CLAUSEBENCH: Enhancing Software License Analysis with Clause-Level Benchmarking

Qiang Ke*‡, Xinyi Hou*‡, Yanjie Zhao†‡, and Haoyu Wang‡
‡Huazhong University of Science and Technology, Wuhan, China
qiangke@hust.edu.cn, xinyihou@hust.edu.cn, yanjie_zhao@hust.edu.cn, haoyuwang@hust.edu.cn

***Abstract*—Open-source software (OSS) has revolutionized modern software development by fostering collaboration across diverse teams. However, as OSS projects grow in size and complexity, managing license compliance has become increasingly challenging. A critical issue lies in accurately recognizing and interpreting the varied clauses within OSS licenses, particularly when multiple licenses coexist, each with distinct permissions, obligations, and restrictions. Traditional license analysis tools, often rule-based, struggle to identify nuanced conflicts between license clauses, leading to potential compliance risks. In response to these challenges, this paper presents a fine-grained, high-quality dataset of 634 SPDX-certified licenses, annotated with 3,396 individual clauses across 14 categories. Each clause has been meticulously reviewed and validated using model-assisted checks to ensure accuracy, providing a solid foundation for detailed clause-level analysis. To improve clause recognition and conflict detection, we introduce CLAUSEBENCH, a benchmarking framework that leverages large language models (LLMs) to detect and interpret license clauses with high precision. CLAUSEBENCH improves detection accuracy by 50% compared to traditional document-level methods and significantly reduces hallucination rates by focusing on individual clauses, where precise distinctions in legal language are crucial. We also implemented a contextual prompt engineering strategy to optimize model performance, achieving 90% accuracy in clause identification. Our work sets a new standard for automated license conflict detection in OSS, demonstrating the potential of LLMs to manage the complexities of legal text interpretation. This work advances the license analysis field and opens the door to future research on integrating LLMs with OSS compliance tools.**

I. INTRODUCTION

As open-source software (OSS) continues to evolve rapidly, open-source licenses play a vital role in establishing the terms for software use, modification, and redistribution. These licenses provide a legal foundation for collaborative development, ensuring that the rights of developers and users are protected. However, the diversity of license terms introduces significant challenges, particularly in projects that use multiple licenses, as differences between license clauses often lead to conflicts. For instance, licenses like the GNU General Public License (GPL) [9] enforce strict copyleft requirements, while permissive licenses like the MIT License [25] allow for more flexible use. When these licenses coexist within a project, conflicts may arise, creating legal and compliance risks. Research has shown that such conflicts are prevalent in OSS projects. Cui et al. [5] found that 27.2% of OSS projects encounter license conflicts. Similarly, a large-scale empirical study by Wu et al.[41] across five major package management platforms confirmed that irregularities and incompatibilities in license usage are common. These conflicts pose significant risks for organizations, as unresolved violations can obstruct software distribution and complicate development processes [30].

While several tools, such as FOSSology [11] and SPDX-based compatibility checkers [17], provide basic functionality for detecting license conflicts, they often rely on predefined rule sets. These approaches struggle to cope with complex and evolving license terms, particularly when multiple licenses with intricate clauses are involved. Furthermore, different interpretations of license terms by developers can exacerbate the difficulty of accurately identifying conflicts [2]. As a result, there is a growing need for more flexible, fine-grained analysis tools that can handle the complexity of individual license clauses and provide more accurate conflict detection.

Given these limitations, recent advances in large language models (LLMs) offer a promising solution. ALBERT [20], an optimized and efficient variant of Bidirectional Encoder Representations from Transformers (BERT), has demonstrated exceptional capabilities in understanding complex natural language, making it well-suited for the semantic analysis of legal texts. Unlike traditional rule-based systems, **LLMs can adapt to different textual contexts and infer meanings from varied linguistic structures, which is particularly advantageous in the case of open-source licenses with their intricate and diverse clause formulations**. For example, LLMs have shown promise in handling complex sentence structures and understanding long-tail license clauses that are less common but critical for compliance [14]. Techniques like Focal Loss [24] further enhance the robustness of these models in detecting clauses that may be underrepresented in training data, improving their performance in recognizing infrequent but legally significant clauses.

Despite these advances, there remains a gap in fine-grained, clause-level benchmarking for license analysis. To address this, we introduced CLAUSEBENCH, a comprehensive clause-level framework. As shown in Figure 2, we collected and merged license data from sources like SPDX, followed by fine-grained annotation of over 600 open-source licenses into 14 distinct clause categories. Each clause was carefully reviewed using both manual and model-assisted checks to ensure accuracy. To further improve LLM performance in detecting and interpreting license clauses, we implemented various prompt strategies,

*Qiang Ke and Xinyi Hou contributed equally to this work.
†Yanjie Zhao is the corresponding author (yanjie_zhao@hust.edu.cn).
‡The full name of the author's affiliation is Hubei Key Laboratory of Distributed System Security, Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology.

including basic, few-shot, contextual, and full prompts, which significantly enhanced clause recognition and reduced hallucinations. **Clause-level analysis led to up to a 50% increase in accuracy over traditional document-level methods and a substantial reduction in hallucination rates**. This framework not only enhances license clause detection but also improves conflict resolution in OSS projects, supporting more reliable compliance analysis. We have publicly released the source code as well as CLAUSEBENCH at https://github.com/security-pride/CLAUSEBENCH.

Contributions. Our primary contributions are as follows:

1) **A fine-grained dataset addressing gaps.** We constructed a high-quality, fine-grained dataset consisting of 634 SPDX-certified licenses, annotated with 3,396 individual clauses across 14 categories. Each clause underwent meticulous review, supported by model-assisted checks, ensuring accuracy and consistency. The dataset achieved a *Cohen's Kappa Coefficient* of 0.896 and an *Inter-Annotator Agreement (IAA)* of 98.08%, confirming its reliability and suitability for benchmarking license clause recognition tasks.
2) **A comprehensive clause-level benchmark.** We developed CLAUSEBENCH, a clause-level benchmarking framework, and evaluated 4 advanced LLMs on over 3,000 clauses. Results show that all models exhibited substantial improvements in clause recognition, achieving a 50% increase in accuracy compared to traditional full-document methods. DeepSeek achieved the highest *Accuracy* (99.83%) with a remarkably low*Hallucination Rate* of 0.12%. We further conducted a fine-grained analysis to compare each model's performance, identifying specific legal distinctions that posed challenges for clause interpretation and conflict detection.
3) **An optimized prompt strategy.** We applied prompt engineering to improve LLM performance in recognizing open-source license clauses, using basic, few-shot, contextual, and full prompts. The contextual prompt, offering detailed explanations and relevant license information, delivered the best results, boosting Mixtral's accuracy by 19% and reducing hallucinations by 26%, while helping DeepSeek maintain 94% accuracy with no hallucinations. Overall, these strategies improved clause detection accuracy by an average of 14%, demonstrating their effectiveness.

II. BACKGROUND AND MOTIVATIONS

The detection of open-source license conflicts faces significant limitations and challenges. Rule-based detection methods struggle to capture the nuanced and evolving nature of legal clauses, often failing to identify non-standard terms and implicit conflicts. Most existing datasets are limited to coarse-grained license annotations, impeding precise clause-level analysis. Although LLMs offer potential in natural language processing applications, their application to license conflict analysis is hampered by insufficient training in specialized legal texts and the inherent variability in clause expression.

Limitations of rule-based detection tools. Current rule-based tools generally rely on static, predefined patterns that are often insufficient to capture the intricate details of legal clauses, particularly in areas such as modification rights, sublicensing, and patent usage [17]. While the SPDX standard's compatibility detection methods effectively handle common conflict scenarios, they lack adaptability to address dynamically evolving clauses [22]. Tools like LiDetector and FOSSology, which depend on fixed patterns and templates for license identification, are restricted in their ability to interpret non-standard clause expressions, limiting their effectiveness [43], [11]. Moreover, open-source projects often incorporate customized or non-standard clauses, which static rule-based methods may overlook, leading to undetected implicit conflicts [39], [23]. This limitation is underscored by empirical research; for example, studies examining license violations in OSS projects reveal that traditional tools frequently miss certain conflict risks, particularly those arising from non-standard clauses [2].

Lack of fine-grained clause annotation in existing datasets. Most existing open-source license datasets are limited to coarse-grained license annotations and lack the fine-grained annotations necessary for accurate clause recognition [13]. When handling complex legal clauses, the absence of fine-grained data affects the accuracy of license conflict detection, causing models to struggle with differentiating subtle clause distinctions [29]. Moreover, while datasets like SPDX provide standardized license descriptions, their clause-level granularity is relatively coarse, making it challenging to support precise detection of complex clauses [44]. This lack of data limits the ability of LLMs to effectively learn complex relationships and semantics between license clauses, which complicates the more refined analysis of license conflicts [18].

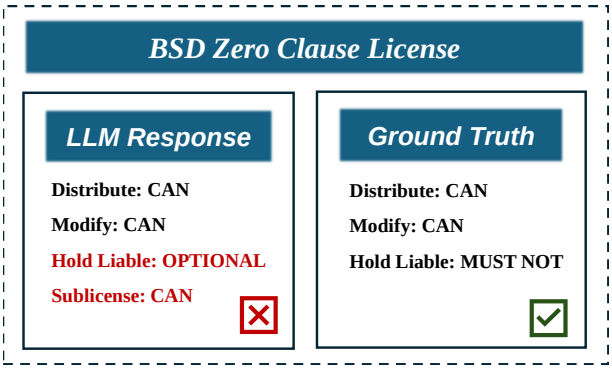

Fig. 1: The hallucination of LLMs.

Challenges of using LLMs in license conflict analysis. Despite the strong performance of LLMs in natural language processing tasks, there are still challenges when applying them to license conflict analysis. First, existing LLMs have not been specifically trained on open-source license texts, leading to the "hallucination" problem where models may generate irrelevant or incorrect content [21], as illustrated in Figure 1. Additionally, the diverse and complex expression of license clauses makes it easy for LLMs to produce inconsistent results when identifying clause content [31]. For instance, complex semantic compositions in clauses often prove difficult for traditional deep learning models to accurately understand, especially when subtle semantic relationships between license terms are involved [33]. Therefore, effective prompt design and the construction of high-quality datasets are essential for

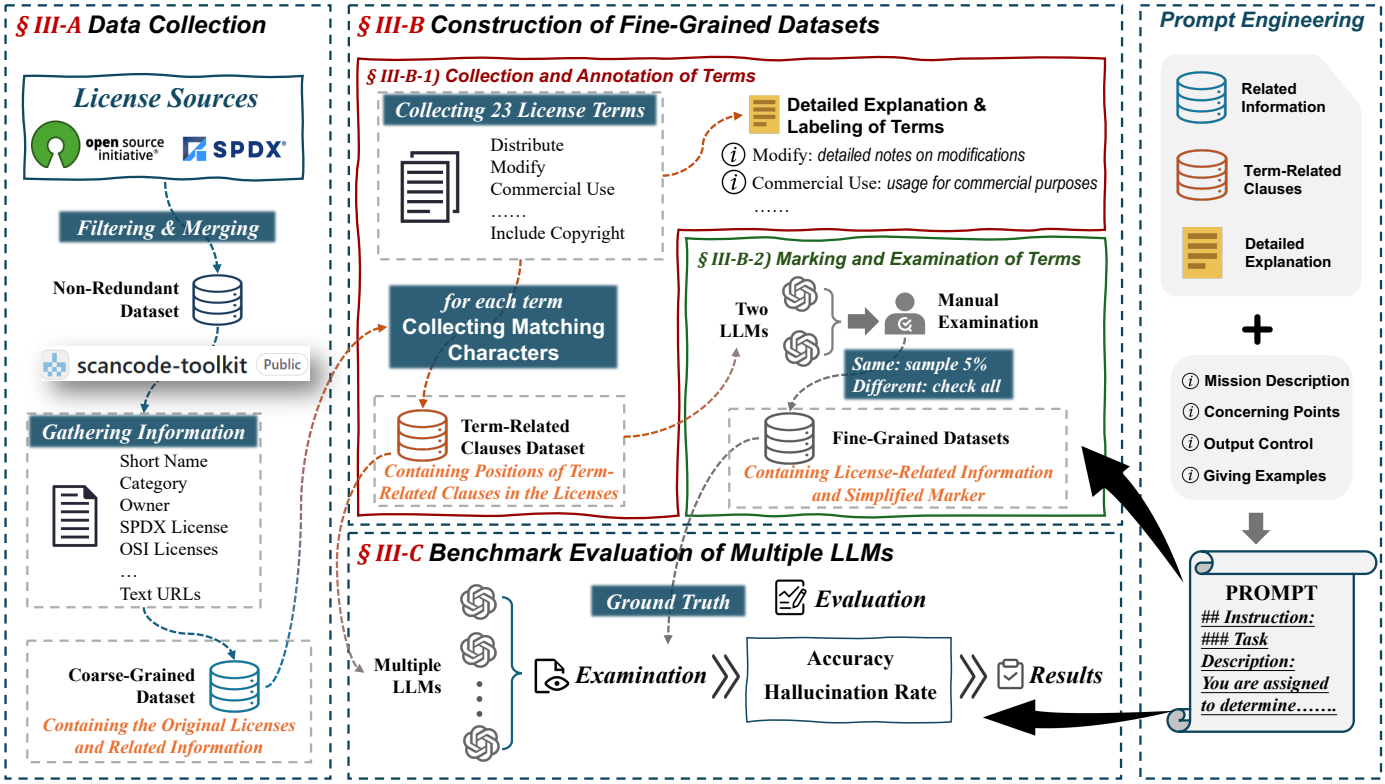

Fig. 2: The construction process of CLAUSEBENCH.

improving the applicability of LLMs in open-source license conflict analysis [18].

Our solutions. To address the limitations in current tools and data resources, our study aims to create a more comprehensive and **fine-grained dataset** for license clause annotation, thereby providing a robust benchmark for training and testing large models. This dataset fills a crucial gap by offering a publicly available, detailed set of clause annotations, allowing for a standardized approach to fine-grained license clause recognition. Additionally, we aim to leverage the capabilities of LLMs for license conflict detection through a carefully designed prompting strategy. Through comprehensive **clause-level benchmarking** of several popular LLMs, we seek to evaluate their performance in terms of recognition, thereby gaining insights into their strengths and limitations. Our work not only advances the use of LLMs for license conflict detection but also lays the foundation for the development of smarter and more reliable methods for OSS license analysis.

III. METHODOLOGY

As shown in Figure 2, we outline our methodology comprising three key steps: ***A. Data Collection***, involving the gathering and filtering of licenses from SPDX and OSI to create a clean, non-redundant dataset; ***B. Construction of Fine-Grained Datasets***, where 23 license terms are annotated and validated through a combination of LLM-based checks and manual reviews; and ***C. Benchmark Evaluation of Multiple LLMs***, in which we evaluate multiple LLMs on their ability to recognize license clauses, using several metrics.

A. Data Collection

Data source. Two major sources were used for open-source license management in this study: the SPDX (Software Package Data Exchange) license list and the Open Source Initiative (OSI) license database.

- **SPDX license list** [34] provides a widely recognized collection of open-source licenses, including standard identifiers, full names, texts, and unique URLs for each license. This enables accurate referencing and compliance in SPDX-compliant documents. From this list, we selected 634 licenses, excluding 71 exception licenses to focus solely on independent licenses.
- **OSI license database** [27], globally recognized for its adherence to the Open Source Definition [26], includes licenses that undergo rigorous review to ensure compliance with core open-source principles. We included 118 licenses from this list, excluding 34 deprecated licenses, resulting in a refined dataset of 84 active OSI licenses.

Data preprocessing. To improve the conciseness and accuracy of the license database, we performed deduplication on the raw data using Monk [6] and Nomo [7]. Monk identifies highly similar licenses through textual similarity analysis, while Nomo detects structural similarities through rule-based pattern matching. Subsequently, we enhanced the license information using the open-source tool ScanCode [1]. ScanCode contains a comprehensive license database and provides detailed supplementary field information for each license, including license category, holder, and copyright notice. The reliability of ScanCode data stems from its open-source nature

and community support, and it has been widely validated for accuracy [16]. In this study, the fields provided by ScanCode include, for example, *key*, *short_name*, *name*, *category*, *owner*, *spdx_license_key*, *standard_notice*, and *homepage_url*. The expanded data fields, detailed in ScanCode's documentation, enhance the LLMs' ability to analyze and classify license clauses with higher accuracy and relevance. Fields such as *category*, *standard_notice*, and *key* assist the LLMs in precisely identifying the normative requirements of each clause during fine-grained clause analysis. Furthermore, fields like *homepage_url* and *text_urls* provide external links for each license, allowing for further consultation of the license's source and updates when necessary.

Through the above collection and preprocessing steps, we constructed a comprehensive, non-redundant **coarse-grained database** of 634 licenses. This process ensures the accuracy and completeness of the dataset, providing a solid foundation for subsequent clause-level analysis.

B. Constructions of Fine-Grained Datasets

The construction of fine-grained datasets for OSS license analysis involves two key steps: 1) the collection and annotation of relevant terms, and 2) the marking and review of these terms to ensure accuracy and consistency.

1) Collection and annotation of terms

We identified 14 distinct terms commonly found in open-source licenses based on prior research [19]. To assist the model in assessing obligations related to these terms and maintain consistency in its output, we categorized them into two main groups: `Rights` and `Obligations` [15]. Table I summarizes these categories, listing each term under the appropriate group. Of the collected terms, 8 were classified as `Rights`, while 6 were categorized as `Obligations`. Each term has been carefully defined and explained to ensure the model has a consistent and comprehensive understanding of their meaning.

TABLE I: Categorization of license terms.

Category	License Terms
Rights	Distribute, Modify, Commercial Use, Relicense, Hold Liable, Use Patent Claims, Sublicense, Use Trademark
Obligations	Include Copyright, Disclose Source, Give Credit, Rename, Contact Author

To comprehensively extract term-related clauses from licenses, we adopted a structured approach using **regular expression-based matching**. This method was designed to capture all instances of relevant clauses across a wide range of licenses, thereby creating a term-related clause dataset to support subsequent analysis by LLMs. Each instance was segmented at the sentence level to ensure that the LLMs could interpret the full context of each clause without losing essential nuances. **Instead of focusing on precise clause boundaries** [18], **we aim for broad coverage of relevant patterns**, minimizing the risk of overlooking key content and reducing potential hallucinations during LLM processing. We then **manually reviewed** a set of representative licenses, including MIT [25], Apache-2.0 [3], GPL-3.0 [9], LGPL-2.1 [8], CC-BY-4.0 [4], and OpenLDAP-2.8 [28]. This review helped identify recurring linguistic patterns and keywords associated with key legal terms, such as `Distribute`, `Modify`, and `Commercial Use`. These patterns formed the basis for a regular expression matching framework capable of detecting a wide range of relevant clauses. The matching table, partially shown in Table II, was iteratively refined during the design process for LLM prompts.

TABLE II: Regex patterns for clause identification.

Clause	Pattern
Distribute	distribute, distribution, redistribute, share copies...
Modify	modify, modification, alter, change, create derivatives...
Commercial Use	offer of sale, resale, use for commercial...
Relicense	relicense, license choice, transfer licensing rights...
Hold Liable	liability, liable, without any warranty, accountability...
Use Patent Claims	use patent claims, assert patent, enforce patent...
Sublicense	sublicense, sub-license, sublicensing, sublicensable...
Use Trademark	use Trademark, trademark, service mark...
Include Copyright	retain copyright, copyright notice, display copyright...
Include License	copy this License, permission notice, keep license...
Disclose Source	disclose Source, source compiled, source code...
Give Credit	credit, acknowledgment, attribution notice...
Rename	rename, not misrepresented, not use the same name...
Contact Author	contact Author, contact us, written consent...

Overall, we analyzed 8,876 instances of term-related clauses. Out of these, 3,396 instances included one or more corresponding statements, while 5,480 instances lacked directly corresponding statements in the original license text. The matching strategy yielded **an average of 5.4 extracted terms per license, effectively covering the essential contents of licenses**, particularly considering that certain licenses may contain ambiguous or redundant clauses. For example, the ABStyles license, consisting of 74 words, included all 4 relevant terms. In contrast, more complex licenses, such as Apache-2.0 and 3D Slicer-1.0, contained 12 and 10 terms, respectively, with a total of approximately 16 terms each.

2) Marking and examination of terms

In the process of annotating and analyzing terms, we utilized two LLMs, i.e., Mistral-large-123B [36] and DeepSeek-v2.5 [35], to generate simplified markers for license terms. Each model independently generated results that encompassed the **simplified markers**, the **pertinent fragments** of the previously extracted instances, and the **associated positional references** within the original documents. This comprehensive output facilitated a clear linkage between the annotations and the source material, enhancing the traceability and interpretability of the results.

Based on our supplementary experiments, we found that the contextual prompt, compared to the basic prompt, few-shot prompt, and full prompt (for full experimental details, see § V-A), yielded the best results, balancing detailed explanations with relevant license information. As a result, we employed the contextual prompt approach during the marking

and examination of terms to optimize model performance in generating accurate and interpretable annotations. The prompt design for license clause analysis is structured into four primary components: task description, simplified markers, input, and output, as presented in Figure 3.

 # Instruction:

Task Description:
You are assigned to determine and interpret the presence and meaning of a specific licensing term within a portion of an open-source software license. Instead of reviewing the entire license, you will be provided with only those sections that are directly relevant to the specified term, along with some contextual license information.
Your goal is to analyze these excerpts and identify whether they address the term and, if so, the term's implications for users according to predefined categories.

Simplified Markers
[Empowering Clauses & Corresponding Markers]
[Responsibility Clauses & Corresponding Markers]
[Ambiguous Cases & Corresponding Markers]

 # Input:
[license_name], [license_info];
[license_terms], [terms_description];
[content_lines], [license_content]

 # Output:
[Expected Output Format]

Fig. 3: Prompt synthesis.

- **Task description.** This part instructs the LLM to interpret the meaning of a specific licensing term by extracting and analyzing relevant sections from an open-source license. The LLM reviews these excerpts to assess whether they adequately address the term and its implications for users. The analysis is structured according to predefined categories, ensuring that the model's output is both consistent and aligned with the overall task objectives.
- **Simplified markers.** Explain the use of simplified clause identifiers, which categorize clauses into two types: *authorization* (permissions) and *responsibility* (obligations). Each type is linked to standardized identifiers to ensure output consistency. To address ambiguity, the *"NOT SPECIFIED"* marker is used for terms with unclear or missing information, ensuring comprehensive coverage. An example output is also provided to clarify the expected format, promoting consistency across annotations.
- **Input.** It consists of three sections: **(1) license information**, which provides the license name, type, and other contextual details; **(2) clause name and explanation**, which links each term to predefined terms and definitions for clarity and consistency; and **(3) related original clause text**, which includes sentences from the original license along with positional information to anchor the model's analysis to the source material.
- **Output.** Require results in a standardized JSON format, including fields for *term*, *marker*, and *explanation*. An example output is provided to reinforce the required format, ensuring consistency and clarity in the analysis of license clauses.

Upon reviewing the initial annotation results, we observed that the **Mistral-large-123B model displayed significant deviations in 11 specific cases**. After cross-referencing these results with the original text, we attributed these deviations to possible hallucinations in the model's interpretation. These cases were manually reviewed and corrected to prevent errors in further analysis. In contrast, the **DeepSeek-v2.5 produced stable results with no major issues**. Out of the 3,396 cases annotated by both models, 275 showed conflicts, as illustrated in Figure 4. To better understand these conflicts, we categorized them into two types: 215 **weak conflicts**, where discrepancies were minor, and 60 **strong conflicts**, where the annotations were directly opposing. Each conflicting case was manually reviewed by two researchers with extensive experience in the OSS field. In cases where disagreements persisted, a third researcher was consulted to make the final decision.

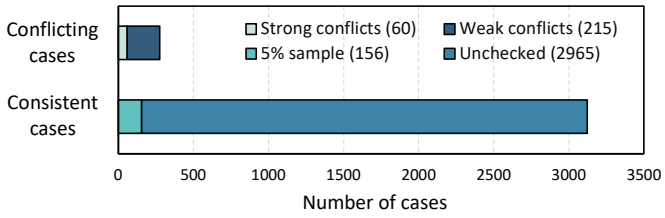

Fig. 4: Distribution of the cases.

For the remaining 3,121 cases where both models produced consistent annotations, a 5% random sample was manually inspected to verify their reliability. Among the 156 cases reviewed, only 3 discrepancies were found, primarily concerning the terms `Disclose Source` and `Rename`. These discrepancies indicate that nuanced interpretations, particularly around complex openness or authorization constraints, can still challenge the models. Through this meticulous process of manual validation, we refined a high-accuracy, fine-grained dataset for license terms, ensuring the precision of term annotations. An example of this dataset, featuring the license *Noweb*, is shown in Figure 5.


```
{
    "term": "Distribute",
    "marker": "CAN",
    "explanation": "4: 'You may
redistribute noweb in whole or in
part provided you acknowledge its
source.'"
},{
    "term": "Modify",
    "marker": "CAN",
    "explanation": "5: 'You may
modify noweb and create derived
works.'"
},{
    "term": "Commercial Use",
    "marker": "CAN",
    "explanation": "6: 'You may
sell noweb if you wish.'"
},{
    "term": "Include Copyright",
    "marker": "MUST",
    "explanation": "5: 'provided
you retain this copyright
notice.'"
},{
    "term": "Disclose Source",
    "marker": "MUST",
    "explanation": "8: 'all
source code for your derived work
is available, at no additional
charge.'"
},{
    "term": "Contact Author",
    "marker": "MUST",
    "explanation": "5: 'may not
be called noweb without my
written consent.'"
}
```


Fig. 5: Fine-grained dataset sample.

C. Benchmark Evaluation of Multiple LLMs

To evaluate the performance of various LLMs, we conduct a benchmark analysis using four models: DeepSeek [35], Mixtral [37], Mistral-large [36], and Qwen [38]. This evaluation

focuses on their ability to analyze OSS license clauses. It is divided into two distinct tasks, each with specific objectives and dataset structures, aimed at assessing both the models' clause recognition capabilities and the effectiveness of the CLAUSEBENCH framework.

The first task examines the LLMs' performance in identifying and interpreting individual clauses within OSS licenses. This task specifically measures each model's accuracy in recognizing fine-grained legal terms and understanding clause-specific details, providing insights into their suitability for clause-level OSS license analysis. The dataset for this task consists of instances directly related to specific clauses, allowing for a detailed assessment of the models' interpretive capabilities. **The second task evaluates the CLAUSEBENCH framework**, focusing on its effectiveness in enhancing OSS license clause analysis and detecting conflicts across complete license documents. Using a comprehensive dataset that includes full OSS license texts, this task evaluates the framework's ability to generalize clause-level insights across entire licenses. The goal is to determine whether CLAUSEBENCH improves the interpretive accuracy of LLMs and enables more thorough OSS license analysis compared to existing methods.

The results of these experiments will be presented in the following evaluation section.

IV. EVALUATION

To evaluate the effectiveness of CLAUSEBENCH, we outline the following research questions (RQs), which capture the primary objectives and focus areas of this study:

RQ1 Dataset validity verification. Is the constructed fine-grained benchmark dataset defined by high quality, accuracy, and consistency?

RQ2 Performance of LLMs on CLAUSEBENCH. How well do some state-of-the-art LLMs (e.g., DeepSeek, Qwen) perform in recognizing OSS license clauses, and do they achieve fine-grained clause recognition accuracy?

RQ3 Performance improvement with CLAUSEBENCH. Can CLAUSEBENCH improve OSS license clause analysis by delivering more precise and efficient clause-level recognition than traditional full-document scanning?

A. Experimental Setup

1) Evaluation metrics

To rigorously evaluate the objectives of three RQs, we employ a comprehensive set of metrics. These metrics, covering annotation consistency, clause recognition accuracy, and performance improvements, provide a robust foundation for assessing the dataset's validity, the accuracy of LLMs, and the effectiveness of our proposed framework, CLAUSEBENCH.

Metrics for RQ1. The following metrics are applied to RQ1 to ensure the quality and consistency of annotations within the dataset, thereby validating its reliability and suitability for model training and benchmarking.

- **Cohen's Kappa Coefficient** (κ): Cohen's Kappa is used to compare annotation consistency between two models, accounting for chance agreement and providing a more robust measure than raw agreement ratios. High Kappa values (e.g., $\kappa > 0.8$) indicate strong consistency, supporting the quality and reliability of the dataset annotations. The Kappa value (κ) is calculated as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{1}$$

Here, p_o is the observed agreement, representing the proportion of instances where both models assigned the same marker. It is calculated as:

$$p_o = \frac{\sum_{i=1}^{k} n_{ii}}{\text{Total}}$$

where n_{ii} are the diagonal elements of the confusion matrix, indicating instances where both models agreed. k is the number of categories, and Total is the total number of instances.
The expected agreement p_e is based on the marginal probabilities for each category, i.e., the proportions of instances each model assigns to each category. Let p_{A_i} and p_{B_i} be the proportions of instances assigned to category i by Model A and Model B, respectively:

$$p_e = \sum_{i=1}^{k} p_{A_i} \times p_{B_i} = \sum_{i=1}^{k} \left(\frac{\sum_{j=1}^{k} n_{ij}}{\text{Total}} \times \frac{\sum_{j=1}^{k} n_{ji}}{\text{Total}} \right)$$

where n_{ij} represents counts of annotations for each cell in the confusion matrix, i.e., the number of instances where Model A assigned category i and Model B assigned category j.

- **Inter-Annotator Agreement (IAA)**: This metric calculates the proportion of consistent annotations between model-assisted and manually adjusted markers. A high IAA value signifies a strong alignment between model outputs and human annotations.

$$\text{IAA} = \frac{N_{\text{Consistent Annotations}}}{N_{\text{Total Annotations}}} \tag{2}$$

Metrics for RQ2 & RQ3. For RQ2, which examines the accuracy of LLMs in recognizing OSS license clauses, and RQ3, which assesses CLAUSEBENCH's performance improvements over traditional methods, we use the following standard classification metrics.

- **Accuracy**: Measures the proportion of correctly identified clauses relative to the total number of annotations.

$$\text{Accuracy} = \frac{N_{\text{Correct Annotations}}}{N_{\text{Total Annotations}}} \times 100\% \tag{3}$$

- **Hallucination Rate**: It is calculated as the proportion of hallucinated instances out of the total instances processed, reflecting the extent to which the model generates inaccurate clauses. A lower Hallucination Rate indicates better accuracy in clause extraction, while a higher rate suggests the model may struggle with generating accurate, relevant outputs for license clauses.

$$\text{Hallucination Rate} = \frac{N_{\text{Hallucinated}}}{N_{\text{Total}}} \times 100\% \tag{4}$$

These metrics provide a comprehensive understanding of annotation consistency (RQ1), the accuracy of LLM

clause recognition (RQ2), and the performance improvements achieved by CLAUSEBENCH in clause-level recognition and analysis (RQ3).

2) Experimental environment

The experiments were conducted on a server equipped with an NVIDIA A100 GPU with 80GB of memory. The server operates under a Linux environment, specifically, Linux version 5.15.0-97-generic, compiled using GCC (Ubuntu 11.4.0-1ubuntu1 22.04) version 11.4.0 and GNU ld (GNU Binutils for Ubuntu) version 2.38.

Four LLMs were used in these experiments. DeepSeek (version 2.5) was accessed via an API for flexible remote processing. Mixtral, a quantized 8x7B version, was deployed locally, using around 26 GB of memory to optimize efficiency without major performance loss. Qwen, with 72B parameters, was deployed in a compressed format requiring 41 GB of memory. Mistral-large, with 123B parameters, was also quantized, using 69 GB of memory to run smoothly on the available hardware.

B. RQ1: Dataset Validity Verification

Since the dataset is foundational for benchmarking LLMs, high annotation consistency and broad coverage of diverse license clauses are essential. We evaluated the dataset's quality by examining annotation accuracy and identifying any omissions in clause extraction. To further assess consistency, we compared the model-assisted annotations with the final labels refined by a human annotator, highlighting the impact of human oversight on ensuring accuracy and reliability.

1) Cohen's Kappa calculation

We employed *Cohen's Kappa Coefficient* to evaluate the annotation consistency between the DeepSeek and Mistral-large in RQ1. This metric evaluates the agreement between two annotators (in this case, two models) while accounting for chance agreement, providing a robust measure of consistency. The calculation was based on the confusion matrix in Table III, which shows annotation counts across five simplified markers: *CAN*, *MUST*, *MUST NOT*, *OPTIONAL*, and *NOT SPECIFIED*.

TABLE III: Confusion matrix for annotation agreement between DeepSeek and Mistral-large.

	CAN	MUST	MUST NOT	OPTIONAL	NOT SPECIFIED
CAN	1396	20	2	1	9
MUST	7	720	2	11	25
MUST NOT	35	30	736	0	37
OPTIONAL	0	6	3	37	4
NOT SPECIFIED	13	27	10	6	259

Using this confusion matrix, we calculated an observed agreement (p_o) of 0.927, indicating a high rate of agreement between the two models. The expected agreement (p_e), which accounts for the probability of random agreement based on marginal probabilities, was 0.297. With these values, ***Cohen's Kappa Coefficient*** (κ) **was determined to be 0.896, reflecting a very high level of agreement between DeepSeek and Mistral-large**. This result indicates strong consistency in their clause annotations, underscoring the reliability of the dataset. Given the minimal discrepancies, the alignment in categorization across defined labels suggests that these models can reliably support the data construction process.

2) IAA calculation

To evaluate annotation consistency, we calculated the *IAA*. Among a total of 3,396 cases, the two models agreed on 3,121. A random 5% sample (156 cases) from these agreements was manually reviewed, and only three clauses required adjustment, resulting in an **observed agreement rate of 98.08%** ($\frac{153}{156}$). Additionally, 275 clauses with inconsistent outputs between the models were manually refined, typically by selecting the output from one model. In rare instances, both outputs were replaced with a new value to improve accuracy. This high *IAA* score demonstrates strong alignment between the models, indicating that the dataset annotations are both consistent and reliable. The high agreement further supports the dataset's credibility as a ground-truth resource for fine-grained clause recognition.

Despite rigorous extraction efforts, some clauses were not fully captured. To ensure that the dataset remains effective, we conducted a statistical analysis of clause extraction counts, as shown in Figure 6. The dataset shows a high coverage rate of key clauses, such as `Distribute`, `Modify`, and `Hold Liable`, which supports its robustness for license analysis despite minor omissions. Less frequent clauses like `Relicense`, `Rename`, and `Contact Author` have minimal impact on overall validity. Clause distribution charts further illustrate the strong representation of essential clauses. Additionally, we conducted post-processing and manual reviews to adjust or supplement any missing annotations, enhancing the dataset's granularity and consistency.

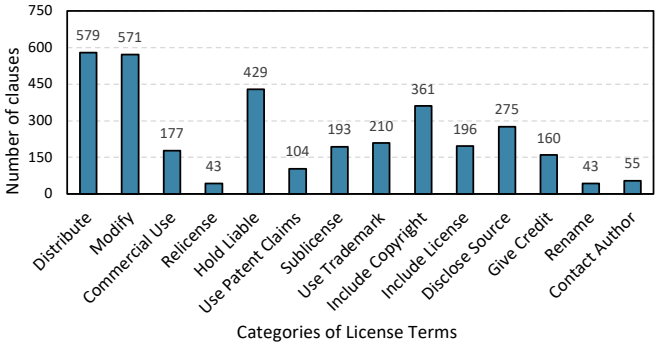

Fig. 6: Distribution of clause extraction counts.

Answer to RQ1. The dataset demonstrates high validity despite minor omissions, with a ***Cohen's Kappa Coefficient*** **of 0.896 and an *IAA* of 98.08%**. This strong alignment, along with high coverage of key clauses, confirms the dataset's reliability. Additionally, post-processing has further improved consistency, ensuring the dataset effectively supports LLM experiments and benchmarking with CLAUSEBENCH.

C. RQ2: Accuracy of LLM Clause Recognition

To address RQ2, we analyzed each model's effectiveness in accurately recognizing and classifying OSS license clauses, with the results shown in Table IV.

TABLE IV: Accuracy and Hallucination Rate of clause-level approach.

Terms	DeepSeek		Mistral-large		Mixtral		Qwen	
	H	A	H	A	H	A	H	A
Distribute	0.17%	99.83%	0.52%	93.96%	6.56%	88.77%	0.69%	96.89%
Modify	N	98.77%	0.88%	90.54%	8.76%	94.40%	0.88%	95.80%
Commercial Use	N	95.48%	N	80.79%	3.95%	83.05%	N	85.31%
Relicense	N	48.84%	N	27.91%	N	13.95%	N	48.84%
Hold Liable	N	97.67%	N	98.37%	2.33%	94.87%	N	99.77%
Use Patent Claims	N	71.15%	N	62.50%	1.92%	60.58%	N	81.73%
Sublicense	N	97.41%	N	97.41%	1.55%	93.26%	N	97.41%
Use Trademark	N	83.81%	N	83.33%	1.90%	78.57%	N	85.24%
Include Copyright	N	98.06%	N	97.51%	0.55%	97.51%	N	97.51%
Include License	N	90.31%	N	88.78%	0.51%	75.00%	N	72.96%
Disclose Source	N	79.64%	0.36%	84.36%	3.27%	76.73%	N	70.18%
Give Credit	N	87.50%	N	95.62%	2.50%	90.62%	N	89.37%
Rename	N	93.02%	N	90.70%	2.33%	86.05%	N	97.67%
Contact Author	N	87.27%	N	90.91%	N	94.55%	N	94.55%
Average	0.12%	93.26%	0.27%	90.28%	3.86%	87.31%	0.27%	90.84%

H: Hallucination rate, **A**: Accuracy, **N**: None.

Among the four models (DeepSeek, Mistral-large, Mixtral, and Qwen), DeepSeek consistently outperforms the others in terms of alignment with the ground truth, demonstrating the lowest rates of accuracy and hallucinations overall. Specifically, **DeepSeek achieves a notable *Accuracy* of 99.83% and a *Hallucination Rate* of 0.12%**, highlighting its strong capability for stable clause recognition. DeepSeek excels in recognizing foundational clauses, such as `Distribute` (99.83% accuracy) and `Commercial Use` (95.84% *Accuracy*), indicating high precision in handling core license terms.

Meanwhile, Mistral-large achieves relatively low *Hallucination Rate* on more complex clauses such as `Disclose Source` (84.36%) and `Give Credit` (95.62%), where other models often struggle. This advantage may be attributed to **Mistral-large's extensive parameter set, which enhances its capacity to handle nuanced legal phrasing**. Similarly, Q wen performs effectively on clauses like `Hold Liable` (99.77%), demonstrating strengths in clauses where accountability and legal entitlements are central. In contrast, Mixtral demonstrates the weakest performance, with an *Accuracy* of 87.31% and a *Hallucination Rate* of 3.86%, indicating notable challenges in clause recognition. Mixtral particularly struggles with complex clauses like `Relicense` (13.95% *Accuracy*) and `Use Patent Claims` (60.58% *Accuracy*), suggesting a limited capacity to interpret clauses requiring deeper legal understanding. Among these clauses, `Use Patent Claims` and `Relicense` also emerge as the most challenging across models, with all models showing elevated accuracy rates for these terms. Notably, DeepSeek and Mistral-large display relatively high accuracy in `Relicense` (48.84% and 27.91%, respectively), further emphasizing the difficulty in achieving accurate clause recognition in this category. This issue may also stem from the relatively limited number of `Relicense` clause instances in the dataset, which restricts the models' exposure to diverse variations of this clause.

Answer to RQ2. **DeepSeek proves to be the most reliable model** with a *Accuracy* of 99.83% and a *Hallucination Rate* of 0.12%, particularly adept at recognizing standard clauses with minimal hallucinations. Mistral-large and Qwen also demonstrate notable strengths, especially in handling attribution-related clauses. In contrast, Mixtral exhibits accuracy, particularly in complex clauses like `Relicense` and `Use Patent Claims`, likely due to insufficient emphasis on complex legal patterns during training. Overall, **complex clauses present challenges across all models**, highlighting the need for further refinement and retraining to better capture nuanced legal distinctions.

D. RQ3: Performance Improvement of CLAUSEBENCH

The clause-level method represents a substantial advancement in the model's ability to accurately and reliably interpret OSS license clauses, particularly in comparison to the traditional full-document scanning approach. By breaking down licenses into several instances about specific clauses, CLAUSEBENCH enables models to capture clause-specific features, significantly improving precision and robustness. As shown in Table V, the clause-level approach significantly improved accuracy and reduced hallucinations across models. Specifically, DeepSeek, which already had a high *Accuracy* of 86.22%, further improved to 93.26%. Meanwhile, Mistral-large, Mixtral, and Qwen each saw notable improvements in *Accuracy*. Mistral-large increased from just over 53.12% to 90.28%, Mixtral improved from nearly 69.67% to 87.31%, and Qwen rose from 58.45% to 90.84%, with **all models achieving approximately 50% relative gains**. This method raised the *Accuracy* of all four models to around 90%, a substantial enhancement in clause recognition performance. Furthermore, the clause-level approach **drastically reduced *Hallucination Rate*, with Mixtral dropping from 25.44% to 3.86%**. The *Hallucination Rate* of the other models remained consistently below 1%, highlighting this method's strength in minimizing unsupported outputs.

TABLE V: Accuracy and Hallucination Rate of the traditional approach.

Terms	DeepSeek		Mistral-large		Mixtral		Qwen	
	H	A	H	A	H	A	H	A
Distribute	N	99.48%	N	64.08%	20.03%	83.07%	N	66.15%
Modify	N	98.95%	N	61.65%	18.74%	70.58%	N	65.32%
Commercial Use	N	91.53%	N	41.24%	29.94%	66.67%	N	42.37%
Relicense	N	58.14%	N	4.65%	60.47%	76.74%	N	23.26%
Hold Liable	N	96.04%	N	63.17%	23.78%	49.42%	N	73.66%
Use Patent Claims	N	50.96%	N	21.15%	33.65%	47.12%	0.96%	27.88%
Sublicense	N	94.82%	N	26.94%	38.34%	61.66%	N	35.75%
Use Trademark	N	50.48%	N	36.19%	39.05%	64.76%	0.48%	47.62%
Include Copyright	N	96.40%	N	61.50%	18.01%	85.87%	N	78.39%
Include License	N	50.51%	N	73.47%	27.55%	66.84%	N	50.00%
Disclose Source	N	66.18%	N	37.09%	20.36%	71.27%	0.36%	43.27%
Give Credit	N	93.12%	N	38.75%	30.63%	71.87%	N	61.87%
Rename	N	44.19%	N	27.91%	27.91%	58.14%	N	44.19%
Contact Author	N	89.09%	N	78.18%	60.00%	69.09%	N	21.82%
Average	N	86.22%	N	53.12%	25.44%	69.67%	0.09%	58.45%

H: Hallucination rate, **A**: Accuracy, **N**: None.

Additionally, the clause-level method significantly enhances the accurate interpretation of specific clauses, yielding varying degrees of accuracy improvement across different terms. For foundational clauses like `Distribute` and `Modify`, recognition *Accuracy* increased by 10-20%. More notably, for complex clauses such as `Sublicense`, Mistral-large and Qwen saw *Accuracy* gains of around 60 percentage points, representing relative improvements of over 200%.

Answer to RQ3. the clause-level method significantly enhances the model's ability to distinguish clause requirements with greater consistency and fewer errors. **All models achieved approximately 50% relative gains in *Accuracy***, with Mixtral's *Hallucination Rate* dropping from 25.44% to 3.86% and

the other models maintaining rates below 1%. This methodology **addresses overgeneralization issues in traditional methods, providing a more reliable framework for license clause analysis and reducing OSS compliance risks**. With its clause-centric training strategy, the clause-level method shows clear advantages in accuracy and interpretive depth, establishing it as a leading solution for precise OSS license clause recognition.

V. Discussion

A. Supplementary experiments

Building upon our previous experiment, we applied prompt engineering to enhance the accuracy of LLMs in identifying and interpreting clauses within license texts. We selected the best-performing model, DeepSeek, and the lowest-performing model, Mixtral, from prior experiments to systematically evaluate their responses to varied prompts. The main objective of this test is to mitigate issues such as LLM hallucinations and recognition errors by implementing a structured approach to prompt design. Due to the variability and complexity of license terms and expressions, we progressively structured four types of prompts to investigate how different levels of context and detail impact model performance. The complete set of prompts used in our experiments can be found in the artifacts[1].

- **Basic prompt.** Only the core task description and the original text of relevant clauses are provided, offering minimal information about the model. The focus is solely on recognizing the clause without any additional context or examples.
- **Few-shot prompt.** Build on the basic prompt by adding output simplified markers with explanations and illustrative examples through few-shot learning. This optimization helps the model by providing reference points, improving its understanding of the task through prior examples, and guiding it to make more accurate predictions for new clauses.
- **Contextual prompt.** Providing detailed explanations for relevant clauses and including specific license information, extends the few-shot prompt and adds valuable context. It is particularly used in the clause-level approach.
- **Full prompt.** In addition to the task description, explanations, and examples, the full prompt also provides the entire text of the license. While this provides comprehensive context, it increases complexity by requiring the model to process the entire document. As a result, the model must process all sections to find relevant clauses, increasing task complexity and time.

TABLE VI: Model Performance Across Different Prompt Types

	Basic		Few-shot		Contextual		Full	
	H	A	H	A	H	A	H	A
DeepSeek	N	84.00%	N	95.00%	N	94.00%	N	90.00%
Mixtral	7.00%	70.00%	40.00%	81.00%	14.00%	89.00%	22.00%	71.00%

H: Hallucination rate, **A**: Accuracy, **N**: None.

[1] https://github.com/security-pride/CLAUSEBENCH

DeepSeek demonstrated consistently strong performance across all prompt types, achieving high accuracy with no detected hallucinations (**H = N**). The few-shot prompt and contextual prompt yielded the highest *Accuracy*, reaching 95% and 94%, respectively, indicating that **combining moderate contextual information with clear labels optimally supports the model's interpretation of complex legal terms**. Accuracy slightly decreased with the full prompt (90%), which suggests that overly detailed prompts may introduce unnecessary information, slightly diminishing interpretative efficiency. The basic prompt provided a lower baseline of effectiveness, as it lacked the additional guidance seen in more contextualized prompts.

In contrast, Mixtral showed relatively high *Hallucination Rate* and lower overall *Accuracy*. However, the few-shot prompt and contextual prompt significantly improved Mixtral's *Accuracy* compared to the basic prompt and full prompt, with increases of approximately 14% and 29%, respectively. Notably, while the few-shot prompt introduced some increase in *Hallucination Rate*, **the contextual prompt achieved both higher *Accuracy* and maintained a comparatively lower *Hallucination Rate***. This outcome suggests that Mixtral benefits from moderate context, as extensive detail in the full prompt may have caused cognitive overload and misinterpretations, while insufficient information in the basic prompt hindered its ability to interpret legal nuances.

This experiment highlights the critical role of prompt engineering in optimizing LLM performance for open-source license conflict analysis. For high-performing LLMs, contextual prompts were shown to enhance interpretive accuracy, demonstrating the importance of providing structured, relevant information. In contrast, LLMs with lower baseline performance in license term recognition benefited from contextual prompts by exhibiting reduced *Hallucination Rate* and improved *Accuracy*. These findings propose an approach to the challenge of insufficient specialized LLMs for open-source license conflict analysis by demonstrating that **carefully tailored, context-rich prompts can reduce hallucinations and improve the reliability** of general-purpose LLMs in license conflict analysis. Furthermore, it highlights the potential of prompt engineering to address some of the limitations in license conflict analysis, though dedicated LLMs trained on license-specific data are likely to yield even more consistent and precise results.

B. Threats to Validity

Dataset coverage limitations. Our dataset comprises 634 standard licenses from the SPDX license list, encompassing all SPDX-certified licenses. However, it excludes non-standard or customized licenses due to the significant time and effort required for their inclusion. Despite this limitation, the dataset remains sufficiently comprehensive for our experiments. Moreover, given the structural similarities in clauses, the ClauseBench framework employed in this study can be extended to recognize clauses in non-standard licenses as well.

Clause extraction omissions. Given the large number of licenses and the need to extract as many relevant original clause statements as possible, we adopted an efficient and controllable keyword-based pattern-matching approach. However, this method may not capture all relevant original statements

in the licenses, especially those with implicit meanings, which is an inherent limitation of the approach. To mitigate this, we initially studied a set of representative licenses and included specific keywords in the matching patterns. Additionally, we refined the matching process throughout the experiments to improve clause extraction coverage as much as possible, achieving over 90% coverage for key clauses such as `Distribution` and `Modify`.

Manual annotation bias. During the manual annotation of 3,396 clauses, there is a potential risk of human error or bias, which could affect model evaluation. To mitigate this, we employed a model-assisted manual annotation process. Consistent model annotations were sampled at 5% for validation, while inconsistent ones were categorized and thoroughly reviewed. We measured annotation consistency using *Cohen's Kappa coefficient (κ)* and *IAA*, achieving κ=0.896 and IAA=98.08%, reflecting a high level of consistency and reliability.

LLM selection constraints. In our experiments, we used four models but did not include cutting-edge options like ChatGPT-4o or Claude 3.5. As a result, the findings may not fully reflect the framework's potential performance with the most advanced models, which could limit the completeness of the evaluation. However, given the complexity and size of the dataset, using such state-of-the-art models would have significantly increased the experimental costs. The models we selected provided a well-balanced performance spectrum, allowing us to effectively demonstrate the framework's generalizability across different models. Furthermore, within the CLAUSEBENCH framework, all selected models achieved approximately 50% higher accuracy compared to traditional methods, confirming the efficiency of our approach.

VI. RELATED WORK

A. License Term Extraction

FindOSSLicense [15] applies manual analysis to classify and summarize sentences from 24 open-source license texts, aiming to identify terms that can effectively contribute to a license modeling framework. *FOSS-LTE* [18] implements an automated method using Latent Dirichlet Allocation (LDA) for topic modeling. While LDA-based topic modeling facilitates automation, it can also introduce noise, potentially reducing the accuracy of term extraction. Xiao et al. [42] present a label-specific document representation approach for multi-label text classification, which can be applied to license term extraction by identifying terms associated with specific categories, enhancing classification effectiveness. German et al. [10] propose a sentence-matching method for automatic license identification in source code files, matching to accurately identify relevant phrases. Tuunanen et al. [39] describe methods for automated software license analysis, highlighting the extraction of critical terms and elements for compliance checks and conflict detection. *LiDetector* [43] advances term extraction through a clustering-based preprocessing stage and a two-phase learning process involving entity extraction and inference of rights and obligations.

Our method builds upon the insights from these prior approaches but introduces a regular expression-based framework refined iteratively through prompt testing with LLMs. This approach utilizes carefully crafted regular expressions that capture linguistic patterns directly associated with terms, effectively reducing the extraction of irrelevant instances. By operating at the sentence level and capturing broad instances across diverse licenses, our approach minimizes the likelihood of missed terms and mitigates potential hallucinations in LLMs, providing consistent input for LLMs to interpret the nuanced meanings within each clause.

B. License Terms Analysis

Traditional license detection tools, such as FOSSology [11], use a binary Symbol Alignment Matrix (bSAM) to identify licensing terms based on symbol patterns. This works well for detecting known license texts but struggles with non-standard or evolving clauses. Another tool, Ninka [10], employs a sentence-based approach for license identification in source code, improving efficiency for common licenses but falling short in capturing nuanced clauses across diverse licenses. Blockchain-based systems such as LUCE [40], [12] and the Compliance Adherence and Governance (CAG) framework [32] have emerged to enforce compliance through smart contracts. LUCE periodically verifies license compliance on a blockchain network, while CAG encodes compliance policies as smart contracts, enhancing accountability in OSS licensing. Recent advancements include machine learning-based models that facilitate understanding of non-standard clauses. In particular, research leveraging attention mechanisms, such as in BERT-based classifiers [20], enables the parsing of lengthy license texts and identification of previously unseen terms through context-driven learning.

To address the limitations of prior tools, we propose a clause-level approach that focuses on the detection and interpretation of individual license clauses. This method allows LLMs to analyze each clause as an independent entity, overcoming the ambiguity and overlap issues seen in whole-document scanning. By isolating specific clauses and tagging them with standardized markers, our clause-level approach significantly enhances interpretive accuracy, facilitating more reliable OSS license compliance.

VII. CONCLUSION

This paper addresses the key challenges in applying LLMs to open-source license analysis by constructing a fine-grained, high-quality dataset, employing clause-level detection methods, and optimizing general-purpose LLMs through targeted prompt engineering. Our dataset, which includes 634 SPDX-certified licenses and a total of 3,396 clauses with simplified annotations, has undergone meticulous review and verification, establishing a solid foundation for license conflict detection. The introduction of the CLAUSEBENCH framework significantly enhances LLM accuracy, achieving approximately 50% improvement over traditional methods, while also substantially reducing hallucinations. Furthermore, our contextual prompt engineering strategy achieved considerable performance gains for general-purpose LLMs in open-source license analysis, with the accuracy rate reaching around 90%. Overall, our work establishes a strong foundation for future advancements in automated, reliable open-source license conflict detection and highlights the potential of LLMs in addressing complex challenges within open-source licensing.

Acknowledgments

This work was supported in part by the Key R&D Program of Hubei Province (2023BAB017, 2023BAB079), HUST CSE-HongXin Joint Institute for Cyber Security, HUST CSE-FiberHome Joint Institute for Cyber Security, and the Xiaomi Young Talents Program.

References

[1] AboutCode Organization, "Scancode toolkit," https://github.com/aboutcode-org/scancode-toolkit, 2024, accessed: 2024-11-03.

[2] D. A. Almeida, G. C. Murphy, G. Wilson, and M. Hoye, "Do software developers understand open source licenses?" in *2017 IEEE/ACM 25th International Conference on Program Comprehension (ICPC)*, 2017, pp. 1–11.

[3] Apache Software Foundation, "Apache License, Version 2.0," 2004, [Accessed: 2024-11-02]. [Online]. Available: https://spdx.org/licenses/Apache-2.0.html

[4] Creative Commons Corporation, "Creative Commons Attribution 4.0 International," 2013, [Accessed: 2024-11-02]. [Online]. Available: https://spdx.org/licenses/CC-BY-4.0.html

[5] X. Cui, J. Wu, Y. Wu, X. Wang, T. Luo, S. Qu, X. Ling, and M. Yang, "An empirical study of license conflict in free and open source software," in *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2023, pp. 495–505.

[6] FOSSology Project, "Monk: Fossology license detection tool," https://github.com/fossology/fossology/wiki/Monk, 2024, accessed: 2024-11-02.

[7] ——, "Nomos: Fossology license detection tool," https://github.com/fossology/fossology/wiki/Nomos, 2024, accessed: 2024-11-02.

[8] Free Software Foundation, "GNU Lesser General Public License, Version 2.1," 1999. [Online]. Available: https://spdx.org/licenses/LGPL-2.1.html

[9] ——, "GNU General Public License, Version 3," 2007, [Accessed: 2024-11-02]. [Online]. Available: https://spdx.org/licenses/GPL-3.0.html

[10] D. M. German, Y. Manabe, and K. Inoue, "A sentence-matching method for automatic license identification of source code files," in *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*. IEEE, 2010, pp. 437–446.

[11] R. Gobeille, "The fossology project," in *Proceedings of the 2008 International Working Conference on Mining Software Repositories*, ser. MSR '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 47–50. [Online]. Available: https://doi.org/10.1145/1370750.1370763

[12] A. Havelange, M. Dumontier, B. Wouters, J. Linde, D. Townend, A. Riedl, and V. Urovi, "Luce: A blockchain solution for monitoring data license accountability and compliance," 2019. [Online]. Available: https://arxiv.org/abs/1908.02287

[13] Y. Higashi, Y. Manabe, and M. Ohira, "Clustering oss license statements toward automatic generation of license rules," in *2016 7th International Workshop on Empirical Software Engineering in Practice (IWESEP)*, 2016, pp. 30–35.

[14] C.-W. Huang and Y.-N. Chen, "Factalign: Long-form factuality alignment of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2410.01691

[15] G. M. Kapitsaki and G. Charalambous, "Modeling and recommending open source licenses with findosslicense," *IEEE Transactions on Software Engineering*, vol. 47, no. 5, pp. 919–935, 2021.

[16] G. M. Kapitsaki and F. Kramer, "Open source license violation check for spdx files," in *Proceedings of the Software Reuse for Dynamic Systems in the Cloud and Beyond*. Springer, 2014, pp. 90–105.

[17] G. M. Kapitsaki, F. Kramer, and N. D. Tselikas, "Automating the license compatibility process in open source software with spdx," *Journal of Systems and Software*, vol. 131, pp. 386–401, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121216300905

[18] G. M. Kapitsaki and D. Paschalides, "Identifying terms in open source software license texts," in *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, 2017, pp. 540–545.

[19] kevin, "Software licenses in plain english," https://tldrlegal.com/, 2012, accessed: 2024-11-03.

[20] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," 2020. [Online]. Available: https://arxiv.org/abs/1909.11942

[21] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:2407601

[22] librariesio, "Check compatibility between different spdx licenses for checking dependency license compatibility," 2015, retrieved from GitHub, Accessed: 2024-11-02. [Online]. Available: https://github.com/librariesio/license-compatibility

[23] A. Mathur, H. Choudhary, P. Vashist, W. Thies, and S. Thilagam, "An empirical study of license violations in open source projects," in *2012 35th Annual IEEE Software Engineering Workshop*, 2012, pp. 168–176.

[24] J. Mukhoti, V. Kulharia, A. Sanyal, S. Golodetz, P. H. S. Torr, and P. K. Dokania, "Calibrating deep neural networks using focal loss," 2020. [Online]. Available: https://arxiv.org/abs/2002.09437

[25] Open Source Initiative, "MIT License," 1988, [Accessed: 2024-11-02]. [Online]. Available: https://spdx.org/licenses/MIT.html

[26] ——, "Open source definition," 2024, accessed: 2024-11-02. [Online]. Available: https://opensource.org/osd

[27] ——, "Open source initiative," 2024, accessed: 2024-11-02. [Online]. Available: https://opensource.org/

[28] OpenLDAP Foundation, "OpenLDAP Public License, Version 2.8," 1998, [Accessed: 2024-11-02]. [Online]. Available: https://spdx.org/licenses/OLDAP-2.8.html

[29] D. Paschalides and G. M. Kapitsaki, "Validate your spdx files for open source license violations," *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:17156519

[30] J. Reddy, "The consequences of violating open source licenses," 2015, accessed: 2024-11-02. [Online]. Available: https://btlj.org/2015/11/consequences-violating-open-source-licenses/

[31] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *ArXiv*, vol. abs/1707.06799, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:25934949

[32] K. Singi, V. Kaulgud, R. J. C. Bose, and S. Podder, "Cag: Compliance adherence and governance in software delivery using blockchain," in *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 2019, pp. 32–39.

[33] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Conference on Empirical Methods in Natural Language Processing*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:990233

[34] SPDX, "Spdx license," 2024, accessed: 2024-11-02. [Online]. Available: https://spdx.dev/

[35] D. A. Team, "Deepseek model v2.5," 2024. [Online]. Available: https://www.deepseek.com/

[36] M. A. Team, "Mistral-large model 123 billion parameters," 2024. [Online]. Available: https://mistral.ai/

[37] ——, "Mistral model 8x7b," 2024. [Online]. Available: https://mistral.ai/

[38] Q. D. Team, "Qianwen model 72 billion parameters," 2024. [Online]. Available: https://tongyi.aliyun.com/qianwen/

[39] T. Tuunanen, J. Koskinen, and T. J. Kärkkäinen, "Automated software license analysis," *Automated Software Engineering*, vol. 16, pp. 455–490, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:21323232

[40] V. Urovi, V. Jaiman, A. Angerer, and M. Dumontier, "Luce: A blockchain-based data sharing platform for monitoring data

license accountability and compliance," 2022. [Online]. Available: https://arxiv.org/abs/2202.11646

[41] J. Wu, L. Bao, X. Yang, X. Xia, and X. Hu, "A large-scale empirical study of open source license usage: Practices and challenges," in *Proceedings of the 21st International Conference on Mining Software Repositories*, ser. MSR '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 595–606. [Online]. Available: https://doi.org/10.1145/3643991.3644900

[42] L. Xiao, X. Huang, B. Chen, and L. Jing, "Label-specific document representation for multi-label text classification," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 466–475. [Online]. Available: https://aclanthology.org/D19-1044

[43] S. Xu, Y. Gao, L. Fan, Z. Liu, Y. Liu, and H. Ji, "Lidetector: License incompatibility detection for open source software," *ACM Transactions on Software Engineering and Methodology*, vol. 32, pp. 1 – 28, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:248366526

[44] X. Zhan, T. Liu, L. Fan, L. Li, S. Chen, X. Luo, and Y. Liu, "Research on third-party libraries in android apps: A taxonomy and systematic literature review," *IEEE Transactions on Software Engineering*, vol. 48, no. 10, pp. 4181–4213, 2022.