特別研究報告

題目

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

指導教員
肥後 芳樹 教授

報告者
長崎 智人

令和 7 年 2 月 6 日

大阪大学 基礎工学部 情報科学科

令和 6 年度 特別研究報告

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

長崎 智人

内容梗概

ソフトウェア開発のコストを削減するには既存のプログラムの再利用が効果的であるため，近年は自由な利用，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を遵守することが必須である．しかし，ライセンスの複雑さ，二次情報の不足，互換性の問題などの理由により，ライセンスの遵守には時間と労力を要する．迅速なライセンス違反検出を実現するために，開発者がライセンスを効率的に分析し，ライセンスの遵守を支援する手段の導入が求められる．

FOSSA 社の運営する tl;drLegal というウェブサイトは，各ライセンスが定める権利・義務を明快にリスト化して公開している．本研究では，ライセンスが規定する行為を「条項」と呼ぶ．tl;drLegal 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．

tl;drLegal 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 は，FOSS-LTE 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，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，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，オープンソースライセンスの権利・義務の判定タスクに LLM を適用し，その精度を調査した．Llama 3.3 70B，Llama 3.1 8B，GPT-4o，GPT-4o mini の 4 つのモデルを対象とした．

実験の結果，再現率においては LiDetector を上回る一方，適合率では下回ることが明らかとなった．加えて実験結果を分析し，ライセンス文書の長さと推定精度に線形関係が無い可能性と，判定項目ごとに推定精度の差がある可能性を考察した．また，以上の結果を受けて，権利・義務の推定への LLM の利用可能性と，推定精度を向上させる方法について考察した．

主な用語

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

目 次

1　まえがき

ソフトウェア開発において，全てのコンポーネントを一から開発するためには非常に高いコストがかかるため，既存のプログラムの再利用が効果的である．そのような理由により，近年は自由な利用，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を遵守することが必須である．

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に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，各ライセンス間の互換性を検証する必要がある．ライセンスの互換性とは，複数のライセンスを組み合わせて用いることが可能であるかを指す性質のことである．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．ソフトウェア開発においては，開発初期段階でライセンス違反の可能性を検出することが求められるため [22]，ライセンス遵守の困難さは重要な課題である．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．

ライセンスは，利用者に与えられる権利，課される義務を明確に定めている．FOSSA 社の運営する tl;drLegal[7] 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．例えば，権利条項として “Commercial Use”（商用利用），義務条項として “Include Copyright”（著作権表示の記載）が挙げられる．tl;drLegal 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．

tl;drLegal 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が存在する．例えば，ライセンスの互換性を検証する手法である LiDetector[21] は，その互換性検証のプロセスの中に，権利・義務を推定するステップを含んでいる．この推定において，LiDetector は FOSS-LTE[10] を含む他の自然言語処理手法より優れた精度を発揮することが報告されている．

一方，近年急速に発展している大規模言語モデル（LLM）は，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，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．具体的には，ライセンスごとに，tl;drLegal 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．

本論文の構成は以下の通りである．第 2章では研究の背景について詳細に説明する．第 3章では実験方法について述べる．第 4章では実験結果について述べる．第 5章では実験結果に対する考察を述べる．第 6章では妥当性への脅威を述べる．第 7章では今後の課題を述べる．

2　背景

本章では，本研究のモチベーションとなるオープンソースライセンスの遵守に関する課題について説明したのち，tl;drLegal，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

ソフトウェア開発において，全てのコンポーネントを一から開発するためには非常に高いコストがかかるため，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．ただし，一部のソフトウェアは，利用条件を記載したドキュメントである「ソフトウェアライセンス」を含んでおり，利用者は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．

ソフトウェアライセンスの中で，ソフトウェアの自由な利用，改変，再配布を許可するものを「オープンソースライセンス」と呼ぶ．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は「オープンソースソフトウェア（OSS）」と呼ばれる．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 [9].

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．しかし，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．本節では，以下の 3 つの視点からライセンス遵守の困難さを説明する．

ライセンス文書の複雑さ

オープンソースライセンスには，MIT License のように数文程度の簡潔なものから，GPL（GNU General Public License）のように数千語に及ぶ詳細なものまで，多様な種類が存在する．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，特に GPL のような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．

二次情報の不足

GPL や Apache License などの著名で広く使用されているライセンスは，第三者による解説や，後述する互換性の情報など，豊富な二次情報が存在する．これらの情報はライセンスを正しく解釈する上で有用である．一方，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二次情報が不足していることが多く，そのようなライセンスを正しく解釈することは困難である．

互換性の問題

ソフトウェアライセンスにおける「互換性」とは，複数のライセンスが矛盾なく共存できる性質を意味する．非互換なライセンスを持つソフトウェア同士を組み合わせた場合は，その両方を遵守できずライセンス違反となる．多数のソフトウェアを利用する場合は特に互換性の問題に留意する必要がある．二次情報が豊富なライセンス同士であれば，それらの互換性についても情報が得られる可能性があるが，二次情報が不足しているライセンスは，互換性の判断がより困難である．

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．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 [22]．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 tl;drLegal によるライセンスの権利・義務の判定

ライセンスは，利用者に許可する権利や利用者が守らなければいけない義務を具体的に定める．例えば,「商用利用を許可する」や「再配布をする際には著作権表示を記載しなければならない」などが記載されている．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 tl;drLegal 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 [7]．tl;drLegal では表 1，表 2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” ，“Cannot”，“Must”に分類している．ライセンスが条項を明示的に許可している場合は “Can”，明示的に禁止している場合は “Cannot”，明示的に義務付けている場合は “Must” に分類する．

図 1に tl;drLegal が提供する Apache License 2.0 のリストを例示する．図 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していることを表す．“Cannot” の列には “Hold Liable” が含まれているが，これは著作者の法的責任を問う権利が認められていない（禁止されている）ことを表す．“Must” の列には “Include Copyright” が含まれているが，これは利用時の著作権表示を明示的に義務付けていることを表す．

先行研究 [21][10] に従い，tl;drLegal の 23 条項を表 1の「権利条項」と，表 2の「義務条

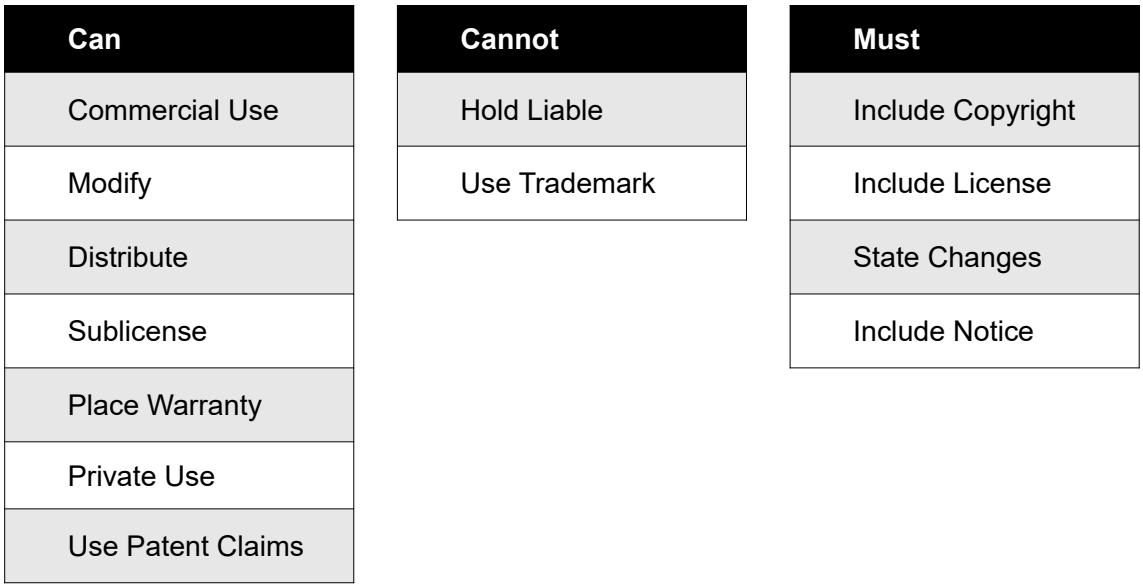


図 1: tl;drLegal が提供する Apache ライセンス v.2.0 の条項リスト [7]

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．ごくまれに，権利条項に “Must” が割り当てられるなどの例外も存在するが，本研究では主要なケースに焦点を当てる．

2.4 オープンソースライセンス分析に関する先行研究

自然言語処理によるオープンソースライセンスの分析手法として，Xu らの LiDetector[21] が挙げられる．LiDetector は，自然言語処理を利用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 が互換性検証を行うプロセスには,「条項の検出」および「条項に対する権利・義務の推定」のステップが含まれる.「条項の検出」のステップでは，機械学習ベースの手法により，表 1，表 2の 23 条項に該当する文をライセンスから検出する.「条項に対する権利・義務の推定」のステップでは，検出された文から，権利（“Can”，“Cannot”）と義務（“Must”）を確率的文脈自由文法を用いて推定する．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図 1のような条項のリストを生成できる．

同論文 [21] では，異なる複数の手法を用いた「条項の検出」および「条項に対する権利・義務の推定」の精度が調査されている．

表 3は手法ごとの「条項の検出」の評価を示す．(A1) 正規表現による手法 [2]，(A2) 意味的類似性による手法 [11]，(A3) FOSS-LTE[10]，(A4) LiDetector が比較されており，LiDetector が再現率，適合率，F1 スコアの項目で最も優れた値を示している．

条項名	概要
Distribute	Distribute original or modified derivative works
Modify	Modify the software and create derivatives
Commercial Use	Use the software for commercial purposes
Relicense	Add other licenses with the software
Hold Liable	Hold the author responsible for subsequent impacts
Use Patent Claims	Practice patent claims of contributors to the code
Sublicense	Incorporate the work into something that has a more restrictive license
Statically Link	The library can be compiled into the program linked at compile time rather than runtime
Private Use	Use or modify software freely without distributing it
Use Trademark	Use contributors' names, trademarks, or logos
Place Warranty	Place warranty on the software licensed

表 1: tl;drLegal で判定対象である権利条項の一覧 [21]

表 4は手法ごとの「条項に対する権利・義務の推定」の評価を示す．(B1) 正規表現による手法 [17]，(B2) Stanford Sentiment Treebank (SST) を利用した手法 [16]，(B3) FOSS-LTE, (B4) LiDetector が比較されており，LiDetector が最も高い正答率を示している．

表 5はこれらの手法を組み合わせた一連の推定精度の評価を示している．LiDetector が再現率，適合率，F1 スコアの項目で最も優れた値を示している．

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[20] 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 [3].

現在では数多くのモデルやサービスが開発され，実際に利用されている．OpenAI 社は 2018 年の GPT-1[14] の開発を皮切りに，複数のモデルやサービスを開発しており，GPT シリーズの最新モデルである GPT-4o[12] や 2024 年 12 月に完全版が発表された OpenAI o1[13] は複数のベンチマークで非常に高い評価を受けている [1]．Gemini は Google が開発した代表的なモデルであり，2023 年 12 月時点で Gemini Ultra が人間の専門家を上回るパフォーマ

条項名	概要
Include Copyright	Retain the copyright notice in all copies or substantial uses of the work
Include License	Include the full text of the license in modified software
Include Notice	Include that NOTICE when you distribute if the library has a NOTICE file with attribution notes
Disclose Source	Disclose your source code when you distribute the software and make the source for the library available
State Changes	State significant changes made to the software
Include Original	Distribute copies of the original software or instructions to obtain copies with the software
Give Credit	Give explicit credit or acknowledgment to the author with the software
Rename	Change software name as to not misrepresent it as the original software
Contact Author	Get permission from the author or contact the author about the module you are using
Include Install Instructions	Include the installation information necessary to modify and reinstall the software
Compensate for Damages	Compensate the author for any damages caused by your work
Pay Above Use Threshold	Pay the licensor after a certain amount of use

表 2: tl;drLegal で判定対象である義務条項の一覧 [21]

手法	再現率	適合率	F1 スコア
(A1) 正規表現	40.06	77.55	52.83
(A2) 意味的類似性	66.95	87.47	75.85
(A3) FOSS-LTE	72.64	62.07	66.94
(A4) LiDetector	**75.70**	**93.28**	**83.58**

表 3: 条項の検出精度 [21]

手法	正解率
(B1) 正規表現	81.27
(B2) SST	82.88
(B3) FOSS-LTE	82.71
(B4) LiDetector	**91.09**

表 4: 条項に対する権利・義務の推定精度 [21]

条項の検出手法	権利・義務の推定手法	再現率	適合率	F1 スコア
(A1) 正規表現	(B1) 正規表現	31.74	61.44	41.86
(A2) 意味的類似性	(B1) 正規表現	54.30	70.94	61.51
(A4) LiDetector	(B2) 正規表現	63.33	78.04	69.92
(A1) 正規表現	(B4) LiDetector	37.12	68.81	48.22
(A3) FOSS-LTE	(B3) FOSS-LTE	60.08	51.34	55.37
(A4) LiDetector	(B4) LiDetector	**69.70**	**85.88**	**76.95**

表 5: 権利・義務の推定の精度 [21]

ンスを示した [18]．Llama は Meta 社が開発するオープンソースの LLM である．2024 年 12 月にリリースされた，Llama3[19] の最新のモデルである Llama 3.3 は，オープンソースであるため経済的に低コストでありながら，優れた性能を発揮している [1]．他にも，Microsoft 社の Copilot，Anthropic 社の Claude[4]，DeepSeek 社の DeepSeek[6] など，数々のモデル，サービスが存在する．

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まだ存在が確認できていない[1]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．

[1]IEEE Xplore（https://ieeexplore.ieee.org/Xplore/home.jsp），arXiv（https://arxiv.org/）にて “LLM”, “license” の両方の検索ワードに該当する論文を対象とし，LLM を利用したライセンス分析を主要論点とする論文が無いことを確認 (2025 年 2 月 5 日時点).

3　実験手法

ライセンスを入力として，図 1に例示するような権利・義務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．本章では，LLM を用いてライセンスの権利・義務の推定を行い，その精度を評価する手法について説明する．

3.1　実験の目的

LLM を用いたライセンスの権利・義務の推定の精度を評価し，LLM の適用可能性を明らかにする．

3.2　権利・義務の推定手法

対象のライセンスについて，表 1，表 2の 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．権利条項と義務条項では手順が異なるため，それぞれに分けて説明する．

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”が割り当てられ，禁止している場合は “Cannot”が割り当てられる．2 つの判定は独立に行われるため，1 つの権利条項に “Can”，“Cannot”の両方が割り当てられる可能性があることに注意する．

義務条項

「ライセンスが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”が割り当てられる．

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とになる．この 34 種類の判定を，以降は判定項目と呼ぶ．

3.3　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が結合されて LLM に与えられる．

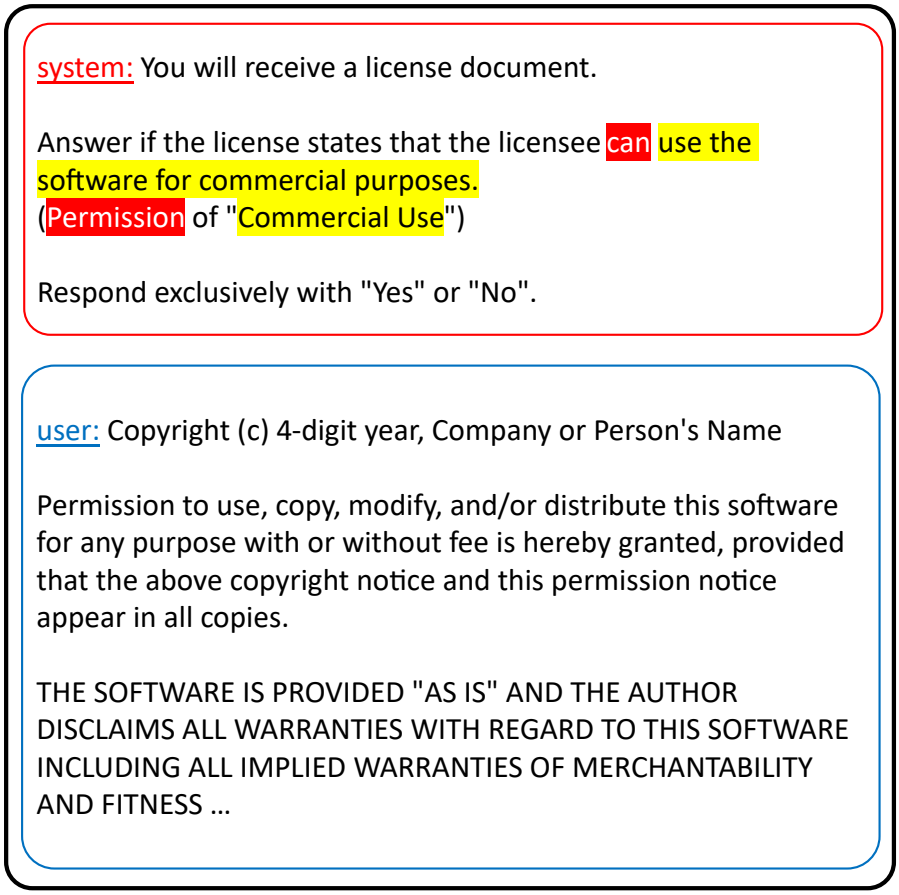


図 2: “Commercial Use” が許可されているか否かを問うプロンプトの例

システムプロンプトでは，はじめにライセンス文を与えられる旨を記述し，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 2の通りである．他の判定を行う質問は，図 2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表 1，表 2中の該当する条項名および概要をそのまま代入したものである．ただし，“Statically Link” は英語の文法の観点から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.”

3.4 対象の LLM

対象としたモデルとその詳細を表 6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．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 Llama3.1 8B，Llama3.3 70B，と，実験時点で高い性能を示すとされていた GPT-4o，GPT-4o mini の 4 つのモデルを対象とした [1]．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[8] による評価を記載している．

モデル名	パラメータ数（億個）	コンテキスト長（千トークン）	リリース日（年. 月. 日）	MMLU の評価（%）
Llama3.3 70B	700	128	2024.12.6	86
Llama3.1 8B	80	128	2024.7.23	71
GPT-4o	不明	128	2024.5.13	86
GPT-4o mini	不明	128	2024.7.18	82

表 6: 対象としたモデルの情報

3.5 評価用データセット

本研究で用いた評価用データセットについて説明する．

本研究の評価用データセットに対する要件を以下のように定義した．

1. オープンソースライセンスであること．

2. すでに条項の判定がなされており，Ground Truth（実際の正解）として利用できること．

以上を踏まえ，tl;drLegal のデータベースの中から OSI 認可というタグを付けられた 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，ライセンス文書中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別の文字列に置換して匿名化した．

3.6 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 [21] 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．計算式を以下に示す．

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

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．計算式を以下に示す．

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

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．

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.7 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．

3.7.1 ライセンス文書の長さと精度の関係

ライセンス文書の長さはライセンスごとに大きく異なる．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る．

本研究で用いるモデルのコンテキスト長は 128,000 トークンであり，この範囲を超えるプロンプトでは超過部分が切り捨てられ，情報の忘却が生じる．しかし，本研究のデータセットではライセンス文書の長さがコンテキスト長を大幅に下回るため，この問題は発生しない．

一方で，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んでいる可能性があるため，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，式中の記号の意味は以下の通りである．

r: 標本相関係数

i: ライセンスに割り振られた番号

$Tokens_i$: ライセンス i のトークン数

$\overline{Tokens}$: トークン数の平均

$Score_i$: ライセンス i のスコア

$\overline{Score}$: スコアの平均

3.7.2 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．そのため，条項ごとに推定難易度が異なる可能性がある．また，許可・禁止・義務の間でも推定のしやすさに差が生じている可能性がある．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．以下の通り，帰無仮説と対立仮説を設定し，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．

帰無仮説：全ての判定項目の正答率が等しい．

対立仮説：判定項目ごとの正答率に差がある．

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．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．

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．

$$P_{upper} = \sum_{i=x}^{n} {}_n\mathrm{C}_i p^i (1-p)^{n-i}$$

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．

$$P_{lower} = \sum_{i=0}^{x} {}_n\mathrm{C}_i p^i (1-p)^{n-i}$$

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 [5] を採用し，5/34（約 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

本章では，評価指標による精度，ライセンスのトークン数と精度の関係，判定項目ごとの推定精度をそれぞれ説明する．

4.1　評価指標による精度

評価指標による精度を図 3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 [21] に記載された値であることに注意する．

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．モデル間の比較では，再現率は GPT-4o mini，Llama3.3 70B，GPT-4o，Llama3.1 8B の順に優れていた．適合率は GPT-4o，Llama3.3 70B，GPT-4o mini，Llama3.1 8B の順に優れていた．F1 スコアは Llama3.3 70B，GPT-4o，GPT-4o mini，Llama3.1 8B の順に優れていた．

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．

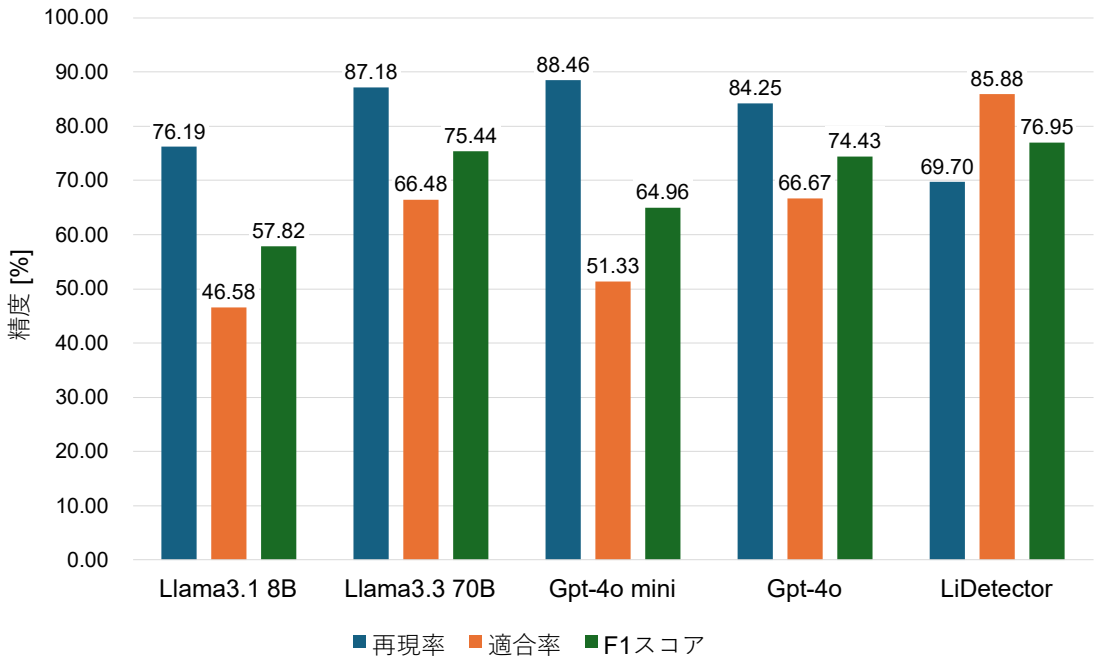


図 3: LLM と LiDetector の精度

4.2 ライセンスのトークン数と精度の関係

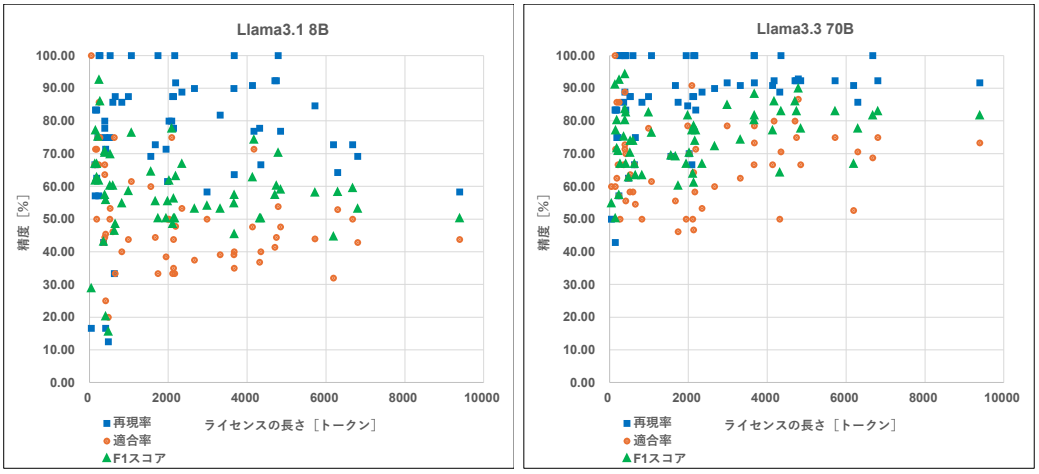
図 4にライセンスごとのトークン数と精度の散布図を示す．標本相関係数を表 7に示す．全てのモデル・評価指標について相関係数は −0.3613 から 0.3489 の範囲であり，またモデルと評価指標によって正の値と負の値の両方が現れた．したがって，ライセンスのトークン数と精度に負の線形関係は無い可能性が高く，ライセンス文書の長さが権利・義務の推定精度に影響を与えない可能性が高い．

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

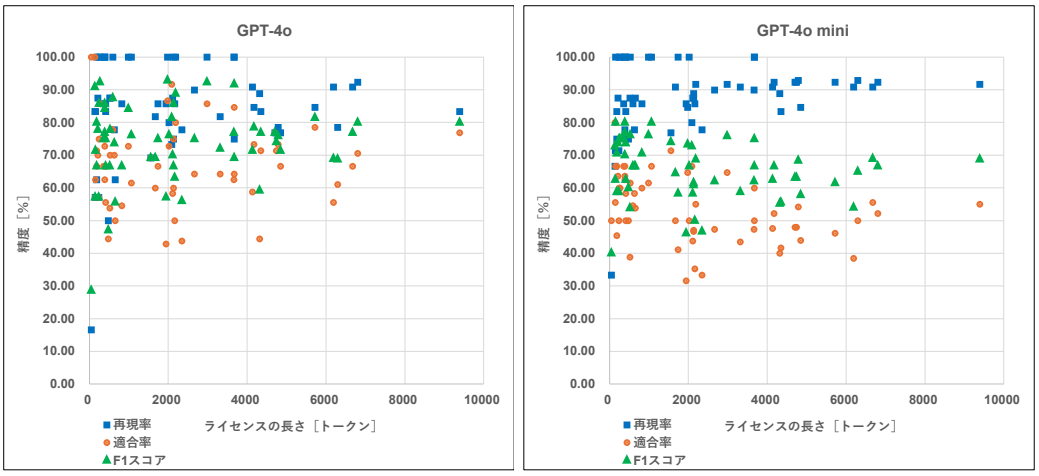
表 7: トークン数と精度の間の標本相関係数

4.3 判定項目ごとの推定精度

図 5は，ライセンス・判定項目ごとの正解・不正解を表した図である．横軸は 34 種類の判定項目を，縦軸は評価用データセットに含まれる 58 のライセンスを表す．図を見ると，判定項目ごとに正答率の偏りがあることがわかる．図 5bの Llama3.3 70B の例を挙げると，4 番目の判定項目（“Modify” の禁止）は全ライセンスで正解している一方，29 番目の判定項目（“Give Credit ” の義務）は不正解が他と比べて多く，正答数は 8 である．表 8に，有意水準を下回った判定項目をモデルごとに示す．すべてのモデルの検定で帰無仮説が棄却された．



(a) Llama3.1 8B　　(b) Llama3.3 70B

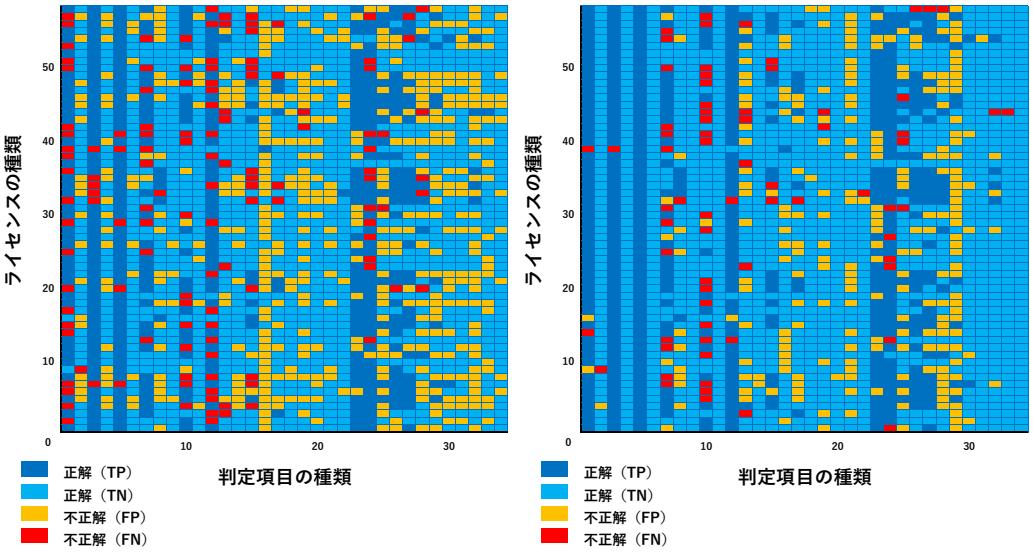


(c) GPT-4o　　(d) GPT-4o mini

図 4: ライセンス文書の長さ（トークン数）と精度の関係

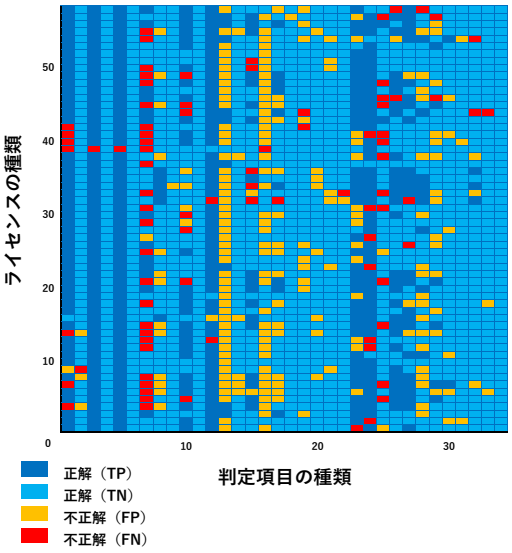
モデル	p 値が有意水準を下回った判定項目	
	上側 p 値	下側 p 値
Llama3.3 70B	“Modify”の許可，禁止	“Use Trademark”の禁止
	“Distribute”の許可，禁止	“Private Use”の許可
	“Use Trademark”の許可	“Statically Link”の許可
	“Hold Liable”の許可	“Include Notice”の義務
	“Relicense”の禁止	“Include Original”の義務
	“Contact Author”の義務	“Give Credit”の義務
	“Compensate for Damages”の義務	
	“Pay Above Use Threshold”の義務	
Llama3.1 8B	“Modify”の許可	“Place Warranty”の禁止
	“Distribute”の許可	“Relicense”の許可
	“Use Trademark”の許可	“Include Notice”の義務
	“Statically Link”の禁止	“Include Install Instructions”の義務
GPT-4o	“Modify”の許可，禁止	“Sublicense”の許可
	“Distribute”の許可，禁止	“Private Use”の許可
	“Use Trademark”の許可	“Place Warranty”の禁止
	“Hold Liable”の許可	“Include Original”の義務
	“Use Patent Claims”の禁止	
	“Pay Above Use Threshold”の義務	
GPT-4o mini	“Commercial Use”の許可	“Sublicense”の許可
	“Modify”の許可	“Place Warranty”の禁止
	“Distribute”の許可	“Statically Link”の許可
	“Use Trademark”の許可	“Include Original”の義務
	“Hold Liable”の許可，禁止	“Give Credit”の義務
	“Include License”の義務	
	“Disclose Source”の義務	
	“State Changes”の義務	
	“Pay Above Use Threshold”の義務	

表 8: 有意水準を下回った判定項目

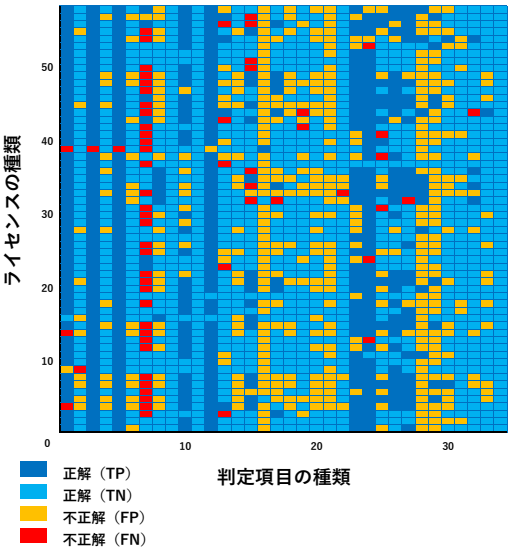


(a) Llama3.1 8B

(b) Llama3.3 70B



(c) GPT-4o



(d) GPT-4o mini

図 5: ライセンス・判定項目ごとの正解・不正解の対応グラフ

5　考察

本章では，実験結果に対する分析および，権利・義務の推定への LLM の適用可能性に対する考察を述べる．

5.1　ライセンス文書の長さと精度の関係

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．

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おり，既に高い正答率を示している判定項目は実際の権利・義務判定でも有用であると考えられる．一方，正答率の低い判定項目については，条項の厳密な定義やプロンプトの改善を行うことにより精度の向上が期待できる．ただし，判定項目ごとの正例の割合の差やライセンスごとの特徴が正答率に影響を与えている可能性があり，この点には注意が必要である．

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・禁止の推定で適用可能であると考えられる．

6　妥当性への脅威

本研究の結果の妥当性には，いくつかの脅威が存在する．本節では，それらを内的妥当性，外的妥当性の観点から説明する．

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，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

本章では，LLM による権利・義務の推定精度を向上させる手段について考察する．

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，本研究では出力を機械的に処理するために利用しなかったが，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．ただし，この厳密な定義を与えるためには専門的な知識が必要である．

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．しかし，権利・義務の正解のリストがすでに用意されているライセンスを大量に収集することは困難である．また，正解のリストを作成するには専門的な知識が必要である．

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，再現率においては既存研究を上回る一方，適合率では下回ることが明らかとなった．また，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，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考察で述べた．

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

大阪大学大学院情報科学研究科コンピュータサイエンス専攻 肥後 芳樹 教授には，中間報告会にて途中経過を発表し，議論する機会をいただき，研究および発表に関して多くの貴重なご助言を賜りました．肥後芳樹教授のご指導により本研究を進めることができました．心より深く感謝申し上げます．

大阪大学大学院情報科学研究科コンピュータサイエンス専攻 松下 誠 准教授には，中間報告会などにおいて研究に関する多くのご助言・ご指摘をいただきました．加えて，報告会後も時間を割いて研究に関連する知識について丁寧にご説明いただきました．心より深く感謝申し上げます．

大阪大学大学院情報科学研究科コンピュータサイエンス専攻 Kula Raula Gaikovina 教授には，中間報告会にて研究に関するご助言・ご指摘をいただきました．加えて，報告会後も時間を割いて疑問へのご回答，および研究に関する最新情報などをご紹介いただきました．心より深く感謝申し上げます．

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 神田 哲也 准教授には，毎週のミーティングをはじめ，多くの時間を割いて直接ご指導いただきました．研究の方針から細部に至るまで貴重な助言を賜るとともに，本論文の執筆および報告に関しても添削や相談にご協力いただきました．神田哲也准教授の多大なご支援により，本研究を進めることができました．心より深く感謝申し上げます．

南山大学理工学部ソフトウェア工学科 井上 克郎 教授には，定期的な研究の進捗報告および相談の機会をいただき，その場での議論を通じて研究を深めることができました．心より深く感謝申し上げます．

福知山公立大学情報学部情報学科 眞鍋 雄貴 講師には，定期報告などの場で研究に対する多くのご指摘・ご助言を賜りました．さらに，本研究に関わる過去の研究や基礎知識に関する論文を複数ご紹介いただきました．心より深く感謝申し上げます．

株式会社東芝デジタルイノベーションテクノロジーセンター 仇 実 氏には，定期報告などの場で貴重なご意見をいただきました．また，研究に関連する多くの情報をご教示いただき，研究分野に対する理解を深めることができました．心より深く感謝申し上げます．

最後に，本研究に関してあらゆるご助言・ご助力をくださった大阪大学大学院情報科学研究科コンピュータサイエンス専攻肥後研究室の皆様，ならびに事務職員 軽部 瑞穂 氏に心より深く感謝申し上げます．

参考文献

[1] Artificial Analysis. Independent analysis of ai models and api providers, 2024. Accessed on January 30, 2025. https://artificialanalysis.ai.

[2] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. PolicyLint: Investigating internal privacy policy contradictions on Google Play. In *Proc. USENIX Security 19*, pp. 585–602, August 2019.

[3] Yadagiri Annepaka and Dr. Partha Pakray. Large language models: a survey of their development, capabilities, and applications. *Knowledge and Information Systems*, pp. 1–56, December 2024.

[4] Anthropic. Ai research and products that put safety at the frontier. Accessed on January 30, 2025. https://www.anthropic.com.

[5] J Martin Bland and Douglas G Altman. Multiple significance tests: the Bonferroni method. *BMJ*, Vol. 310, No. 6973, p. 170, 1995.

[6] DeepSeek-AI. Deepseek llm: Scaling open-source language models with longtermism. *arXiv*, 2024. https://arxiv.org/abs/2401.02954.

[7] FOSSA. tl;drlegal. Accessed on January 30, 2025. https://www.tldrlegal.com.

[8] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[9] Open Source Initiative. The open source definition, 2007. Accessed on January 30, 2025. https://opensource.org/osd.

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

[11] Petros Karvelis, Dimitris Gavrilis, George Georgoulas, and Chrysostomos Stylios. Topic recommendation using Doc2Vec. In *Proc. IJCNN2018*, pp. 1–6, 2018.

[12] OpenAI. Gpt-4o system card. *arXiv*, 2024. https://arxiv.org/abs/2410.21276.

[13] OpenAI. Openai o1 system card. *arXiv*, 2024. https://arxiv.org/abs/2412.16720.

[14] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. Accessed on January 30, 2025. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[15] Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, Pranav Sandeep Dulepet, Saurav Vidyadhara, Dayeon Ki, Sweta Agrawal, Chau Minh Pham, Gerson C. Kroiz, Feileen Li, Hudson Tao, Ashay Srivastava, Hevander Da Costa, Saloni Gupta, Megan L. Rogers, Inna Goncearenco, Giuseppe Sarli, Igor Galynker, Denis Peskoff, Marine Carpuat, Jules White, Shyamal Anadkat, Alexander Miserlis Hoyle, and Philip Resnik. The prompt report: A systematic survey of prompting techniques. *arXiv*, 2024. https://arxiv.org/abs/2406.06608.

[16] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. EMNLP2013*, pp. 1631–1642, October 2013.

[17] Georgios S. Solakidis, Konstantinos N. Vavliakis, and Pericles A. Mitkas. Multilingual sentiment analysis using emoticons and keywords. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 2, pp. 102–109, 2014.

[18] Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv*, 2024. https://arxiv.org/abs/2312.11805.

[19] Llama Team. The llama 3 herd of models. *arXiv*, 2024. https://arxiv.org/abs/2407.21783.

[20] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS2017*, pp. 5998–6008, 2017.

[21] Sihan Xu, Ya Gao, Lingling Fan, Zheli Liu, Yang Liu, and Hua Ji. LiDetector: License incompatibility detection for open source software. *ACM Trans. Softw. Eng. Methodol.*, Vol. 32, No. 1, pp. 22:1–22:28, February 2023.

[22] 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.