

Understanding Open-source Licenses with License Term Identification Using Large Language Models

TOMOHITO NAGASAKI^{1,a)} WATARU OTODA¹ TETSUYA KANDA²
KATSURO INOUE³ YUKI MANABE⁴ YOSHIKI HIGO¹

Received: April 24, 2026, Accepted: June 10, 2026

Abstract: Open-source license compliance is essential but often time-consuming. This study examines the applicability of large language models (LLMs) to license interpretation, performing binary classification on every combination of license, license term, and proposition type (“Can,” “Cannot,” or “Must”). Using Chain-of-Thought (CoT) prompting to steer the models’ reasoning, we found that LLMs deliver reasonably accurate predictions, indicating their potential as compliance aids. Analysis shows that accuracy varies by license term and proposition type, and highlights cases where model outputs diverge from the ground truth. The results suggest that integrating external legal knowledge and clearer definitions of license terms could further improve classification performance.

Keywords: open-source license analysis, license terms, large language models, Chain-of-Thought

1. Introduction

Using open-source software (OSS) is a common strategy to reduce software development costs. Each OSS is distributed under an open-source license (hereinafter “license”), which specifies the conditions for its use, and permits use only when those conditions are met. Although complying with licenses is essential, understanding them can often be challenging. Accurate interpretation of a license can be burdensome, as it requires legal knowledge and an understanding of the software’s usage. In addition, custom or less common licenses often lack third-party explanations or documentation compared to mainstream licenses. Moreover, verifying the license compatibility of multiple OSS components, i.e., whether the licenses permit their combination, is essential. Therefore, complying with licenses requires time and effort. To prevent costly revisions arising from the late detection of license violations, potential issues should be identified early in the development process. Hence, the difficulty of ensuring license compliance remains a significant challenge.

We refer to *license terms* as the actions that a license allows or requires users to perform. The website tl;drLegal [3] defines 23 license terms and classifies them into three proposition types (“Can,” “Cannot,” or “Must”) for each license. Performing similar analysis manually is time-consuming, so previous studies applied natural language processing (NLP) to predict license term propositions [6], [12]. Notably, LiDetector [12] outperforms previous NLP-based methods, achieving a recall of 69.70%, a precision of 85.88%, and an F1 score of 76.95%. It uses machine

learning models to identify license segments corresponding to 23 terms defined in tl;drLegal and label each with a proposition type. Additionally, following the other previous study [6], LiDetector classifies the 23 license terms into two categories, as illustrated in **Table 1**: *rights* and *obligations*. Rights primarily describe actions that are either permitted or prohibited and are assigned “Can” or “Cannot.” Obligations primarily describe actions that are required and are assigned “Must.”

Large language models (LLMs) have demonstrated superior performance on a wide range of NLP tasks [1]. Their ability to interpret legal language makes them attractive for license text analysis, and recent studies have begun to apply LLMs to this domain [2], [5], [7]. Nevertheless, the use of LLMs for license document analysis remains in its early stages and little attention has been paid to the reasoning processes behind LLMs’ inferences.

In this study, we applied LLMs to classify the propositions of the license terms. We employed Chain-of-Thought (CoT) prompting to guide the LLM’s reasoning process and analyzed both the accuracy and the reasoning processes.

This study shows that LLMs can serve as a tool for license

Table 1 License terms [3] categorized into rights and obligations [12].

Right	Obligation
Distribute	Include Copyright
Modify	Include License
Commercial Use	Include Notice
Relicense	Disclose Source
Hold Liable	State Changes
Use Patent Claims	Include Original
Sublicense	Give Credit
Statically Link	Rename
Private Use	Contact Author
Use Trademark	Include Install Instructions
Place Warranty	Compensate for Damages
	Pay Above Use Threshold

¹ The University of Osaka, Suita, Osaka 565–0871, Japan

² Notre Dame Seishin University, Okayama 700–8516, Japan

³ Ritsumeikan University, Ibaraki, Osaka 567–8570, Japan

⁴ The University of Fukuchiyama, Fukuchiyama, Kyoto 620–0886, Japan

^{a)} t-nagask@ist.osaka-u.ac.jp

analysis, that accuracy varies with license terms and proposition types, and that work should incorporate legal knowledge and term descriptions to improve performance.

2. Objective and Approach

2.1 Research Objective and Research Questions

The objective of this study is to evaluate the performance of predicting the propositions of the license terms of open-source licenses using LLMs and to assess the applicability of LLMs. We set the following research questions:

RQ1: How accurately can LLMs predict the propositions of the license terms defined by tl;drLegal?

RQ2: Are there differences in performance across different classification tasks related to license terms and proposition types?

RQ3: What are the characteristics of the answers provided by LLMs that do not match the ground truth?

2.2 Approach Overview

Following previous studies, 23 license terms are categorized into two groups: rights and obligations, as shown in Table 1. For each of 11 rights, the LLM is asked two questions: whether a license permits a certain license term (“Can”) and whether it prohibits the license term (“Cannot”). “Can” and “Cannot” may both be predicted as positive for the same right, but we do not reconcile such cases. For each of 12 obligations, the LLM is asked one question: whether a license requires compliance with a certain license term (“Must”). In total, these queries constitute 34 binary classification tasks for each license.

To address RQ1, we evaluate the classification performance of each model using three standard metrics of precision, recall, and F1 score, following the methodology of previous work [12]. For RQ2, we investigate differences in classification accuracy across 34 classification tasks common to all licenses, defined by combinations of license terms and proposition types. Finally, to answer RQ3, we analyze the instances that deviate from the ground truth.

3. Experiment

3.1 Prompt Design for Classification Task

A prompt for each binary classification task is designed as follows. Chain-of-Thought (CoT) prompting [11] and role-play prompting [8] are used to improve the accuracy of the LLMs’ responses. **Figure 1** illustrates an example prompt designed to classify whether a license permits “Commercial Use”. The [License document] in blue text contains the full text of the license.

The prompt consists of three parts. The system prompt instructs the model to behave as a legal language analyst specializing in software licenses (Prompt 1). The user prompt guides the model through structured reasoning: it extracts text related to the target term, evaluates whether the license explicitly grants the permission, and checks the consistency of its inference (Prompt 2). A separate user instruction requests a conclusive “Yes” or “No” answer based on the prior reasoning (Prompt 3).

For other tasks, the “**permit/permited**” in red text in Fig. 1 is replaced with “**prohibit**” or “**obligate**”, and the “**Commercial Use**” and its description in orange text are changed to the corresponding license term and description, as listed in Table 1.

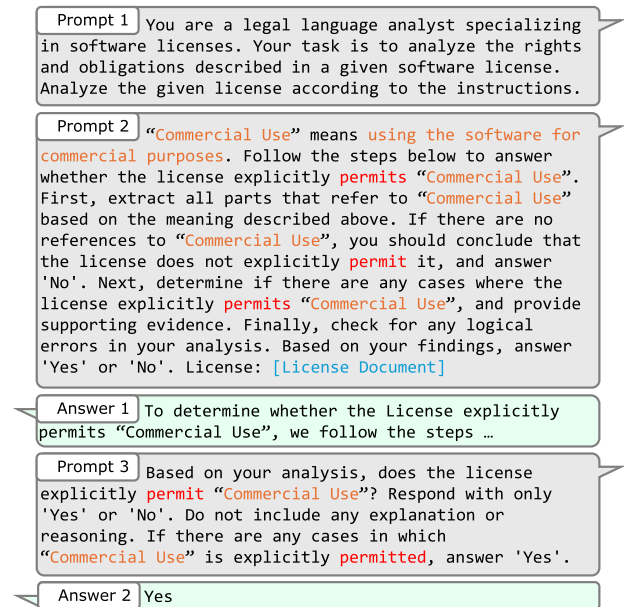


Fig. 1 Example prompt for classifying permission of “Commercial Use”.

We use descriptions provided in the previous study [3], but due to grammatical considerations, the description of the term “Statistically Link” is modified to: “compiling the library into the program linked at compile time rather than runtime.”

3.2 Dataset and Models

We use a dataset of OSI-approved open-source licenses obtained from the tl;drLegal database. Since there are no license terms labeled for PHP License 3.0.1, we excluded it from the dataset, which finally contains 58 licenses. To prevent data leakage of open-source licenses that may appear in LLM corpora, we obfuscate proper nouns such as license names and organization names by shifting each alphabetical character one position.

We selected two models each from local environments and cloud-based with varying performance levels: Llama 3.1 8B and Llama 3.3 70B [9], and GPT-4o and GPT-4o mini [10]. GPT-4o series are a high-performance cloud-based model, while the Llama 3 series allowing them to run locally and offering stronger data security in practical use cases.

3.3 Evaluation of Differences in Classification Accuracy

We test, for each model, the null hypothesis that classification accuracy is equal across license terms and proposition types using a binomial test. We compute a p-value for each task and reject the null hypothesis if any p-value falls below the significance threshold. We do not account for the proportion of positive examples per classification task or for differences between licenses. We apply the Bonferroni correction to control the family-wise error rate for multiple testing. With a conventional 5% significance level, the adjusted threshold is $5\%/34 \approx 0.147\%$, and the null hypothesis is rejected if any p-value falls below this level.

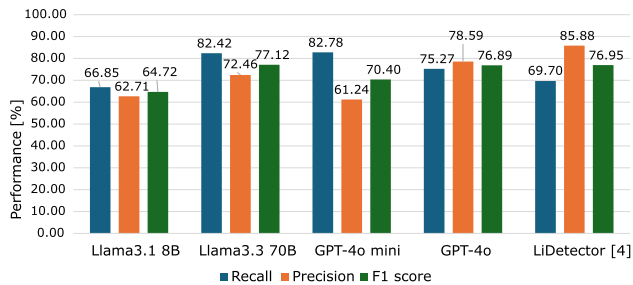
4. Results and Discussions

4.1 RQ1: How Accurately Can LLMs Predict the Proposition of the License Terms Defined by tl;drLegal?

The performances for metrics are summarized in Fig. 2. Note

Table 2 Classifications whose p-values fall below the significance level.

Llama3.3 70B		GPT-4o	
Upper-tail p-value	Lower-tail p-value	Upper-tail p-value	Lower-tail p-value
Modify (Cannot), Distribute (Cannot), Private Use (Cannot), Use Patent Claims (Cannot), Pay Above Use Threshold (Must)	Private Use (Can), Use Patent Claims (Can), Include Notice (Must), Give Credit (Must)	Private Use (Cannot), Use Patent Claims (Cannot), Pay Above Use Threshold (Must)	Commercial Use (Can), Sublicense (Can), Private Use (Can), Use Patent Claims (Can), Include License (Must)

**Fig. 2** Performance of LLMs and LiDetector.

that the results for LiDetector are taken from the original paper [12] and are based on a different dataset. The F1 scores of the four LLMs ranged from 64.72 to 77.12, demonstrating performance comparable to or slightly better than that of LiDetector. The recall of all models except Llama3.1 8B exceeded that of LiDetector, while precision was lower across the board. The high-capacity models (Llama3.3 70B and GPT-4o) achieved performance comparable to state-of-the-art methods.

Our experiments show that even a simple CoT prompting strategy, without any fine-tuning, can perform comparably to LiDetector. Given the high recall of LLMs, which reduces the risk of overlooking obligations or prohibitions in practical scenarios, LLMs have a potential to support the early stages of license review by generating coarse summaries and identifying candidate sections for further examination. Given the rapid advancement of LLMs, they have the potential to become a practical alternative to traditional methods for license analysis.

Answer to RQ1: LLMs show sufficient accuracy to be useful for license analysis.

4.2 RQ2: Are There Differences in Performance across Different Classification Tasks Related to License Terms and Proposition Types?

In hypothesis testing, both upper-tail and lower-tail p-values were computed for each classification task. Upper-tail p-values below the significance level indicate easier tasks for LLMs, whereas lower-tail p-values indicate more difficult ones.

The null hypothesis was rejected for all models, indicating significant performance differences across tasks. **Table 2** presents only the results for the high-capacity models (Llama3.3 70B and GPT-4o), as space constraints on the page. Across all models (including small models), tasks such as “Private Use (Cannot)” and “Pay Above Use Threshold (Must)” showed low upper-tail p-values, indicating higher accuracy. In contrast, tasks like “Private Use (Can)” and, in several models, “Give Credit (Must),” had low lower-tail p-values, suggesting they are more challenging.

Answer to RQ2: LLMs show varying levels of accuracy depending on the classification task.

4.3 RQ3: What Are the Characteristics of the Answers Provided by LLMs That Do Not Match the Ground Truth?

We focused on the Llama3.3 70B, which achieved the highest F1 score in our experiment, and the five most popular licenses on GitHub as of 2024 [4]: the MIT License (MIT) ^{*1}, the Apache License version 2.0 (Apache-2.0) ^{*2}, the GNU General Public License version 3.0 (GPL-3.0) ^{*3}, the GNU Affero General Public License version 3.0 (AGPL-3.0) ^{*4}, and the BSD 3-Clause License (BSD-3-Clause) ^{*5}. We found 24 cases where the ground truth and the LLM’s answers were inconsistent. In 6 cases, the model produced the correct answer during the reasoning process (Answer 1 in Fig. 1) but gave an incorrect final output (Answer 2 in Fig. 1).

The remaining 18 cases are shown in **Table 3**. 3 of them appear to be clear errors not requiring legal expertise: “MIT: Include Notice (Must),” “GPL-3: Place Warranty (Can),” “AGPL-3: Include Original (Must).” For the remaining cases, it can be inferred from the reasoning process that there was a lack of legal knowledge or clear understanding of license terms. For example, the interpretation of Making, running, and propagating without conveying as “Private Use” differs between LLMs and tldrLegal. Providing appropriate legal knowledge for “Private Use” could help the LLM’s proper reasoning. Similarly, LLMs differ from tldrLegal on whether including copyright notices counts as “Give Credit.” Clarifying the definition of “Give Credit” could improve agreement between them.

Answer to RQ3: The reasoning process suggests that wrong answers are the result of a lack of legal knowledge and a limited understanding of license terms.

5. Threats to Validity

Internal Validity: Prompt sensitivity may cause issues with reproducibility, as variation in prompt wording and structure may affect the responses generated by the LLMs. We used established prompt engineering techniques and designed the prompts based on previous research, thereby reducing the risk. Statistical bias may also be present in the statistical tests comparing performance across different classification tasks, since we did not account for differences in the proportion of positive examples or variations across licenses. Therefore, our discussion is limited to pointing out the presence of differences, and any trends observed across terms should be regarded as hypothetical rather than the result of detailed analysis. There is a risk of training data leakage, as some open-source licenses may have been included in the LLM train-

^{*1} <https://opensource.org/license/mit>

^{*2} <https://opensource.org/license/apache-2-0>

^{*3} <https://opensource.org/license/gpl-3-0>

^{*4} <https://opensource.org/license/agpl-v3>

^{*5} <https://opensource.org/license/BSD-3-Clause>

Table 3 Cases where the ground truth and the LLM (Llama3.3 70B)’s final output differ.

License: Term (Proposition Type)	Reasoning Summary	Ground Truth / Final Output
MIT: Include License (Must)	Required stated as “The above copyright notice and this permission notice shall be included” but not the full text.	Yes / No
MIT: Include Notice (Must)	Inclusion of copyright is required stated as “The above copyright notice and... shall be included”, corresponding to “Include Notice.”.	No / Yes
MIT: Give Credit (Must)	Inclusion of copyright is required stated as “The above copyright notice and... shall be included”, corresponding to “Give Credit.”	No / Yes
Apache-2.0: Relicense (Can)	Addition of other licenses or terms to derivative works permitted in Section 4, corresponding to “Relicense.”	No / Yes
Apache-2.0: State Changes (Must)	Notices stating that you changed the files is required in Section 4, but a detailed log or description of changes isn’t required.	Yes / No
Apache-2.0: Give Credit (Must)	Inclusion of copyright, patent, trademark, and attribution notices is required in Section 4, corresponding to “Give Credit.”	No / Yes
GPL-3.0: Private Use (Can)	Making, running, and propagating without conveying is permitted in Section 2, corresponding to “Private Use.”	No / Yes
GPL-3.0: Place Warranty (Can)	No statement about “Place Warranty.”	Yes / No
GPL-3.0: Give Credit (Must)	Stating modifications and date is required in Section 5, corresponding to “Give Credit.”	No / Yes
AGPL-3.0: Private Use (Can)	Making, running, and propagating without conveying is permitted in Section 2, corresponding to “Private Use.”	No / Yes
AGPL-3.0: Use Patent Claims (Can)	A non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims is granted in Section 11.	No / Yes
AGPL-3.0: State Changes (Must)	Stating that it was modified and the date is required in Section 5, but detailing the specific changes are not required.	Yes / No
AGPL-3.0: Include Original (Must)	Distributing and making available the source of covered work is required in Section 6 and 13, corresponding to “Include Original.”	No / Yes
AGPL-3.0: Give Credit (Must)	Inclusion of copyright, modification notices, and author attributions is required in Sections 4, 5, and 7, corresponding to “Give Credit.”	No / Yes
AGPL-3.0: Include Install Instructions (Must)	Installation Information is required in Section 6, but not correspond to “Include Install Instructions.”	Yes / No
BSD-3-Clause: Sublicense (Can)	No statement about “Sublicense.”	Yes / No
BSD-3-Clause: Place Warranty (Can)	No statement about “Place Warranty.”	Yes / No
BSD-3-Clause: Contact Author (Must)	Name of the copyright holder and contributors with outspecific prior written permission is prohibited in Section 3, corresponding to “Contact Author.”	No / Yes

ing data. To mitigate this, we anonymized proper nouns in the license texts, which probably reduces the impact on validity.

External Validity: This study is based on a specific dataset consisting of 58 OSI-approved licenses from tl;drLegal. The results may not generalize to other licenses, especially lesser-known or newly created ones. In addition, we only evaluated a limited set of LLMs. Different models may produce different outcomes. However, we believe the risk of these biases is limited in practical scenarios, as both the licenses and the LLMs are widely used.

6. Conclusion

In this study, we evaluated the performance of LLMs in predicting the proposition of the license terms of open-source licenses based on the 23 terms defined by tl;drLegal. The results demonstrated that high-performing LLMs such as Llama3.3 70B and GPT-4o achieved F1 scores comparable to or exceeding those of LiDetector, even without any task-specific fine-tuning. Additionally, we observed performance variations across different classification tasks and discussed that the lack of legal knowledge and a limited understanding of license terms could contribute to reduced classification accuracy. These findings suggest that providing legal knowledge and clear understanding of the details of the license terms could improve classification accuracy.

LLMs are shown to be a promising tool for open-source license analysis and compliance support. As future work, improving prompts and applying fine-tuning techniques may further enhance classification accuracy.

Acknowledgments This work was supported by JSPS KA-

KENHI Grant NOs. JP23K28065, JP24K14895, JP24H00692, JP25K03102, JP26K02889, JP26H02500.

References

- [1] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q. and Xie, X.: A Survey on Evaluation of Large Language Models, *ACM Trans. Intelligent Systems and Technology*, Vol.15, No.3, pp.1–45 (2024).
- [2] Cui, X., Wu, J., Ling, X., Luo, T., Yang, M. and Ou, W.: Exploring Large Language Models for Analyzing Open Source License Conflicts: How Far Are We?, *Proc. ICSE2025*, pp.291–302 (2025).
- [3] FOSSA: tl;drlegal (online), available from (<https://www.tldrlegal.com>) (accessed 2025-11-07).
- [4] Innovation Graph: License Rankings - Global Metrics (online), available from (<https://innovationgraph.github.com/global-metrics/licenses>) (accessed 2025-11-17).
- [5] Kahol, A., Tummepalli, A.C. and Anish, P.R.: OSS-LCAF: Open-Source Software License Conflict Analysis Framework, *Proc. ICSE2025*, pp.303–314 (2025).
- [6] Kapitsaki, G.M. and Paschalides, D.: Identifying Terms in Open Source Software License Texts, *Proc. APSEC2017*, pp.540–545 (2017).
- [7] Ke, Q., Hou, X., Zhao, Y. and Wang, H.: ClauseBench: Enhancing Software License Analysis with Clause-Level Benchmarking, *Proc. ICSE2025*, pp.255–266 (2025).
- [8] Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., Zhou, X., Wang, E. and Dong, X.: Better Zero-Shot Reasoning with Role-Play Prompting, *Proc. NAACL2024*, Vol.1, pp.4099–4113 (2024).
- [9] Llama team: The Llama 3 Herd of Models (2024), available from (<https://arxiv.org/abs/2407.21783v3>).
- [10] OpenAI: GPT-4o System Card (2024), available from (<https://arxiv.org/abs/2410.21276>).
- [11] Wang, W., Hosseini, P. and Feizi, S.: Chain-of-Defensive-Thought: Structured Reasoning Elicits Robustness in Large Language Models against Reference Corruption (2025), available from (<https://arxiv.org/abs/2504.20769v1>).
- [12] Xu, S., Gao, Y., Fan, L., Liu, Z., Liu, Y. and Ji, H.: LiDetector-

tor: License Incompatibility Detection for Open Source Software, *ACM Trans. Software Engineering and Methodology*, Vol.32, No.1, pp.22:1–22:28 (2023).



Tomohito Nagasaki received his B.E. degree from the University of Osaka, Japan, in 2025. He is currently pursuing the master's degree at the Graduate School of Information Science and Technology, The University of Osaka.



Wataru Otoda received his master's degree in information science and technology from the University of Osaka in 2025. His research interests were primarily in software supply chain and software license during his master's degree, and he is currently working on computing systems research and development at NTT, Inc.



Tetsuya Kanda received his master's degree and Ph.D. degree in information science and technology from the University of Osaka, Japan in 2013 and 2016, respectively. Immediately after graduation he served as a Postdoctoral Fellow at Nara Institute of Science and Technology. He joined the University of Osaka in 2017

and was appointed Assistant Professor from 2018 to 2024. He is currently an Associate Professor at Notre Dame Seishin University. His research interests include software evolution, software documentation, and software ecosystem analysis.



Katsuro Inoue received his Dr. Eng. degree from the University of Osaka, Japan, in 1984. He joined the University of Osaka in 1984 and served as a Professor from 1995 to 2022. He was an Assistant Professor at the University of Hawaii at Manoa from 1984 to 1986. He is currently a Professor at Ritsumeikan University.

His research interests include software engineering, particularly software maintenance, software reuse, empirical software engineering, program analysis, and software bill of materials (SBOM).



Yuki Manabe received his Ph.D. degree in Information Science from the University of Osaka, Japan, in 2011. He was a Specially Appointed Assistant Professor at the University of Osaka from 2011 to 2013 and an Assistant Professor at Kumamoto University from 2013 to 2020. Since 2020, he has been a Lecturer at The

University of Fukuchiyama. His research interests include software engineering, open source software licenses, repository mining, software reuse, programming education, and software bills of materials (SBOMs).



Yoshiki Higo received his master's degree and Ph.D. degree in information science and technology from the University of Osaka in 2004 and 2006, respectively. At present, he is a professor at the University of Osaka. His research interests include mining software repositories, program analysis, and automated program repair.