Identifying Semantic Gaps between Legal Case Descriptions and Logical Fact Formulas Using LLMs: Definition and Preliminary Evaluation

Wachara Fungwacharakorn[1,*], May Myo Zin[1] and Ken Satoh[1]

[1]Center for Juris-Informatics, ROIS-DS, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 100-0003, Japan

Abstract

In computational law, formalization of legal rule-based reasoning relies heavily on the subsumption phase, where natural language case descriptions are aligned with formal logical fact formulas. However, several challenges arise due to semantic gaps. This paper investigates the capability of large language models (LLMs) to identify these semantic gaps during legal subsumption. We define a structured taxonomy categorizing semantic gaps into four distinct types (contradicted, unmentioned, open-textured, and specified) and preliminarily evaluate off-the-shelf LLMs (GPT-4.1, GPT-4.1 mini, Gemini 3.0 Pro, and Gemini 3.0 Flash-Lite). The results show that all models achieve moderate-to-high recall scores. However, more diverse qualitative and quantitative datasets are required to confirm the LLMs' capability in semantic gap identification.

Keywords

knowledge representation, large language models, legal subsumption, computational law, semantic gaps, automated legal reasoning, open-textured legal concepts

1. Introduction

Researchers in computational law have long been interested in formalizing legal rule-based reasoning. Foundational approaches have employed first-order logic programs such as Prolog [1] and Proleg [2], where each atomic formula represents either a legal condition or a legal conclusion. Legal reasoning with logic programs is typically divided into three phases. First, the *fact-finding* phase involves identifying and describing the facts of a case. Second, the *subsumption* phase aligns the case description with logical fact formulas that represent terminal legal conditions. Third, the *judgment* phase derives legal conclusions from the formalized rules and aligned facts.

This paper focuses on the subsumption phase. Recent studies [3] have shown that large language models (LLMs) are promising for automating the alignment between natural language case descriptions and formal representations. However, several challenges remain. First, case descriptions often omit facts that are presumed to be understood, resulting in missing conditions during legal reasoning. Second, many legal conditions are open-textured, such as the notion of a "safe distance," and therefore require interpretation by judges. Third, automated alignment methods may overlook legally relevant circumstances that should influence the application of a rule. Such circumstances are often identified by judges, particularly in higher courts, as exceptions or refinements intended to avoid counterintuitive consequences in future cases.

2. Defining Semantic Gaps in Legal Subsumption

To systematically analyze these challenges, this paper investigates them as instances of *semantic gaps*, referring to discrepancies between meanings expressed in natural language and their corresponding formal representations.

Joint Workshop on Statistics and Knowledge Integration for Logic, Learning, Ethical Decisions, and LLMs (SKILLED-LLMs 2026), co-located with FLoC 2026, Lisbon, Portugal

*Corresponding author.

wacharaf@nii.ac.jp (W. Fungwacharakorn); maymyozin@nii.ac.jp (M. M. Zin); ksatoh@nii.ac.jp (K. Satoh)

0000-0001-9294-3118 (W. Fungwacharakorn); 0000-0003-1315-7704 (M. M. Zin); 0000-0002-9309-4602 (K. Satoh)

We focus on first-order logic for representing legal rules. As in standard first-order logic, a formula is *atomic* if it can be expressed as $P(t_1, \ldots, t_n)$ where P is an n-ary predicate symbol and $t_1, \ldots, t_n$ are arguments. Predicate symbols are denoted by a camel-case phrase starting with uppercase letters, while variables appearing in arguments are denoted by lowercase letters. For any conjunction F of atomic formulas $\varphi_1, \ldots, \varphi_m$ (we use commas instead of logical conjunctions $\land$ to shorten the expression and we assume that all variables in F are existentially quantified), we denote the set $\{\varphi_1, \ldots, \varphi_m\}$ as S_F, the set of all variables in F as V_F, and the conjunction of all formulas in F with a variable v as F^v.

Since we deal with informal natural language, we define two semi-formal notions. For any case description C, we denote by $\mathbb{I}_C$ the set of all *reasonable* interpretations in law and we denote by $\mathbb{F}_C$ the conjunction of atomic formulas that *reasonably* represents C.

Given a conjunction formula F and a case description C, the task is to identify formulas in S_F and variables in V_F satisfying the following properties.

1. $\varphi \in S_F$ is said to be *contradicted* if $\forall i \in \mathbb{I}_C : i \vDash \neg\varphi$.
2. $\varphi \in S_F$ is said to be *unmentioned* if ($\forall i \in \mathbb{I}_C : i \nvDash \neg\varphi$) and ($\forall i \in \mathbb{I}_C : i \nvDash \varphi$).
3. $\varphi \in S_F$ is said to be *open-textured* if ($\exists i \in \mathbb{I}_C : i \vDash \neg\varphi$) and ($\exists i \in \mathbb{I}_C : i \vDash \varphi$).
4. $v \in V_F$ is said to be *specified* if
   $\exists S \subseteq S_{F^v} : [S \neq \emptyset$; ($\forall i \in \mathbb{I}_C : i \vDash S$); $\mathbb{F}_C^v$ implies S; and S does not imply $\mathbb{F}_C^v]$.

While semantic gaps and open-textured legal concepts have been discussed in prior computational law studies (e.g., [4, 5]), this paper proposes a structured taxonomy of semantic gaps tailored to common challenges arising in legal subsumption tasks. The taxonomy was developed through analysis of recurring mismatches observed during the automatic translation of case descriptions into logical fact formulas.

To demonstrate the defined semantic gaps, we consider a traffic rule stating that a person driving a vehicle on a main road who wants to turn left into a side road must activate a turn signal at the junction. We represent the rule using the following conjunction formula:

$$Vehicle(v), MainRoad(r), SideRoad(s), Junction(j, r, s),$$
$$Drives(x, v, r), WantsToTurnLeft(x, r, s), ActivatesTurnSignal(x, j)$$

Next, we consider a case description: a driver of an agricultural tractor wanted to turn right from a 7.6 m-wide road and activated a turn signal at the junction outside a built-up area. The semantic gaps aimed to be identified are as follows:

1. $WantsToTurnLeft(x, r, s)$ is contradicted since the case description mentions that the driver wants to turn right, which contradicts the formula stating that the driver wants to turn left.
2. $SideRoad(s)$ is unmentioned since the case description does not mention any facts regarding a side road.
3. $MainRoad(r)$ is open-textured since the case description mentions that the first road is 7.6 m wide. It can be considered a *main* road under some reasonable interpretations, but not under others.
4. 'v' and 'j' are specified because the case description provides additional information about the entities represented by these variables. While the formula requires a vehicle 'v' and a junction 'j', the case description specifies that 'v' is an agricultural tractor and that 'j' is located outside a built-up area.

3. Preliminary Evaluation

We conducted a preliminary evaluation of four LLMs (GPT-4.1, GPT-4.1 mini, Gemini 3.0 Pro, and Gemini 3.0 Flash-Lite). All models achieved recall scores at least 0.50 in all semantic gap categories, suggesting that LLMs can identify a substantial portion of semantic gaps annotated in the ground truth. However, the reliability of this evaluation is still limited, as the ground truth was preliminarily constructed based on only six traffic-law case descriptions and has not yet been verified by legal experts. Future evaluation will involve more diverse qualitative and quantitative datasets to confirm the LLMs' capability in semantic gap identification.

Acknowledgments

This work was supported by the "Strategic Research Projects" grant from ROIS (Research Organization of Information and Systems), the "R&D Hub Aimed at Ensuring Transparency and Reliability of Generative AI Models" project of the MEXT, by JSPS KAKENHI Grant Numbers, 25H00522 and 25H01112, and JST as part of Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE), Grant Number JPMJAP25B2. We would like to thank all reviewers for their helpful comments.

Declaration on Generative AI

During the preparation of this work, the first author used Google Gemini for the literature review and ChatGPT for the grammar and spelling check. After using these tools and services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, H. T. Cory, The British Nationality Act as a logic program, Communications of the ACM 29 (1986) 370–386.

[2] K. Satoh, K. Asai, T. Kogawa, M. Kubota, M. Nakamura, Y. Nishigai, K. Shirakawa, C. Takano, PROLEG: An Implementation of the Presupposed Ultimate Fact Theory of Japanese Civil Code by PROLOG Technology, in: T. Onada, D. Bekki, E. McCready (Eds.), New Frontiers in Artificial Intelligence, Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 153–164.

[3] M. M. Zin, H. T. Nguyen, K. Satoh, S. Sugawara, F. Nishino, Improving translation of case descriptions into logical fact formulas using LegalCaseNER, in: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, 2023, pp. 462–466.

[4] D. Song, A. Vold, K. Madan, F. Schilder, Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training, Information Systems 106 (2022) 101718.

[5] J. Savelka, K. D. Ashley, M. A. Gray, H. Westermann, H. Xu, Explaining legal concepts with augmented large language models (GPT-4), arXiv preprint arXiv:2306.09525 (2023).

A. Preliminary Evaluation Report

We evaluated four LLMs (GPT-4.1, GPT-4.1 mini, Gemini 3.0 Pro, and Gemini 3.0 Flash-Lite) on their ability to identify the semantic gaps. All models were instructed using the following prompt template. Default model settings were used and the temperature was set to zero.

```
You are an expert in computational law and legal subsumption. Your task is to identify
    semantic gaps between a natural language case description and a conjunction of first
    -order atomic formulas.

There are four types of semantic gaps:
1. "Contradicted":
- Structure Type: An atomic formula
```

```
- Informal Definition: The case description directly contradicts the atomic formula.
- Formal Definition: The negation of the atomic formula is true under all reasonable
    interpretations of the case description.
2. "Unmentioned":
- Structure Type: An atomic formula
- Informal Definition: The case description does not mention the atomic formula at all.
- Formal Definition: Neither the atomic formula nor its negation is true under any
    reasonable interpretations of the case description.
3. "Open-textured":
- Structure Type: An atomic formula
- Informal Definition: The legal status of the atomic formula depends on interpretation.
- Formal Definition: The atomic formula is true under some reasonable interpretations of
     the case description, while its negation is true under others.
4. "Specified":
- Structure Type: A variable (within a formula)
- Informal Definition: The case description provides a specific predicate to the
    variable and the predicate is useful for later cases.
- Formal Definition: When translating the case description into a conjunction of first-
    order atomic formulas, the sub-conjunction of the translated formulas containing
    that variable specifically implies a non-empty sub-conjunction (which is  true under
     all reasonable interpretations) of the input formula containing that variable, but
    not vice versa.

### Predicate Definitions:
{definitions}

### Case Description:
{case_description}

### Logical Formulas:
{formulas}

Identify the semantic gaps for the logical formulas based on the case description.
Output your answer STRICTLY as a JSON array of objects. Each object must have the
    following keys:
- "type": The type of gap (must be one of: "Contradicted", "Unmentioned", "Open-textured
    ", "Specified").
- "structure": A variable (if the type is "Specified") or an atomic formula (otherwise).
- "description": A brief explanation of why this gap exists based on the text.

Example Case Description:
The driver of an agricultural tractor wanted to turn right from a 7.6 m wide road and
    activated a turn signal at the junction outside a built-up area.

Example Logical Formulas:
Vehicle(v),MainRoad(r), SideRoad(s), Junction(j,r,s), Drives(x,v,r), WantsToTurnLeft(x,r
    ,s), ActivatesTurnSignal(x,j)

Example Output:
[
  {{
    "type": "Contradicted",
    "structure": "WantsToTurnLeft(x,r,s)",
    "description": "The case description states that the driver wants to turn right,
        which directly contradicts the formula stating that the driver wants to turn
        left."
  }},
  {{
```

```
    "type": "Unmentioned",
    "structure": "SideRoad(s)",
    "description": "The case description does not mention whether the driver wants to
        turn into a side road"
  }},
  {{
    "type": "Open-textured",
    "structure": "MainRoad(r)",
    "description": "The case description only mentions that the first road is 7.6 m wide
        ; it can be considered as a main road under some reasonable interpretations but
        cannot under others"
  }},
  {{
    "type": "Specified",
    "structure": "v",
    "description": "The case description should be translated into the formula with
        AgriculturalTractor(v), which specifically implies Vehicle(v) and
        AgriculturalTractor is a useful predicate for later cases"
  }},
  {{
    "type": "Specified",
    "structure": "j",
    "description": "The case description should be translated into the formula with
        Junction(j,r,s),OutsideBuiltUpArea(j), which specifically implies Junction(j,r,s
        ) and OutsideBuiltUpArea is a useful predicate for later cases"
  }}
]
```

The case descriptions were adapted from six German traffic-law cases. The logical formulas were constructed manually based on the target traffic rules. The results are then evaluated by comparing with the semantic gap *type* and *structure* in the manually annotated ground truth, neglecting the semantic gap *description*. Each semantic gap structure identified by an LLM was counted as *true positive* if it was identified as the same type annotated in the ground truth, and *false positive* if it was not. Meanwhile, each semantic gap structure in the ground truth was counted as *false negative* if it was not correctly identified by the LLM. The performance was then measured using the precision, recall, and F1-score, following standard classification performance metrics, as reported in the following table.

Table 1
Consolidated Model Performance across Semantic Gap Types

Type	GPT-4.1			GPT-4.1 mini			Gemini 3.0 Pro			Gemini 3.0 Flash-Lite			Support
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	
Contradicted	0.50	1.00	0.67	0.50	1.00	0.67	0.50	1.00	0.67	0.50	1.00	0.67	1
Unmentioned	0.38	1.00	0.55	0.29	0.67	0.40	0.60	1.00	0.75	1.00	1.00	1.00	3
Open-textured	0.50	0.75	0.60	0.67	0.50	0.57	1.00	0.50	0.67	1.00	0.50	0.67	4
Specified	0.82	0.90	0.86	0.40	0.73	0.52	0.69	0.90	0.78	0.83	1.00	0.91	10