

オープンソースライセンスの権利・義務の分析へ向けた 大規模言語モデルの適用可能性

長崎 智人[†] 音田 渉[†] 神田 哲也^{††} 眞鍋 雄貴^{†††} 井上 克郎^{††††}
肥後 芳樹[†]

[†] 大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

^{††} ノートルダム清心女子大学情報デザイン学部情報デザイン学科 〒700-8516 岡山県岡山市北区伊福町 2-16-9

^{†††} 福知山公立大学情報学部情報学科 〒620-0886 京都府福知山市市堀 3370

^{††††} 南山大学理工学部ソフトウェア工学科 〒466-8673 愛知県名古屋市昭和区山里町 18

E-mail: [†]{t-nagask,wa-otoda,higo}@ist.osaka-u.ac.jp, ^{††}kanda@m.ndsu.ac.jp, ^{†††}manabe-yuki@fukuchiyama.ac.jp,
^{††††}katsuroinoue@gmail.com

あらまし 近年、ソフトウェア開発のコスト削減のため、自由な利用・改変・再配布を許可するオープンソースソフトウェア（OSS）の利用が拡大している。OSSはオープンソースライセンスに利用条件を明記しており、その遵守が必須である。しかし、ライセンスの遵守には時間と労力を要するため、効率的な分析手法が求められている。本研究では、ライセンスが明確に定める利用者の権利・義務について、その推定に大規模言語モデル（LLM）を適用して精度を調査した。その結果、適合率では既存研究を下回る一方、再現率では上回ったことから、LLMは主に義務・禁止の推定に適した性質を持つことを明らかにした。さらに、ライセンス文書の長さとの関係、判定項目ごとの精度差を分析した。

キーワード オープンソースライセンス, 大規模言語モデル (LLM), 権利条項・義務条項, 権利・義務の推定

1. ま え が き

近年、ソフトウェア開発のコスト削減のため、自由な利用、改変、再配布を許可するオープンソースソフトウェア（OSS）の利用が広がっている。OSSはオープンソースライセンス（以下、ライセンス）と呼ばれる文書に利用条件を明記しており、OSSを利用する際はこのライセンスの遵守が必須である。

しかし、ライセンスの中には長文で複雑な条件を持つライセンスや、使用される頻度が低いため二次情報が乏しいライセンスなど、正しく解釈するのが困難なライセンスが存在する。さらに、複数のOSSを利用する場合は、各ライセンスを組み合わせる用いることが可能であるかを検証する必要がある。この性質をライセンスの互換性と呼ぶ。このような理由から、ライセンスの遵守には時間と労力を要する。そこで、開発者がライセンスを効率的に分析できるようにし、ライセンスの遵守を支援する手法の導入が求められる。

FOSSA社の運営するtldrLegal[1]というウェブサイトは、各ライセンスが定める権利・義務を分かりやすくリスト化して公開している。ここで、本研究では、ライセンスが規定する行為を「条項」と呼び、なかでも許可・禁止の対象となる条項を「権利条項」、義務付けの対象となる条項を「義務条項」と呼ぶ。tldrLegalでは23の条項を定義し、それぞれに“Can”（許可），“Cannot”（禁止），“Must”（義務）を割り当てることにより権利・義務を整理している。

tldrLegalで行われている権利・義務の判定を自然言語処理を用いて行う先行研究であるLiDetector[2]は、FOSS-LTE[3]を含む複数の自然言語処理手法と比較して、優れた判定精度を持つことを報告している。

一方、近年急速に発展している大規模言語モデル（LLM）は、ディープラーニング技術に基づく高度な自然言語処理モデルであり、ライセンス文書の解釈にも応用が期待できる。

本研究では、ライセンスの権利・義務の推定にLLMを適用し、その精度を調査した[4]。具体的には、ライセンスごとに、tldrLegalで定義されている23の条項が“Can”（許可），“Cannot”（禁止），“Must”（義務）のそれぞれに該当するか否かをLLMに質問した。対象とするモデルはLlama 3.3 70B, Llama 3.1 8B, GPT-4o, GPT-4o miniの4つであり、評価には適合率、再現率、F1スコアを用いた。

2. 背 景

本章では、オープンソースライセンスの遵守における課題について説明したのち、tldrLegal, 先行研究, 大規模言語モデルについて説明する。

2.1 オープンソースライセンス

ソフトウェア開発のコスト削減には、既存のプログラムの再利用が効果的である。ソフトウェアには著作権が適用されるため、著作権者の許可なく第三者が利用することはできないが、一部のソフトウェアは利用条件を記載したドキュメン

Can	Commercial Use, Modify, Distribute, Sublicense, Place Warranty, Private Use, Use Patent Claims
Cannot	Hold Liable, Use Trademark
Must	Include Copyright, Include License, State Changes, Include Notice

表 1: tldrLegal による Apache License 2.0 の条項リスト [1]

トである「ソフトウェアライセンス」を含んでおり、その条件を遵守する場合に限りソフトウェアを利用できる。

ソフトウェアの自由な利用、改変、再配布を許可するソフトウェアライセンスを「オープンソースライセンス」と呼び、オープンソースライセンスのもとで公開されたソフトウェアを「オープンソースソフトウェア (OSS)」と呼ぶ。Open Source Initiative (OSI) は、オープンソースソフトウェアを 10 の条件により厳密に定義している [5]。

2.2 オープンソースライセンスの遵守における課題

OSS の利用にはオープンソースライセンスの遵守が必須であるが、ライセンスを正しく理解し、遵守するには多くの課題が存在する。まず、ライセンスの理解には法的知識やソフトウェアの利用形態に関する知識が必要であるため、詳細で複雑なライセンスを独力で正確に解釈する作業は負担が大きい。加えて、GPL (GNU General Public License) や Apache License のような主流ライセンスとは対照的に、利用頻度の低いライセンスや開発者による独自のライセンスは、第三者による解説などの二次情報が不足していることが多い。また、複数の OSS を利用する場合は、各ライセンスを組み合わせることが可能であるかを検証する必要がある。この性質をライセンスの互換性と呼ぶ。

このように、ライセンスの遵守には時間と労力が必要であるが、ソフトウェア開発の後期にライセンス違反が発覚し、多くの依存関係を持つコンポーネントの変更を余儀なくされる事態を避けるため、開発初期段階でライセンス違反の可能性を検出する必要がある [6]。そのため、ライセンス遵守の困難さは大きな問題である。したがって、開発者がライセンスを効率的に分析し、遵守を支援する手段の導入が求められる。

2.3 tldrLegal によるライセンスの権利・義務の判定

ライセンスは、「商用利用を許可する」や「著作権表示を記載しなければならない」など、利用者の権利・義務を具体的に定める。本研究では、「商用利用」「著作権表示の記載」のような、ライセンスが規定する行為を「条項」と呼ぶ。

FOSSA 社の運営する tldrLegal というウェブサイトは、ライセンスごとに権利・義務を整理したリストを提供している [1]。tldrLegal では以下に示す 23 の条項が定義されており、ライセンスごとに 23 条項を “Can” (許可), “Cannot” (禁止), “Must” (義務) に分類している。

- (1) Commercial Use, (2) Modify, (3) Distribute, (4) Sublicense, (5) Use Trademark, (6) Hold Liable, (7) Private Use, (8) Place Warranty, (9) Use Patent Claims, (10) Relicense, (11) Statically Link, (12) Include Copyright,

- (13) Include License, (14) Include Notice, (15) Disclose Source, (16) State Changes, (17) Include Original, (18) Give Credit, (19) Rename, (20) Contact Author, (21) Include Install Instructions, (22) Compensate for Damages, (23) Pay Above Use Threshold

表 1 に tldrLegal が提供する Apache License 2.0 の条項リストを例示する。例えば、表 1 中の “Can” の列には “Commercial Use” が含まれているが、これは Apache License 2.0 が利用者に対しソフトウェアの商用利用を明示的に許可することを表す。

先行研究 [2], [3] に従い、tldrLegal の 23 条項を「権利条項」(上記の 1 から 11 の条項) と「義務条項」(上記の 12 から 23 の条項) の 2 つに分類する。権利条項は主に許可または禁止の対象となる行為であり, “Can” (許可) または “Cannot” (禁止) が割り当てられる。義務条項は主に義務として規定される行為であり, “Must” (義務) が割り当てられる。

また、先行研究 [2] では別の先行研究 [3], [7] をもとに、各条項を説明する概要を与えている。

2.4 オープンソースライセンス分析に関する先行研究

自然言語処理を用いてオープンソースライセンスを分析する手法として、ライセンスの互換性を検証する手法、およびツールである LiDetector [2] が挙げられる。LiDetector が互換性検証を行うプロセスには、ライセンスの権利・義務の推定を行うステップが含まれる。このステップでは、機械学習ベースの手法により tldrLegal で定義された 23 条項に該当する文をライセンスから検出し、検出された文から確率的文脈自由文法を用いて権利 (“Can”, “Cannot”) と義務 (“Must”) を推定する。この二つのステップで得られた結果を利用すると、任意のライセンスに対し表 1 のような条項のリストを生成できる。

LiDetector はライセンスの権利・義務の判定について他の自然言語処理手法より優れた精度を計測しており、**69.70%** の再現率、**85.88%** の適合率、**76.95%** の F1 スコアで検出可能である。

2.5 大規模言語モデル (LLM) の利用

大規模言語モデル (LLM) は、大規模なテキストデータを用いて学習したディープラーニング技術に基づく高度な自然言語処理モデルであり、テキスト生成や分類を含む様々な自然言語処理タスクを実行する。2017 年に LLM の中心技術である Transformer [8] が発表されて以降、LLM の開発は急速に広がり、パラメータ数や学習データの規模の拡大とともに高精度なモデルが開発され続けている [9]。代表的なモデルとして、OpenAI 社の GPT [10], Google 社の Gemini [11], Meta 社の Llama [12] などが挙げられる。

LLM を利用したオープンソースライセンス分析に関する研究は我々の知る限りまだ存在が確認できていないため、その適用可能性を確かめる必要がある。本研究では、ライセンスの権利・義務の判定に LLM を適用し、その適用可能性を調べる。

3. 実験方法

ライセンスを入力として、表 1 に例示するような権利・義務

system: You will receive a license document.

Answer if the license states that the licensee **can use the software for commercial purposes.**
(**Permission** of "Commercial Use")

Respond exclusively with "Yes" or "No".

user: Copyright (c) 4-digit year, Company or Person's Name

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS ...

図 1: "Commercial Use" が許可されているか否かを問うプロンプトの例

の判定を行うタスクを、本研究では「権利・義務の推定」と呼ぶ。本研究では、LLM の適用可能性を明らかにすることを目的に、LLM を用いたライセンスの権利・義務の推定の精度を評価する。

3.1 権利・義務の推定手法

対象のライセンスについて、tl;drLegal で定義された 23 条項が "Can", "Cannot", "Must" のそれぞれに該当するか否かを、以下に述べる方法で判定する。

権利条項: 「ライセンスが該当の権利条項を明示的に許可しているか否か」と、「ライセンスが該当の権利条項を明示的に禁止しているか否か」の 2 つを LLM に判定させる。許可している場合はその権利条項に "Can" を割り当て、禁止している場合は "Cannot" を割り当てる。2 つの判定は独立に行うため、1 つの権利条項に "Can", "Cannot" の両方を割り当てる可能性がある。

義務条項: 「ライセンスが該当の義務条項を明示的に義務付けているか否か」を LLM に判定させる。義務付けている場合はその義務条項に "Must" を割り当てる。

11 の権利条項と 12 の義務条項が存在するため、1 つのライセンスにつき 34 種類の判定を行う。この 34 種類の判定を、以降は判定項目と呼ぶ。

3.2 プロンプト

プロンプトとは LLM に与える質問である。本研究では、1 つのライセンスに対する 1 つの判定項目につき、1 つのプロンプトを与える。

プロンプトはシステムプロンプトとユーザプロンプトの 2 つの要素からなり、システムプロンプトの後ろにユーザプロンプトを結合し LLM に与える。

システムプロンプトでは、はじめにライセンス文を与える

モデル名	モデル仕様 (パラメータ数/ コンテキスト長)	リリース日 年-月-日	MMLU の 評価値 (%)
Llama3.3 70B	700 億 / 128k	2024-12-06	86
Llama3.1 8B	80 億 / 128k	2024-07-23	71
GPT-4o	- / 128k	2024-05-13	86
GPT-4o mini	- / 128k	2024-07-18	82

表 2: 対象としたモデルの情報

旨を記述し、その後に権利・義務がライセンスに記載されているか否かを問う。曖昧な回答を避けるため、"Yes", "No" のみで回答するように指示する。

ユーザプロンプトでは、対象とするライセンスの全文を与える。

例えば、"Commercial Use" が許可されているか否かを問うプロンプトは図 1 の通りである。他の判定を行う質問は、図 1 の赤いハイライト部分に "can/Permission", "cannot/Prohibition", "must/Obligation" のうち該当するものを代入し、図の黄色のハイライト部分に該当する条項名および先行研究 [2] による概要をそのまま代入したものである。ただし、"Statically Link" の概要は文法上そのまま代入することができないため、概要を次のように変更した: "compile the library into the program linked at compile time rather than runtime."

3.3 対象の LLM

対象としたモデルとその詳細を表 2 に示す。本研究では、幅広い性能のモデルの推定精度を調べるために複数のモデルを使用した。Llama は機密保持が必要な場面での利用に適した、ローカル環境で実行できる比較的小規模なモデルである。GPT は、実験時点で高い性能を示すモデルとされている [11]。精度の参考指標として、LLM の性能を評価する代表的なベンチマークである MMLU [13] による評価を記載している。

3.4 評価用データセット

本研究のデータセットの要件は、オープンソースライセンスであることと、すでに条項の判定がなされており正解として利用できることの 2 つである。それらを踏まえ、tl;drLegal のデータベースの中から OSI によってオープンソースライセンスと認定されていることを示す OSI-Approved タグが付けられた 58 のライセンスを評価用データセットとした。ただし、PHP License 3.0.1 は権利・義務の判定が行われていなかったため除外した。

オープンソースライセンスはその二次情報も含めて Web 上で容易に入手可能な文書であり、LLM の学習時にも用いられている可能性が高い。よって、ライセンス文書に含まれるライセンス名や団体名等の固有名詞は推定精度に影響を与える可能性がある。そのため、文中に含まれる固有名詞を、文中での意味を損なわないように配慮しつつ、例えばアルファベットを 1 文字ずつずらすなどの方法で別の文字列に置き換えた。

3.5 評価方法

LLM が質問に対してどれだけ正確に回答できるかを、精度

指標を用いて評価する。本研究では、先行研究 [2] と同様に「適合率」「再現率」「F1 スコア」の 3 つの指標を用いた。それぞれの指標と、関連する用語の定義は次の通りである。

正例, 負例: 正解もしくは回答が “Yes” に分類される質問を正例, “No” に分類される質問を負例とする。

True Positive (TP) / True Negative (TN): 実際に正例/負例であり、かつモデルが正しく正例/負例と判定したものの数。

False Positive (FP) / False Negative (FN): 実際には負例/正例であるが、モデルが誤って正例/負例と判定したものの数。

適合率 (Precision): モデルが正例と予測したもののうち、実際に正例である割合。

再現率 (Recall): 実際に正例であるもののうち、モデルが正しく正例と予測した割合。

F1 スコア (F1 Score): 適合率と再現率の調和平均。適合率と再現率は多くの場合トレードオフの関係にあり、F1 スコアはその両方を考慮した評価指標である。

3.6 その他の分析項目

評価指標に加え、後述する 2 つの分析を行う。

3.6.1 ライセンス文書の長さ精度の関係

本研究の評価用データセットに含まれるライセンスのトークン数は 43~9,388 トークンの範囲であり、プロンプトはそれぞれのモデルのコンテキスト長も超えない。しかし、ライセンス文書が長文になるほど判定する条項以外の多くの情報を含むため、長文のライセンスに対する推定精度の低下が懸念される。

そこで、ライセンスのトークン数と評価指標のスコアの間には負の線形関係があると仮定し、標本相関係数を計算する。標本相関係数は、2 つの変数間の線形関係の強さと方向を示す指標である。標本相関係数 r は、-1 から 1 の範囲で表され、1 に近いほど強い正の線形関係、-1 に近いほど強い負の線形関係があることを意味する。標本相関係数 r は以下の式で求められる。

$$r = \frac{\sum (Tokens_i - \overline{Tokens})(Score_i - \overline{Score})}{\sqrt{\sum (Tokens_i - \overline{Tokens})^2} \sqrt{\sum (Score_i - \overline{Score})^2}}$$

ただし、 i はライセンスに割り振られた番号、 $Tokens_i$ はライセンス i のトークン数、 \overline{Tokens} はトークン数の平均、 $Score_i$ はライセンス i のスコア、 \overline{Score} はスコアの平均を表す。

3.6.2 判定項目ごとの推定精度の差

判定対象となる条項ごとに、定義の明確さや理解しやすさ、ライセンス文に直接的な表現で書かれるか否かなどに違いがあるほか、許可・禁止・義務の間でも同様の差異が存在する。これらの理由から、34 の判定項目ごとに推定精度が異なる可能性が高いと考えられる。

判定項目ごとの推定精度の差を客観的に判断するために、LLM のモデルごとに仮説検定を行う。帰無仮説を “全ての判定項目の正答率が等しい”，対立仮説を “判定項目ごとの正答率に差がある” とし、帰無仮説を棄却することで質問ごとに推定精度の差があることを検証する。ただし、判定項目ごとの正例の割合やライセンスごとの差異は考慮しないものとする。

判定項目ごとに以下の p 値を求め、有意水準を下回る条項が 1 つでもあれば帰無仮説を棄却する。以下では、全ての判定項目で等しいと仮定する正答率を p (モデルの全体正答率)、ライセンス数を $n (= 58)$ 、各判定項目の正答数を x とおく。

上側 p 値: 正答数以上の正答が得られる確率。上側 p 値が有意水準を下回ると、モデルがその判定項目を高い正答率で判定すると推定できる。計算式を以下に示す。

$$P_{upper} = \sum_{i=x}^n nC_i p^i (1-p)^{n-i}$$

下側 p 値: 正答数以下の正答が得られる確率。下側 p 値が有意水準を下回ると、モデルがその判定項目を低い正答率で判定すると推定できる。計算式を以下に示す。

$$P_{lower} = \sum_{i=0}^x nC_i p^i (1-p)^{n-i}$$

一般的な有意水準は 5 % であるが、複数回の検定を行うことで生じる誤検出率を制御するためにボンフェローニ補正 [14] を採用し、5/34 (≈ 0.147) % を下回る p 値が 1 つでも現れれば帰無仮説を棄却する。

4. 実験結果

4.1 評価指標による精度

評価指標による精度を図 2 に示す。ただし、LiDetector の精度は先述の評価用データセットで評価したものではなく、論文 [2] に記載された値であることを注意する。

LLM の評価指標を比較すると、全てのモデルで再現率が適合率を大幅に上回った。LiDetector と比較すると、LLM の再現率は全てのモデルで LiDetector を上回ったが、適合率は LiDetector を下回った。F1 スコアは、Llama3.3 70B と GPT-4o が LiDetector に近い値を示したものの、すべてのモデルで LiDetector を下回る結果となった。

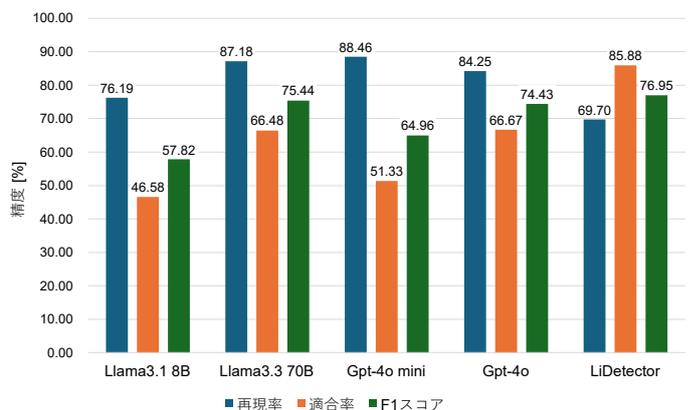


図 2: LLM と LiDetector の精度

4.2 ライセンスのトークン数と精度の関係

ライセンスのトークン数と精度の標本相関係数を表 3 に示

モデル	標本相関係数		
	再現率	適合率	F1 スコア
Llama3.3 70B	0.3489	0.1282	0.3141
Llama3.1 8B	0.1348	-0.3393	-0.1112
GPT-4o	0.1108	0.0076	0.1250
GPT-4o mini	0.2457	-0.3613	-0.1928

表 3: トークン数と精度の間の標本相関係数

す。全てのモデル・評価指標について相関係数は -0.3613 から 0.3489 の範囲であり、またモデルと評価指標によって正の値と負の値の両方が現れた。したがって、ライセンスのトークン数と精度に負の線形関係は無い可能性が高く、ライセンス文書の長さが権利・義務の推定精度に影響を与えない可能性が高い。

4.3 判定項目ごとの推定精度

ライセンス・判定項目ごとの正解・不正解の結果を図 3 に示す。仮説検定の結果、以下に示すように、すべてのモデルの検定で帰無仮説が棄却された。

Llama3.3 70B は、“Modify”の許可、禁止の判定を含む 10 個の判定項目で上側 p 値が有意水準を下回った。また、“Use Trademark”の禁止の判定を含む 6 個の判定項目で下側 p 値が有意水準を下回った。

Llama3.1 8B は、“Modify”の許可の判定を含む 4 個の判定項目で上側 p 値が有意水準を下回った。また、“Place Warranty”の禁止の判定を含む 4 個の判定項目で下側 p 値が有意水準を下回った。

GPT-4o は、“Modify”の許可の判定を含む 8 個の判定項目で上側 p 値が有意水準を下回った。また、“Sublicense”の許可の判定を含む 4 個の判定項目で下側 p 値が有意水準を下回った。

GPT-4o mini は、“Commercial Use”の許可の判定を含む 10 個の判定項目で上側 p 値が有意水準を下回った。また、“Sublicense”の禁止の判定を含む 5 個の判定項目で下側 p 値が有意水準を下回った。

5. 考 察

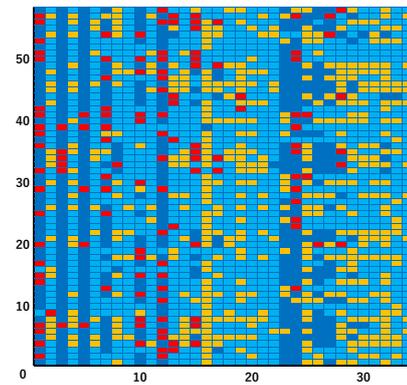
5.1 ライセンス文書の長さとの精度の関係

ライセンス文書の長さが権利・義務の推定精度に影響を与える場合、長文ライセンスの分割や一部削除など、ライセンス文書を短文化することによる推定精度の向上が期待できる。しかし、4.2 節でライセンス文書の長さが権利・義務の推定精度に影響を与えない可能性が高いことを示した。したがって、権利・義務の推定精度の向上には、ライセンス文書を短文化する以外のアプローチが必要であると考えられる。

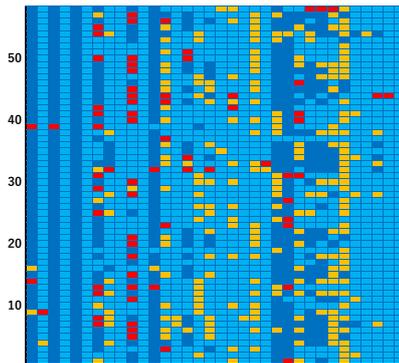
5.2 判定項目ごとの精度の差

4.3 節では、「34 の全ての判定項目の正答率が等しい」という仮説が全てのモデルで棄却されたことを述べた。これは、LLM が判定項目ごとに得意不得意があることを示している。

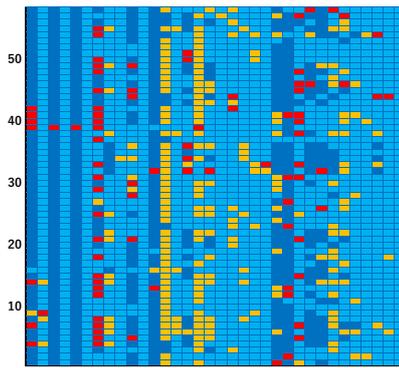
“Modify”と“Distribute”は、許可“Can”の判定について全てのモデルで有意に高い正答率であり、“Cannot”の判定についても複数のモデルで有意に高い正答率である。これらの判定項



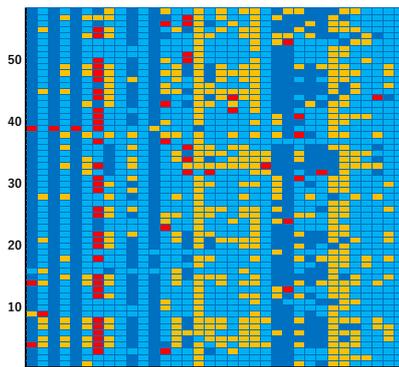
(a) Llama3.1 8B



(b) Llama3.3 70B



(c) GPT-4o



(d) GPT-4o mini

図 3: ライセンス・判定項目ごとの正解・不正解の対応グラフ
縦軸は 58 種類のライセンス、横軸は 34 種類の判定項目に対応する。判定項目は、(1) “Commercial Use”の許可、(2) “Commercial Use”の禁止、..., (34) “Pay Above Use Threshold”の義務、のように 2.3 節で示す順に左から並べている。青色および水色のセルは正解 (TP, TN)、黄色および赤色のセルは不正解 (FP, FN) を表す。

目は、実際の権利・義務判定でも有用であると考えられる。

一方、“Include Original”の“Must”、“Place Warranty”の“Can-not”の判定は3つのモデルで有意に低い正答率であり、推定精度の改善が求められる。

5.3 権利・義務の推定へのLLMの適用可能性

本研究では、LLMが再現率でLiDetectorを上回った。一方、適合率ではLiDetectorを下回った。実際のライセンス分析では、許可されていない行為を誤って許可されていると判断してはならない一方、定められた義務および禁止事項は多少の誤検出を許容してでも漏れなく検出できることが望まれる。ゆえに、LLMは主に義務・禁止の推定に適した性質を持っていると考えられる。

6. 妥当性への脅威

6.1 内的妥当性

プロンプトの表現や構造の違いがLLMの回答に影響を与えている可能性があり、結果の再現性に課題が生じる。また、判定項目ごとの推定精度の差の検定では、判定項目ごとの正例の割合の差やライセンスごとの差異を考慮していない点に注意を要する。

LLMの学習データにオープンソースライセンスが含まれている場合、精度が過大評価されている可能性がある。ただし、固有名詞の匿名化などの処理を施したため、これによる妥当性への脅威は小さいと考えられる。

6.2 外的妥当性

本研究は特定のデータセットに基づくため、他のライセンスで同様の精度が得られることは保証できない。また、LLMについても、他のモデルでは異なる結果が得られる可能性がある。実験対象のライセンスやLLMを追加し、より深く検証することは今後の課題である。

7. 今後の課題

7.1 プロンプトの改善

プロンプトの改善により、推定精度の向上が期待できる。例えば、推定の過程を明示的に出力させる技法であるChain-of-Thought（思考の連鎖）の利用は推定精度を向上させる可能性がある[15]。また、各条項の厳密な定義をプロンプト内で明記することにより、定義に基づいた正確な推定が可能になると考えられる。

7.2 ファインチューニング

ファインチューニングとは、学習済みモデルを追加学習させ、パラメーターを特定のタスクに特化するように調整する作業である。多数のライセンスと権利・義務のリストを学習データとするファインチューニングにより、権利・義務の精度向上が期待できる。

8. あとがき

本研究では、LLMを利用したオープンソースライセンスの権利・義務の推定精度を調査した。その結果、適合率では既存研究を下回る一方、再現率では上回ったことから、LLMは主

に義務・禁止の推定に適した性質を持つことを明らかにした。加えて、ライセンス文書の長さが精度に影響を与えた可能性が低いことと、判定項目ごとに推定精度の差があることについて考察した。最後に、今後の課題としてプロンプトの改善やファインチューニングによって推定精度の向上が見込めることを述べた。

LLMを活用したオープンソースライセンス分析の研究はまだ発展途上である。将来的には、権利・義務の推定にとどまらず、ライセンス違反の検出や互換性の検証など、より高度なタスクへの応用が期待される。

謝辞

本研究は、JSPS 科研費 JP24H00692, JP21K18302, JP21H04877, JP23K24823, JP22K11985, JP24K14895, JP21K02862, JP23K28065, 2024 年度南山大学パッへ研究奨励金 I-A-2 の助成を得て行われた。

文 献

- [1] FOSSA, “tldrlegal,” 2025. Accessed Jan. 2025. <https://www.tldrlegal.com>
- [2] S. Xu, Y. Gao, L. Fan, Z. Liu, Y. Liu, and H. Ji, “LiDetector: License incompatibility detection for open source software,” ACM Trans. Softw. Eng. Methodol., vol.32, no.1, pp.22:1–22:28, Feb. 2023.
- [3] G.M. Kapitsaki and D. Paschalides, “Identifying terms in open source software license texts,” Proc. APSEC2017, pp.540–545, 2017.
- [4] 長崎智人, “大規模言語モデルを用いたオープンソースライセンスの権利・義務の推定,” 令和6年度特別研究報告, 大阪大学, 2025.
- [5] Open Source Initiative, “The open source definition,” 2007. Accessed Jan. 2025. <https://opensource.org/osd>
- [6] 東裕之輔, “コンテナ開発におけるOSSの法的リスク特定自動化に関する研究,” 博士論文, 和歌山大学, 2023.
- [7] 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.
- [8] A. Vaswani, N.M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” Proc. NeurIPS2017, pp.5998–6008, 2017.
- [9] Y. Annapaka and D.P. Pakray, “Large language models: a survey of their development, capabilities, and applications,” Knowledge and Information Systems, pp.1–56, Dec. 2024.
- [10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018. Accessed on Jan. 2025. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
- [11] Artificial Analysis, “Independent analysis of AI models and API providers,” 2024. Accessed Jan. 2025. <https://artificialanalysis.ai>
- [12] Llama team, “The Llama 3 herd of models,” 2024. <https://arxiv.org/abs/2407.21783>
- [13] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, “Measuring massive multitask language understanding,” 2021. <https://arxiv.org/abs/2009.03300>
- [14] J.M. Bland and D.G. Altman, “Multiple significance tests: the Bonferroni method,” BMJ, vol.310, no.6973, p.170, 1995.
- [15] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff, P.S. Dulepet, S. Vidyadhara, D. Ki, S. Agrawal, C. Pham, G. Kroiz, F. Li, H. Tao, A. Srivastava, H.D. Costa, S. Gupta, M.L. Rogers, I. Goncarenco, G. Sarli, I. Galyner, D. Peskoff, M. Carpuat, J. White, S. Anadkat, A. Hoyle, and P. Resnik, “The prompt report: A systematic survey of prompting techniques,” 2024.