions of parameters. From left to right, the models are: Mistral-7b (7B), Mixtral-8x7b (46.7B), Miqu-1-70b (70B)

The similarity of the curves' directional coefficients suggests a consistent performance level. The difficulty for LLMs to accurately extract whole terms, in alignment with expert annotations, may stem from their generalist nature and reliance on the broad knowledge acquired during their original training, highlighting a lack of specialization in this specific task.

4.2.2 Correlations between time, size and performance

From an industrial perspective, it is crucial to balance performance against both inference/fine-tuning time and the size of LLMs. Figure 8 illustrates that the cost, measured in billions of parameters, associated with enhancing performance is substantial. For instance, increasing the model size from 7 billion to 70 billion parameters improves the F1 score by approximately 0.15, but this improvement demands significantly more computational power. Achieving an F1 score above 0.8, under this configuration, necessitates a considerably large model size or architecture revolution. Consequently, we contend that LLMs are particularly valuable in contexts with limited data, where they can complement traditional approaches within hybrid systems (Breton et al. 2024).

This observation is corroborated by the data presented in Table 7, which details the time required for inference and fine-tuning, where applicable. The table clearly shows that an increase in model size directly correlates with longer run times. Therefore, smaller models can be significantly advantageous in production environments where time constraints are critical. Ultimately, it is essential to strike an optimal balance between performance, size, and duration to meet specific needs effectively.

4.2.3 Analysis of fine-tuning and one-shot prompt performance

Following the fine-tuning process, all models demonstrated a notable enhancement in accuracy when compared to their performance in the zero-shot setting. For instance, the accuracy of the Mixtral-8x7b model increased from 0.526 in the zero-shot scenario to 0.628 post fine-tuning. Interestingly, the performance of the models when subjected to a one-shot prompt was found to be comparable, or better, to that

Table 7 Duration of our runs for inference and fine-tuning if required. The duration of our runs is provided for reference purposes only. Please note that these results may vary depending on your specific configuration

	Strategy		Result
Model	Prompt style	Fine-tuning	Duration
Rule-based (Sleimi et al. 2018)			*No metric provided*
GPT-4	Zero-Shot	No	0 h, 15 min and 11 s
	One-Shot	No	0 h, 13 min and 54 s
Miqu-1-70b	Zero-Shot	No	3 h, 2 min and 11 s
	One-Shot	No	3 h, 32 min and 43 s
	Zero-Shot	Yes	5 h, 13 min and 21 s
Mixtral-8x7b	Zero-Shot	No	1 h, 33 min and 47 s
	One-Shot	No	2 h, 14 min and 47 s
	Zero-Shot	Yes	2 h, 43 min and 26 s
Mistral-7b	Zero-Shot	No	0 h, 37 min and 4 s
	One-Shot	No	0 h, 28 min and 31 s
	Zero-Shot	Yes	1 h, 45 min and 7 s

observed after fine-tuning. Regrettably, the improvement in precision was achieved in detriment of recall.

Consistent with the aforementioned insights regarding model sizes, Mistral-7b is the only model that exhibits an improvement in its F1 score following the application of fine-tuning. Our analysis leads to the conclusion that fine-tuning with limited data proves beneficial primarily for smaller-sized large language models.

This analysis further supports our conclusion that beyond a certain scale, LLMs are more effective when they leverage their original training knowledge. In such scenarios, they serve as a foundational system, enabling traditional methods to assume a more prominent role once sufficient data is available. This approach underscores the value of integrating LLMs with traditional models to optimize performance in data-driven tasks.

The impact of employing a one-shot prompt style presents a stark contrast to the outcomes observed with fine-tuning. While fine-tuning appeared beneficial primarily for smaller models, introducing a one-shot example within the prompt conversely diminished performance. Specifically, incorporating a single example in the prompt reduced Mistral-7b's F1 score from 0.398 to 0.165. Conversely, GPT-4's performance improved, with its F1 score increasing from 0.664 to 0.690. This suggests that the effectiveness of including an example in the prompt is contingent upon the model's size, indicating that only models with a sufficient number of parameters can adeptly manage the inclusion of examples within prompts.

4.2.4 Comparative analysis of LLMs, rule-based system and legal-BERTerm

We aim to compare the performance of large language models with legal-BERTerm and Rule-Based systems.

First, legal-BERTerm performs with an average F1 score exceeding 0.63. The average precision is 0.81, indicating that the fine-tuning of Legal-CamemBERT-base enables it to extract legal terms with high precision. The average recall, on the other hand, is about 0.54, highlighting the model's difficulties in extracting all terms comprehensively. However, this technique and its results depend on datasets manually created by experts. The creation of such datasets is inherently time-consuming and requires substantial expertise.

Secondly, with the rule-based system, it is important to highlight that, in their study, the authors (Sleimi et al. 2018) reported an inability to obtain results for the "reference" concept. This was due to the absence of extraction rules for this particular concept, illustrating a significant advantage of utilizing LLMs. With LLMs, the involvement of an expert is primarily required for the initial creation of definitions used in prompt engineering. Conversely, in Rule-Based systems, an expert must annotate numerous examples and subsequently analyse and create syntactic patterns. This task is time-consuming. This distinction underscores the efficiency and scalability of LLMs in extracting a wide range of concepts without the need for extensive manual rule creation.

However, we have noticed discrepancies between expert annotations and false positives identified by LLMs. While some annotations by LLMs are accurate yet overlooked by experts; an example of this can be observed in Fig. 1, derived from

Table 8 In-depth comparison between Rule-Based (RB) system (Sleimi et al. 2018) and our LLM approach after second round of validation. Detailed performance about the height legal concepts including: Perfect Match (TP), Partial Match (TP), False Positive, False Negative, Precision, Recall, F1

		Perfect match		Partial match		False positive		False negative		Precision		Recall		F1	
Legal concept	Ground Truth	RB	LLM	RB	LLM	RB	LLM	RB	LLM	RB	LLM	RB	LLM	RB	LLM
Action	157	91	6	64	124	2	20	2	27	0,987	0,867	0,987	0,828	0,987	0,847
Actor	138	110	73	19	47	4	12	9	18	0,970	0,909	0,935	0,870	0,952	0,889
Artifact	252	182	58	56	92	3	23	14	102	0,988	0,867	0,944	0,595	0,966	0,706
Condition	172	148	50	18	52	13	14	6	70	0,927	0,879	0,965	0,593	0,946	0,708
Location	35	27	4	7	18	0	0	1	13	1,000	1,000	0,971	0,629	0,986	0,772
Modality	80	74	46	4	24	3	1	2	10	0,963	0,986	0,975	0,875	0,969	0,927
Reference	72	–	15	–	34	–	0	–	23	–	1,000	–	0,681	–	0,810
Time	67	56	18	7	24	0	0	4	25	1,000	1,000	0,940	0,627	0,969	0,771
Total	973	688	270	175	415	25	70	38	288	0,972	0,907	0,958	0,704	0,965	0,793

the ground truth evaluation dataset. In the sentence, “verify the mileage" is not annotated as an action by experts, but successfully annotated by LLMs. This discrepancy may stem from the fact that the primary action in the rule is “made available", when "verify the mileage" serving as the objective, rather than an action.

To solve this problem, an expert assesses the model’s false positive outputs. Due to the time-intensive nature of this task, all the results presented in the previous section have not undergone false positive evaluation. Nevertheless, we have opted for the most capable model (GPT-4 model with one-shot prompt), and we intend to provide updated results following this second round of validation. Among the 327 false positive annotations identified in the LLM, as verified by the expert’s dataset, 257 were subsequently reclassified as true positives, while 70 remained categorized as false positives.

This process enabled the GPT-4 model (one-shot) to achieve a precision of **0.907**, close to the precision of 0.972 achieved by the rule-based system (recalculated to consider only the eight utilized concepts). On the recall front, however, the rule-based approach maintained its superiority, with a recall of 0.958 compared to the 0.704 achieved by LLMs. Table 8 provides a breakdown of the results across the eight concepts, aiding in the identification of limitations. For instance, the concept “Artifact" appears to yield the poorest outcomes, which may indicate various issues such as: bad definition of the concept in the prompt, inconsistency in the terms to be extracted, or a lack of sufficient training data, among others.

Despite these achievements, the impressive performance of LLMs is not without its drawback. The concern arises in the context of "Partial Match" performance. While LLMs possess the capability to extract the terms, relying on them exclusively for extraction purposes could result in incomplete extractions. The rule-based strategy recorded only 175 instances of "partial matches" whereas LLMs encountered approximately 415 instances. This issue is inherently linked to the fundamental nature of Large Language Models, underscoring a significant open issue regarding the ability of LLMs to delineate information with precise boundaries. This highlights a critical area for future research and development in the field of natural language processing, specifically in enhancing the precision of LLMs in information extraction tasks.

5 Conclusion

In our article, we delved into the utilization of Large Language Models (LLMs) specifically for the task of legal term extraction, aiming to alleviate the significant demand for annotated data in legal text analysis. The necessity for automated legal processing cannot be overstated. Historically, regulatory frameworks have been documented across extensive collections of texts, requiring businesses to dedicate considerable human resources to interpret, monitor, and ensure adherence to these legal requirements. This process is not only resource-intensive but also prone to human error, given the complexity and volume of legal documents.

Moreover, the dynamic nature of legal regulations, with legislative bodies and political institutions regularly revising and updating laws, further complicates the

landscape for compliance. Such frequent changes demand continuous monitoring and analysis to understand their implications for business operations, a task that is both cumbersome and costly for companies.

Since the scientific community began addressing this subject, a multitude of approaches has been explored. The evolution of methodologies has ranged from initial rule-based extraction tools to advanced language models like BERT, each reliant on annotated data to some extent. In the case of rule-based systems, the process involves annotating a substantial number of examples and subsequently developing a system to trigger extraction based on these rules. This necessity for extensive annotated datasets is a common challenge, not only for rule-based approaches but also for deep learning methods such as Bi-LSTM or BERT models. These advanced techniques require the injection of large volumes of annotated data to effectively train the system, highlighting a persistent dependency on high-quality, annotated datasets across different stages of technological advancement in text extraction.

The adoption of LLMs in this context presents a transformative opportunity. By automating the extraction of legal terms and relevant information from a plethora of legal documents, LLMs can significantly reduce the reliance on manual labour and mitigate the risk of oversight. Furthermore, the capability of LLMs to learn from unannotated data can diminish the necessity for extensively annotated corpora, traditionally a major bottleneck in the deployment of machine learning solutions in legal tech.

Our article not only explored the potential of four LLMs: GPT-4, Miqu-1-70b, Mixtral-8x7b and Mistral-7b in streamlining legal terms extraction, but also has found the strategy that offers the best results. Through our investigation, we aimed to highlight how LLMs can be used in information extraction task and harness their full power without excessive involvement of experts.

We implemented two strategies on these models: a) prompt engineering and b) fine-tuning. Through prompt engineering, we adapted the input text to more closely align with our specific task. We explored two distinct methodologies within prompt engineering: the zero-shot and one-shot approaches. The zero-shot approach comprises only the task description, the concept definitions, and the desired output format. Conversely, the one-shot approach builds upon the zero-shot foundation by incorporating an example input along with its expected outcome. Our findings indicate that simpler and shorter prompts yield better results for smaller-sized LLMs, such as Mistral-7b. In contrast, larger LLMs like GPT-4 exhibit the capacity to handle more complex prompts, including those with embedded examples. On the other hand, our exploration of fine-tuning techniques revealed a contrasting trend: larger LLMs experienced a decrease in performance when their original knowledge base was modified, whereas smaller models like Mistral-7b showed an enhancement in their performance. This divergence underscores the nuanced relationship between model size and the model accuracy.

GPT-4 with one-shot prompt, delivered superior performance, achieving a precision of approximately 0.676, a recall of about 0.704, and an F1 score around 0.690. Notably, GPT-4 was able to identify new terms not previously annotated in the evaluation dataset by experts. This discovery prompted a second round of validation for the false positives. After this additional expert assessment, GPT-4's precision improved significantly to approximately 0.907, compared to 0.972 for the rule-based system. Its recall, while not quite matching the rule-based system's

0.958, was still substantial at 0.704. These results underscore the potential of LLMs to effectively perform terms extraction within the legal domain, demonstrating not only their precision and recall capabilities, but also their ability to uncover previously unidentified terms.

A significant limitation has been identified when utilizing LLMs for legal information extraction. Although these systems successfully extract and classify terms, there has been an observed increase in partial matches. Partial matches occur when the LLM extracts only a subpart of the intended annotation, rather than the entire term. In comparison to rule-based systems, where partial matches accounted for 175 elements, and 400 elements for the LLMs. This discrepancy highlights a trade-off inherent in employing more generalized systems capable of zero-shot extraction. The flexibility and broad applicability of LLMs may come at the expense of increased partial matches and potentially reduced boundary precision in terms extraction tasks.

In conclusion, LLMs exhibit considerable potential for conducting information extraction within the legal domain, even when operating with zero prior knowledge input. This raw performance of LLMs, independent of the end-user, incurs a significant cost, as the foundational model must initially be trained on an extensive dataset. The most effective approach identified in our study involves prompt engineering, incorporating examples directly within the prompts, particularly when applied to the largest LLM, GPT-4.

However, challenges remain regarding the precise extraction of terms with accurate boundaries, presenting an unresolved issue that will direct our future research endeavours. To address and potentially overcome these limitations, we plan to investigate the prospects of system hybridization, combining the strengths of LLMs with those of previous technologies, such as rule-based systems and deep learning methods.

An alternative exploration involves leveraging a LLM specifically tailored for legal tasks. It is important to acknowledge that this approach necessitates a larger dataset. Gesnouin (2024) conducted extensive fine-tuning of a Llama model using French legal documents, demonstrating the potential for specialized adaptations. Consequently, our future research will examine the extent to which augmenting the foundational model with additional data can significantly alter the outcomes. This investigation aims to assess whether targeted data enrichment can enhance the model's performance in legal information extraction, thereby offering insights into the effectiveness of specialized training on domain-specific LLM applications.

Finally, the one-shot prompt strategy will be further developed by assessing the impact of using prompts that include multiple examples. This extended approach aims to determine whether providing several illustrative cases within a single prompt can improve the model's accuracy and adaptability.

Funding Open access funding provided by Université de Toulouse.

References

Al-Ageili M, Mouhoub M (2022) An ontology-based information extraction system for residential land-use suitability analysis. Int J Software Eng Knowl Eng 32:1019–1042

Alex B, Haddow B, Grover C (2007) Recognising nested named entities in biomedical text. Biological, translational, and clinical language processing. 65–72

Biesner D et al (2022) Anonymization of German financial documents using neural network-based language models with contextual word representations. Int J Data Sci Analytics 13:1–11

Blair-Stanek A, Holzenberger N, Van Durme B (2023) Can gpt-3 perform statutory reasoning? arXiv:abs/2302.06100

Breton J, Billami MB, Chevalier[1] M, Trojahn[1] C (2024) Empowering camembert legal entity extraction with llm boostrapping. In: Knowledge Eng Knowledge Manage: 24th Int Conf, EKAW 2024, Amsterdam, The Netherlands, November 26–28, 2024, Proceed. 86

Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:abs/1810.04805

Do P-K, Nguyen H-T, Tran C-X, Nguyen M-T, Nguyen M-L (2017) Legal question answering using ranking svm and deep convolutional neural network. arXiv:abs/1703.05320

Eddy SR (1996) Hidden markov models. Curr Opin Struct Biol 6:361–365

Farmakiotou D et al. (2000) Rule-based named entity recognition for greek financial texts. In: Proceed Workshop Comput lexicography Multimed Dictionaries (COMLEX 2000). 75–78

Ferraro G et al. (2020) Automatic extraction of legal norms: Evaluation of natural language processing tools. In: New Frontiers in Artificial Intelligence: JSAI-isAI International Workshops. 64–81

Friedman S, Magnusson I, Sarathy V, Schmer-Galunder S (2022) From unstructured text to causal knowledge graphs: a transformer-based approach. arXiv:abs/2202.11768

Gesnouin J et al. (2024) Llamandement: large language models for summarization of french legislative proposals. arXiv:abs/2401.16182

Goel A et al. (2023) Llms accelerate annotation for medical information extraction. Mach Learn Health (ML4H) 82–100

Hu EJ et al. (2021) Lora: low-rank adaptation of large language models. arXiv:abs/2106.09685

Huang Z, Xu W, Yu K (2015) Bidirectional lstm-crf models for sequence tagging. arXiv:abs/1508.01991

Korvigo I, Holmatov M, Zaikovskii A, Skoblov M (2018) Putting hands to rest: efficient deep cnn-rnn architecture for chemical named entity recognition with no hand-crafted rules. J cheminformatics 10:1–10

Lafferty J, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. Departmental Papers (CIS)

Leitner E, Rehm G, Moreno-Schneider J (2019) Fine-grained named entity recognition in legal documents. In: Int Conf Semantic Syst. 272–287

Liu Z et al (2017) Entity recognition from clinical texts via recurrent neural network. BMC Med Inform Decis Mak 17:53–61

Louis A, van Dijck G, Spanakis G (2023) Finding the law: enhancing statutory article retrieval via graph neural networks. In: Proceed 17th Conf European Chapter Association Comput Linguistics. 2753–2768

Mandal A, Ghosh K, Ghosh S, Mandal S (2021) Unsupervised approaches for measuring textual similarity between legal court case reports. Artificial Intell Law 29:1–35

Martinez-Gil J (2023) A survey on legal question-answering systems. Comput Sci Rev 48:100552

Monarcha-Matlak A (2021) Automated decision-making in public administration. Procedia Comput Sci 192:2077–2084

Moodley K, Hernandez Serrano PV, van Dijck G, Dumontier M (2019) Similarity and relevance of court decisions: a computational study on cjeu cases. In: Legal Knowledge and Information Systems. 63–72

Mumford J, Atkinson K, Bench-Capon T (2022) Reasoning with legal cases: a hybrid adf-ml approach. In: Legal knowledge and information systems. 93–102
O'Shea K (2015) An introduction to convolutional neural networks. arXiv:abs/1511.08458
Santosh TS, Bock P, Grabmair M (2023) Joint span segmentation and rhetorical role labeling with data augmentation for legal documents. In: European Conf Inform Retrieval. 627–636
Sassier P, Lansoy D (2008) Ubu loi. Arthème Fayard, France
Satoh K, Baldoni M, Giordano L (2020) Reasoning about applicable law in private international law in logic programming. In: Legal Knowledge and Information Systems. 281–285
Savelka J (2023) Unlocking practical applications in legal domain: evaluation of gpt for zero-shot semantic annotation of legal texts. arXiv:abs/2305.04417
Shen Y et al. (2022) Parallel instance query network for named entity recognition. arXiv:abs/2203.10545
Sherstinsky A (2020) Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Phys D 404:132306
Sleimi A et al (2021) An automated framework for the extraction of semantic legal metadata from legal texts. Empir Softw Eng 26:1–50
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 Int Require Eng Conf (RE)
Vaswani A (2017) *et al.* Attention is all you need. Adv neural inform process syst
Waltl B, Bonczek G, Matthes F (2018) Rule-based information extraction: advantages, limitations, and perspectives. Jusletter IT (02 2018)
Wang Z, Wu Y, Lei P, Peng C (2020) Named entity recognition method of brazilian legal text based on pre-training model. J Phys: Conf Ser 1550:032149
Xia C, He T, Li W, Qin Z, Zou Z (2019) Similarity analysis of law documents based on word2vec. In: 2019 IEEE 19th Int Conf Software Quality, Reliability Security Companion (QRS-C). 354–357
Yan H, Deng B, Li X, Qiu X (2019) Tener: adapting transformer encoder for named entity recognition. arXiv:abs/1911.04474
Yang Y, Uy M, CS Huang, A (2020) Finbert: a pretrained language model for financial communications. arXiv:abs/2006.08097
Yujian L, Bo L (2007) A normalized levenshtein distance metric. IEEE Trans Pattern Anal Mach Intell 29:1091–1095
Zhang J, Shen D, Zhou G, Su J, Tan C-L (2004) Enhancing hmm-based biomedical named entity recognition by studying special phenomena. J Biomed Inform 37:411–422
Zin MM, Nguyen HT, Satoh K, Sugawara S, Nishino F (2023) Improving translation of case descriptions into logical fact formulas using legalcasener. In: Proceed Nineteenth Int Conf Artificial Intell Law. 462–466

Authors and Affiliations

Julien Breton[1,2] · Mokhtar Mokhtar Billami[2] · Max Chevalier[1] · Ha Thanh Nguyen[3] · Ken Satoh[3] · Cassia Trojahn[1] · May Myo Zin[3]

✉ Julien Breton
julien.breton@irit.fr

[1] Informatics Research Institute of Toulouse (IRIT), Toulouse, France

[2] Berger-Levrault, Toulouse, France

[3] National Institute of Informatics (NII), Tokyo, Japan