Local Proceedings of the 20th International Workshop on Juris-Informatics (JURISIN 2026)

in association with
the 18th JSAI International Symposia on AI (JSAI-isAI 2026)

JURISIN 2026 Co-chairs

Ken Satoh, Center for Juris-Informatics, Japan
Yoshinobu Kano, Shizuoka University, Japan

June 7-8, 2026

Preface

The International Workshop on Juris-Informatics, JURISIN, is held annually as an conjunction workshop to the annual national conference of the Japanese Society for Artificial Intelligence, JSAI, as a forum for exchange between researchers in law and informatics. The first JURISIN was held in 2007; initially, most of the papers were on legal reasoning legal argumentation theory and legal ontology, and only two presentations were made from foreign researchers. Since then, JURISN has been held once a year in conjunction with the national conference of JSAI and has grown to become the leading international workshop on legal informatics in the Asian region. In this year's 20th JURISN, 21 papers were submitted, and reflecting the recent development of AI, there has been an increase in the number of submissions in new fields that did not exist in the past, such as research applying large-scale language models and intellectual property rights. 21 papers were reviewed by 3 program committee members each, and 17 of them were accepted. Only two of the accepted papers were written by Japanese first authors, indicating the international recognition of the workshop. In addition, 8 of the 17 accepted papers will be published as LNAI Proceedings and 9 of the accepted papers is published in this volume. We hope that JURISIN 2026 will further broaden the international exchange between researchers in law, informatics, and AI. We would like to thank the Steering Committee, the Advisory Committee, the Program Committee, and the members of the Organizing Committee of JSAI-isAI.

June 7, 2026 Yoshinobu Kano
Takasaki Ken Satoh

Table of Contents

Program Committee

Michał Araszkiewicz	Jagiellonian University
Ryuta Arisaka	Kyoto University
Agata Ciabattoni	TU Wien
Giuseppe Contissa	University of Bologna
Marina De Vos	University of Bath
Huimin Dong	TU Wien
Wachara Fungwacharakorn	National Institute of Informatics, Sokendai University
Randy Goebel	University of Alberta
Guido Governatori	Central Queensland University
Tokuyasu Kakuta	Chuo University
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Yuntao Kong	NII
Anelia Kurteva	University of Birmingham
Davide Liga	University of Luxembourg
Makoto Nakamura	Niigata Institute of Technology
María Navas-Loro	UPM
Ha-Thanh Nguyen	National Institute of Informatics
Le-Minh Nguyen	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
Yoshiaki Nishigai	Chiba University
Katsumi Nitta	Institute of Science Tokyo
Yasuhiro Ogawa	Nagoya City University
Shozo Ota	The University of Tokyo
Monica Palmirani	CIRSFID, ALMA-AI
Livio Robaldo	Legal Innovation Lab Wales, University of Swansea
Víctor Rodríguez Doncel	Universidad Politécnica de Madrid
Seiichiro Sakurai	Meiji Gakuin University
Diogo Sasdelli	University for Continuing Education Krems
Ken Satoh	Center for Juris-Informatics, ROIS, Japan
Akira Shimazu	JAIST
Cor Steging	Rijksuniversiteit Groningen
Satoshi Tojo	Asia University
Katsuhiko Toyama	Nagoya University
Vu Tran	The Institute of Statistical Mathematics, Japan
Bart Verheij	University of Groningen
Sabine Wehnert	Ruhr University Bochum
Adam Wyner	Swansea University
Hiroaki Yamada	Institute of Science Tokyo
Masaharu Yoshioka	Hokkaido University
May Myo Zin	Center of Juris-Informatics, ROIS-DS

Thomas Ågotnes University of Bergen

Additional Reviewers

V
Vo, Trung

How the ECtHR Frames Artificial Intelligence: A Distant Reading Analysis

Michael Sierra[1][0009-0009-7343-7493]
[1] Hebrew University of Jerusalem, Jerusalem, Israel
michael.sierra@mail.huji.ac.il

Abstract. This study examines how the European Court of Human Rights (ECtHR) conceptualizes artificial intelligence (AI) in its case law through distant reading and computational text analysis. Drawing on a corpus of ten ECtHR judgments from 2018 to 2024 that explicitly mention "artificial intelligence," collected from HUDOC and analyzed with Voyant Tools, it explores how judges frame AI in written decisions.

The analysis suggests that AI is most often discussed in connection with risk, security, surveillance, state power, data processing, and privacy. References to AI frequently appear alongside terms such as "threat," "weapons," "surveillance," "state," and "security," indicating that the Court tends to address AI through concerns about national security and potential human-rights infringements, especially under Articles 8 and 6.

The findings suggest that the ECtHR is still at an early stage of doctrinal engagement with AI. More cautiously, they show that when AI appears in the Court's judgments, it is framed more often through surveillance, public authority, and rights protection than through innovation or efficiency. Given the small and exploratory corpus, the study should be understood as an initial mapping of judicial discourse rather than a basis for broad generalization.

Keywords: Artificial Intelligence in Judicial Discourse, Distant Reading and Text Analysis, European Court of Human Rights (ECtHR).

1 Introduction

How do judges of the ECtHR describe and frame AI in their rulings, and what are the main contexts in which AI is addressed? This research question seeks to explore how a prominent international judicial body understands and conceptualizes the notion of AI, a rapidly evolving technological and ethical domain. The ECtHR, responsible for upholding the European Convention on Human Rights, plays a critical role in interpreting how new technological developments affect fundamental rights. Although legal scholarship has increasingly examined how AI impacts privacy [1], due process [2], and discrimination [3], there is a gap in empirical research focusing on how judges themselves speak about AI in their rulings. By employing distant reading methods [4], particularly through tools such as Voyant, this study aims to identify recurring

patterns in the linguistic portrayal of AI. This approach complements close reading by identifying recurring lexical patterns, co-occurrences, and thematic clusters across the corpus. It does not, by itself, establish judicial meaning; rather, it provides an empirical map of recurring textual patterns that must be interpreted in light of the surrounding legal and factual context. Unlike traditional close reading, this method highlights thematic patterns and latent meanings across a corpus. Previous research has addressed the normative challenges AI poses to human rights, often using legal and philosophical frameworks [5]. However, these works have not utilized digital tools to explore actual case law quantitatively. This article bridges this gap by combining qualitative legal theory with computational text analysis. The article adopts an exploratory design and focuses only on ECtHR judgments that explicitly refer to "artificial intelligence" in the text of the decision. This design allows for a transparent and verifiable corpus, but it also introduces an important limitation: the study does not capture the broader universe of cases involving algorithmic governance, biometric systems, automated decision-making, or digital surveillance where AI-related issues may be present without being expressly labelled as such. The findings should therefore be read as a mapping of explicit judicial references to AI, rather than as a comprehensive account of the Court's technology jurisprudence. The article examines the full set of ECtHR judgments located through a HUDOC search for explicit references to "artificial intelligence" during the period 2018-2024. The resulting corpus is small, but that smallness is itself analytically significant: it indicates that explicit judicial engagement with AI in the Court's published case law remains limited at this stage.

Distant reading and text analysis are particularly relevant to this study because they enable the systematic exploration of large corpora of legal texts, such as ECtHR judgments, in ways that traditional close reading cannot. By applying tools like Voyant, researchers can detect patterns in language use, frequency, co-occurrence, and semantic framing that may remain unnoticed in individual case analysis. This computational approach facilitates the identification of recurring rhetorical strategies, dominant legal themes, and conceptual associations tied to **AI**. Moreover, such methods are inherently more objective, as they rely on quantitative data and reduce the extent to which the researcher's personal interpretations or theoretical commitments are embedded in the analysis. This contributes to greater methodological transparency and replicability. In this way, distant reading can shed new light on the research question by revealing how judges implicitly or explicitly construct narratives around AI, what terminology they prefer, and in which legal contexts, such as privacy, surveillance, or discrimination, AI most often appears. Ultimately, this method complements doctrinal and theoretical legal scholarship with empirical insights, offering a broader and more nuanced understanding of how the ECtHR judiciary conceptualizes emerging technologies.

Moreover, studying the jurisprudence of the ECtHR on AI is compelling for several interrelated doctrinal and systemic reasons. Firstly, international law scholars regard the ECtHR as the world's most effective international human rights tribunal [6]. Although the European Convention on Human Rights (ECHR), drafted in 1950, makes no express reference to emerging technologies, the Court is increasingly called upon to apply foundational rights, as privacy (Article 8), freedom of expression (Article 10), fair trial (Article 6), and non-discrimination (Article 14), to novel AI contexts, including algorithmic decision-making, facial recognition, surveillance systems, and predictive policing. This interpretive endeavour showcases how a time-bound human rights framework can adapt to unprecedented technological challenges. The ECtHR's application of the "living instrument" doctrine, the principle that the Convention evolves in light of contemporary realities, makes it a critical venue for examining how fundamental rights are recalibrated in response to AI-driven societal shifts. For example, in *Gaughran v. the United Kingdom*, *2020*, the Court weighed state interests against digital privacy concerns in biometric surveillance cases [7]. Secondly, AI systems, especially opaque or automated ones, exacerbate normative tensions with core ECtHR values like accountability, transparency, and dignity. The Court's rulings constitute a vital legal forum where such conflicts are litigated, frequently exposing jurisprudential lacunae and prompting calls for new standards, particularly concerning explainability and human oversight [8].

Additionally, ECtHR jurisprudence exerts significant transnational influence. Although its decisions are binding on its 46 Member States, they also carry persuasive authority globally, shaping international human rights discourse and informing soft-law initiatives, including the Council of Europe's Framework Convention on Artificial Intelligence, Human Rights, Democracy, and the Rule of Law [9]. Furthermore, examining ECtHR case law offers a form of anticipatory governance. Even in the relative absence of explicit AI rulings, the Court's established reasoning in related technology cases (such as mass surveillance and automated profiling) enables scholars, policymakers, and technologists to anticipate how future AI-related claims might be resolved.

From a comparative legal and philosophical perspective, the ECtHR's evolving doctrine illuminates how AI disrupts traditional legal constructs such as intent, causality, and subjectivity and how responsibility is attributed within hybrid human-machine environments. The Court's case law thus provides fertile ground for exploring how these enduring legal concepts are reinterpreted through a human rights lens [10]. Moreover, unlike courts primarily focused on economic or administrative regulation (e.g., the CJEU or national courts), the ECtHR is dedicated to safeguarding a regional human rights treaty. Its mandate naturally centers on rights at risk from AI deployment, such as privacy, dignity, and due process [11]. The ECtHR has long developed doctrinal tools, like the margin of appreciation, necessity in a democratic society, and positive obligations, in cases involving surveillance (*Klass and others v. Germany, 1978*, [11] *S.* and *Marper v. UK*, 2008 [12] *Roman Zakharov v. Russia,2015* [13]). These doctrines are being extended to algorithmic governance contexts [14].

Finally, the ECtHR operates within a broader ecosystem, including the Council of Europe's AI policy instruments (e.g., CAHAI). This synergy between judicial decisions and policy frameworks permits a more holistic and forward-looking approach than is typical in other jurisdictions [15]. In conclusion, analysing ECtHR jurisprudence provides a unique perspective on the domestication of AI within the human rights legal order, complementing technical, regulatory, or commercial databases. It offers a rich doctrinal, normative, comparative, and anticipatory vantage point on AI's intersection with fundamental rights in Europe and beyond.

2 Building the Corpus

Constructing the corpus required a targeted and transparent selection strategy because the ECtHR does not maintain a separate thematic database for AI-related judgments. An initial HUDOC search returned 44 documents. These documents were manually reviewed, and only judgments containing an explicit and substantively relevant reference to "artificial intelligence" were retained for analysis, resulting in a final corpus of ten judgments.

The corpus consists of all HUDOC judgments located through a targeted search for explicit references to "artificial intelligence" during the period 2018–2024 and

subsequently screened for substantive relevance. The resulting dataset is limited in size, but it is analytically useful because it captures the Court's earliest explicit encounters with AI vocabulary in adjudication. The HUDOC is maintained by the Council of Europe. It provides free public access to the Court's case law, including judgments**,** decisions**,** advisory opinions, and legal summaries**.** The database covers material in English and French**,** the Court's two official languages, and is an essential resource for legal practitioners, scholars, and policymakers seeking to analyze the jurisprudence of the ECtHR. HUDOC includes advanced search functionalities, enabling users to filter cases by article of the European Convention on Human Rights, keywords, date, respondent state, and other parameters, making it an indispensable tool for empirical legal research and doctrinal analysis. The analysis, therefore, does not claim to capture all technology-related ECtHR jurisprudence, but only the narrower subset of judgments in which AI is expressly mentioned in the text of the decision.

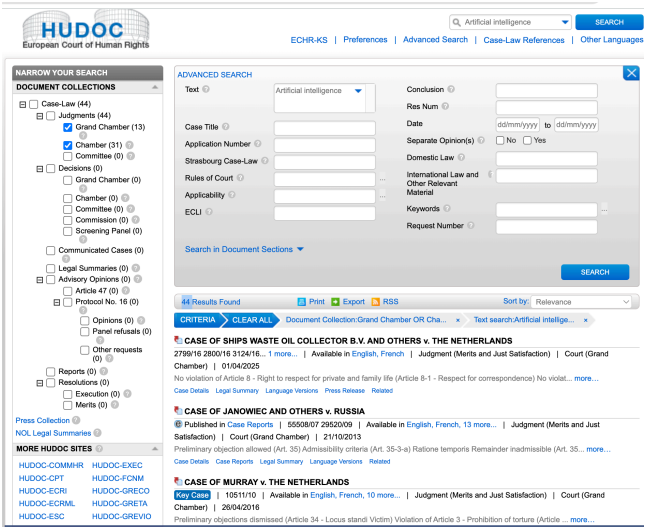


Fig.1. A figure of the results of the search on the HUDOC database (44 results).

Since the search on the HUNDOC was a result of the keywords 'artificial' + 'intelligence', there were files that included these two words but were about 'artificial' or 'intelligence' and not about AI, so the data was cleaned manually, and the non-relevant ones were deleted. After that, only 10 files of Judgments were, and from them the corpus was built. In other words, the corpus consists of 10 judgments delivered by the ECtHR between 2018 and 2024, in which the term "artificial intelligence" appears explicitly. Each Word file was then transformed into a text file. The corpus had, according to the summary function of Voyant, 10 documents (n=10) with 577,673 total words and 19,272 unique word forms (see Fig. 2). In addition to computational outputs, each case was qualitatively coded according to year, legal context, and the degree to which AI

appeared as a central issue, a supporting technological element, or an incidental reference. This coding did not serve as a formal statistical variable, but as an interpretive aid for evaluating the significance of recurrent lexical patterns.

No formal sensitivity analysis was conducted with alternative search terms, stop-word settings, or corpus boundaries. The findings should therefore be treated as provisional and dependent on the present corpus-construction choices. Future research should test whether similar patterns emerge when the dataset is expanded to adjacent terms such as algorithmic decision-making, profiling, biometric surveillance, or automated systems.

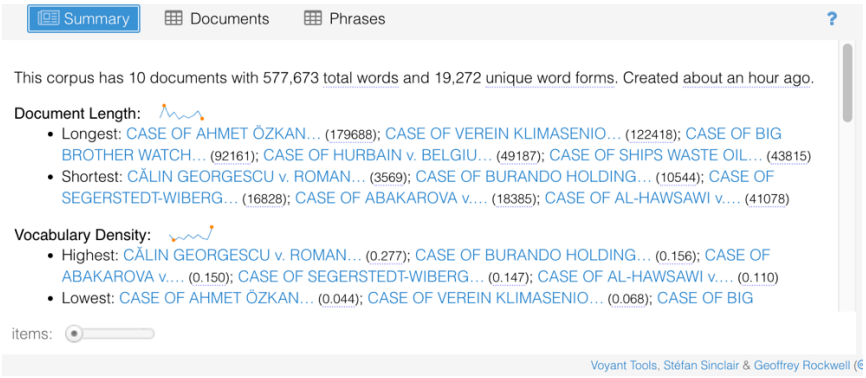


Fig. 2: The information regarding the corpus according to the 'summary' function of Voyant.

The rationale for the corpus selection was that although the number of judgments mentioning AI is still small, the topic is emerging and rapidly gaining relevance in the legal field. Therefore, even a relatively small but focused corpus can reveal how courts are beginning to grapple with the complexities of AI. Importantly, the ECtHR provides a valuable window into transnational legal discourses that influence European member states.

The judgments vary in subject matter and jurisdictional context, including issues such as surveillance, data retention, predictive policing, automated decision-making, and state security. By analyzing judgments across multiple thematic categories, this corpus allows a multi-dimensional examination of how AI is framed legally and ethically. The small size of the corpus imposes clear limits on generalization. For that reason, the findings should be read as an exploratory mapping of emerging judicial discourse rather than as evidence of a settled or comprehensive ECtHR doctrine on AI.

Moreover, the cases in the corpus do not all engage AI to the same degree. In some judgments, AI forms part of the core factual or normative dispute, particularly where surveillance, biometric processing, or automated technologies are implicated. In others, AI is mentioned only briefly or incidentally, for example, as part of a broader discussion of digital technologies or evidentiary context. This distinction is important for interpretation, because the corpus reflects different intensities of judicial engagement rather than a uniform body of AI case law.

3 Knowledge Modeling and Tabular Data Construction

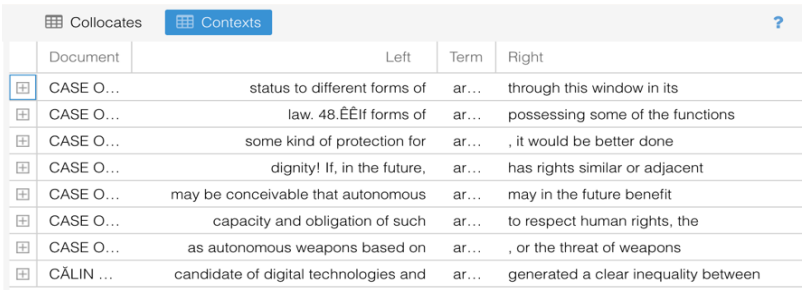
To support the research, I created a structured spreadsheet on Microsoft Excel that includes the following data fields for each judgment: (1) Case title and date of ruling; (2) A summary of the factual context; (3) Judgment outcome (violation/no violation).

Furthermore, I linked all the data files in the last column so that I can easily find and refer to the data in accessibility. This table serves as a knowledge model that enables the mapping of how AI appears in different legal contexts and how it correlates with specific human rights concerns. For example, linking AI with Article 8 (right to privacy) or Article 6 (fair trial) may indicate judicial concerns about surveillance or automated adjudication. This relational data structure also supports cross-case comparisons and pattern recognition.

4 Distant Reading using Voyant

The .txt files of the judgment were uploaded to the Voyant Tool. Before conducting the distant reading, the words 'court' and 'convention' were added as stop words since they appear in each document in the corpus and are not particular or relevant. To conduct the distant reading, distinct tools in Voyant were employed, including four beyond the default skin. These tools reveal not only word frequencies but also semantic relations, distribution trends, and thematic associations as explained in the following lines. In methodological terms, the Voyant outputs are used here as interpretive prompts rather than as autonomous evidence of doctrinal position.

4.1 Contexts: The first tool displays the term 'artificial intelligence' as it appears in various cases within the corpus, along with the words that immediately precede and follow it. We can observe that it frequently co-occurs with terms related to legal responsibilities, ethical challenges, and technological risks. The discourse often frames AI within the context of dignity, protection, and legal status, revealing recurring legal concerns connected to human rights, autonomy, and regulatory obligations. These linguistic patterns suggest that both judicial and scholarly texts are actively negotiating the normative implications of AI and its integration into existing legal frameworks.


Collocates Contexts ?

Document	Left	Term	Right
CASE O…	status to different forms of	ar…	through this window in its
CASE O…	law. 48.ÊÊIf forms of	ar…	possessing some of the functions
CASE O…	some kind of protection for	ar…	, it would be better done
CASE O…	dignity! If, in the future,	ar…	has rights similar or adjacent
CASE O…	may be conceivable that autonomous	ar…	may in the future benefit
CASE O…	capacity and obligation of such	ar…	to respect human rights, the
CASE O…	as autonomous weapons based on	ar…	, or the threat of weapons
CĂLIN …	candidate of digital technologies and	ar…	generated a clear inequality between

Fig. 3: The term 'artificial intelligence' analyzed by the context function in Voyant.

4.2 The Documents Terms function: the second tool that was used is the Documents Terms. Corpus Terms is a table view of terms that appear in the entire corpus. Features include: reordering by term and count; a sparkline graph of the term frequency trends across the corpus. The query 'artificial intelligence' was used to see its frequency in the corpus. We can see that the term 'artificial intelligence' appears explicitly in multiple ECHR documents, with varying density. In some cases, it is mentioned only once but occupies a significant proportion of the document, indicating focused discussions on AI in shorter opinions or concurrences. In others, it is repeated several times, suggesting sustained engagement. The trend lines suggest that AI is often discussed throughout the documents, not only in introductions or conclusions, implying substantive treatment.

4.3 Cirrus (Word Cloud): The third Voyant tool that was used is Cirrus (Word Cloud), which displays the most frequent terms in the corpus in a visually engaging format. This tool is useful for quickly identifying prominent themes, recurring vocabulary, and potential keywords that warrant deeper investigation in the text analysis process. The visualization reveals that judges frequently discuss AI in connection with "data," "article," and "processing," indicating a focus on AI within data governance and legal textual analysis (e.g., rights codified in articles). We can see that although the corpus is built from judgments that are about AI, the term "AI" itself is not a dominant word in the corpus and does not even appear in the word cloud. This suggests that the discussion around AI in these judgments is often implicit, embedded in broader legal or technical terminology, or mediated through related concepts such as "data," "surveillance," or "algorithmic decision-making," rather than being explicitly framed in terms of "artificial intelligence."

Fig. 5: Word cloud.

4.4. Trends Tool: the fourth Voyant tool that was used is the 'Trends', which, according to Voyant, shows a line graph of the relative frequencies across the corpus (for multiple documents) or within a document. In our case, Voyant was asked to show 3 terms: "risk*" (light pink); "threat" (red); "artificial intelligence*" (yellow). This updated graph from Voyant Tools compares the relative frequency of the three terms across the corpus. We can see that peaks of the term "artificial intelligence*" (yellow) are visible in specific cases, such as the 3rd, 7th, and 10th, where AI played a more

central role. This temporal visualization indicates that discourse on AI is growing but remains sporadic.

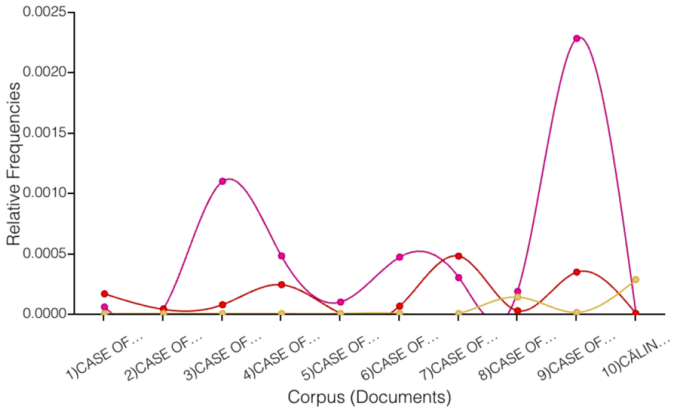


Fig. 6: Trends in the corpus.

It also seems that "threat" (red) and "risk*" (light pink) appear more frequently and consistently than "artificial intelligence", indicating maybe that judges are more focused on general notions of danger or harm than on AI specifically. This reflects that AI is still not central in judicial risk assessments in this corpus. Moreover, the peaks of "threat" (red) and "risk*" (light pink) occur in different documents. For example, document 6 shows a peak for "threat" (red) and not "risk*" (light pink), and document 7 has a moderate rise in "risk*" (light pink) but not in "threat" (red). This may suggest that the two terms are being used in distinct contexts, perhaps referring to different types of legal or factual concerns. Judges may distinguish between legal threats (e.g., national security) and risks (e.g., procedural, reputational, or operational). Additionally, "artificial intelligence*" (yellow) appears at a low but gradually increasing frequency, particularly from document 7 onward. This might suggest a shift toward increased judicial awareness of AI-related issues or that AI appears in newer, perhaps more specialized or high-profile cases. Moreover, there is a weak correlation between AI and Risk/Threat terminology. Since peaks in "artificial intelligence" do not align with those in "risk" or "threat", this suggests that Judges are not yet framing AI as a major risk or threat in the analyzed cases, or the use of "risk" and "threat" refers to other legal domains (e.g., physical harm, terrorism, business harm).

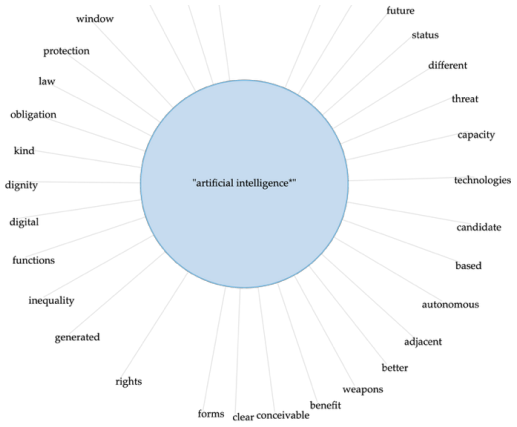
In conclusion, this graph supports the argument that AI is currently peripheral in judicial discourse compared to broader terms like "risk" and "threat", and that courts do not systematically associate AI with danger, at least in this corpus. This may indicate a gap between policy discourse on AI risks and judicial framing of such technologies. The frequency trend of the term "artificial intelligence" across judicial decisions in the corpus reveals a clear pattern: AI is not uniformly present in judicial reasoning but appears selectively and increasingly in later documents. Its near absence in the earlier cases suggests that AI was not yet a salient legal concern or was mentioned only tangentially. A noticeable uptick in documents 7 and 8 indicates emerging judicial engagement with AI-related issues, while the spike in document 10 suggests a case

where AI plays a central or heavily debated role. This trend may reflect the growing relevance of AI in legal contexts, either due to its increased deployment (e.g., in surveillance, automated decision-making, or algorithmic evidence) or due to judicial awareness of its normative implications. The graph supports the view that judicial discourse on AI is not yet stabilized, but is rapidly evolving, with isolated judgments serving as early interpretive anchors.

4.5. Links Tool: Highlights co-occurrence patterns, showing that "artificial intelligence" is closely associated with negative or cautionary terms such as "threat," "weapons," and "window." This suggests that AI is framed primarily in security or risk-based contexts rather than opportunity-oriented narratives. Notably, AI is also linked to “rights,” “protection,” “weapons,” “state,” “threat,” and “appeal.” This suggests that judicial discourse is deeply concerned with human rights and constitutional protections concerning AI; security and military implications (e.g., autonomous weapons or surveillance tools); state responsibility and sovereignty in AI deployment and judicial review mechanisms (“appeal”) that may be relevant when AI influences administrative or judicial decisions. The connection to words like “forms,” “window,” and “processing” may also suggest procedural dimensions, such as how AI is used in legal processes, or concerns about due process when AI is involved. Overall, the map supports the argument that judges tend to frame AI not only as a technological tool but as a phenomenon that intersects with legal rights, state authority, and institutional accountability.

A closer look at the relevant passages suggests that these associations are not uniform across the corpus. In some judgments, terms such as “risk” or “threat” arise from the broader factual matrix of the case, such as security, criminal investigation, or military technologies, rather than from a doctrinal characterization of AI itself. In other cases, however, references to AI appear more directly alongside concerns about surveillance, biometric processing, automated assessment, or state power. The value of the distant-reading output, therefore, lies in identifying recurring discursive environments, not in proving that each neighboring term is conceptually attached to AI in the same way.

Fig. 7: links.



4.6. Context Tool (Collocates): Demonstrates the proximity of emotionally and legally charged words to "AI," such as "threat," "surveillance," and "control." These collocates imply an atmosphere of suspicion and concern around AI deployment.

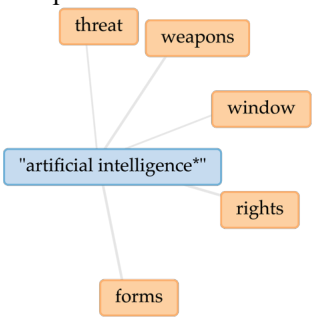


Fig. 8: Context tool.

4.7. Topic Modeling: Identifies recurring topics across the corpus. The topics linked to AI frequently include terms like "data," "law," "security," "state," and "rights," confirming the dominant themes of privacy, regulation, and risk.

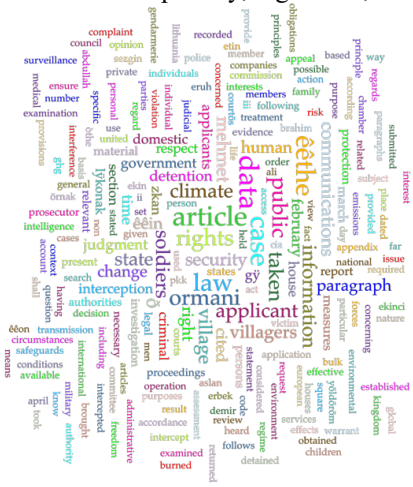


Fig. 9: Topic Modeling.

4.7. Terms Berry: Offers a visual network of word relationships. The term "intelligence" appears in nine documents, most often in conjunction with "law," "security," "data," and "public." This visual confirms the thematic centrality of regulation and state power in the AI discourse.

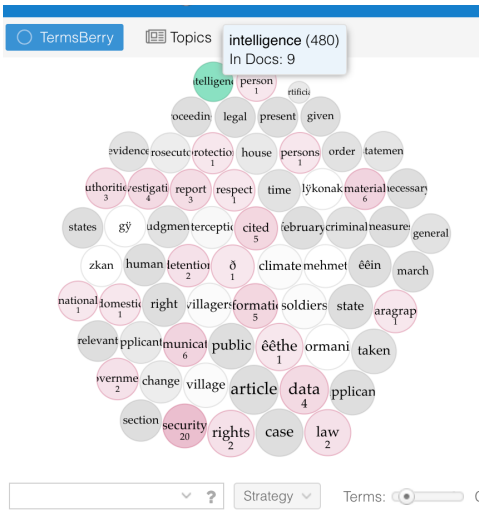


Fig. 10: Terms Berry.

As Cameron Blevins wrote in his article: “Counting word frequency is a somewhat blunt instrument that, if used carefully, can certainly yield meaningful results. Utilizing sparklines to visualize individual word frequencies offers two advantages for historical inquiry: (1) Coherently display general trends. (2) Reveal outliers and anomalies” [16].

Accordingly, the analysis should not be read as showing that the Court has developed a coherent doctrine of AI across all ten judgments; rather, it shows that AI appears with varying degrees of centrality across a small but growing set of cases. The distant-reading outputs are descriptive rather than self-interpreting. To assess whether the identified lexical patterns correspond to meaningful elements of judicial reasoning, the following section briefly revisits selected judgments in context.

5 Contextual Reading of Selected Cases

The distant-reading results should not be treated as self-interpreting. To assess whether the lexical patterns identified in the corpus correspond to meaningful elements of judicial reasoning, this section briefly revisits selected cases in context. The aim is not to replace the computational analysis with full doctrinal reconstruction, but to test whether the recurrent patterns identified through distant reading are borne out in the factual and legal structure of particular judgments. A first and especially important example is *Călin Georgescu v. Romania, 2025* [17]. While this case highlights how AI-related arguments are entering the Strasbourg system, the 'rights-sensitive analysis' mentioned actually originated at the domestic level. In the proceedings before the Romanian Constitutional Court, AI was directly linked to the factual basis of the electoral dispute, with references to 'massive exposure and preferential treatment' on TikTok and to the 'non-transparent and manipulatory use' of AI algorithms.

However, in doctrinal terms, the ECtHR did not endorse or expand upon these AI-specific concerns. In its March 2025 decision, the Court declared the application

inadmissible, ruling that Article 3 of Protocol No. 1 (right to free elections) does not apply to presidential contests. This case, therefore, supports a more nuanced claim for distant-reading: explicit AI terminology in the corpus often reflects the factual record of domestic disputes rather than a settled ECtHR doctrine. It illustrates that while AI-related manipulation is being litigated at the national level, the Strasbourg Court has remained procedurally cautious, avoiding a substantive ruling on AI's impact on democratic integrity in this specific instance.

By contrast, *Big Brother Watch and Others v. the United Kingdom,2021* [18] is structured around bulk interception, intelligence sharing, communications data, and safeguards under Articles 8 and 10. The judgment is doctrinally centred on surveillance powers, authorization, oversight, and the protection of privacy and journalistic confidentiality. This matters for the interpretation of the distant-reading output. Where words associated with security, secrecy, or intrusion cluster around technologically inflected cases, they do not necessarily show that the Court is conceptually framing AI itself as a threat. They may instead reflect the broader legal environment of surveillance litigation. Read doctrinally, *Big Brother Watch* thus serves as an important limiting example: it shows that technologically charged vocabulary often belongs to the surrounding architecture of privacy and surveillance adjudication, rather than to a distinct AI doctrine.

A similar point emerges from *Youth Initiative for Human Rights v. Serbia, 2013* [19]. That case concerns access to information about electronic surveillance and the role of civil society in enabling informed public debate about secret state practices. The Court held that the refusal to disclose the requested information impaired the applicant organisation's ability to contribute to public discussion on surveillance. Although the judgment is not framed through AI, it is doctrinally significant because it shows that the Court's technology-related jurisprudence often develops through questions of secrecy, accountability, and democratic oversight. In relation to the distant-reading analysis, such a case helps explain why references to technological governance may be embedded in a wider human-rights discourse about information asymmetry and institutional power, even where AI terminology is absent or marginal. The contrast becomes even clearer in *Ships Waste Oil Collector B.V. and Others v. the Netherlands, 2025* [20]. The judgment itself concerns the transmission and use of intercept data in competition-law proceedings and is doctrinally focused on Article 8 safeguards, arbitrariness, abuse, and the review of data transfers between authorities. The reference to AI does not arise in the operative reasoning of the Grand Chamber on those issues. Instead, it appears in a separate opinion, in a speculative reflection on whether future forms of AI might one day seek Convention protection and whether such protection would require a new Protocol. This passage is analytically interesting, but it should not be overstated. It does not show that the Court has already articulated a substantive position on AI rights or AI personhood. Rather, it illustrates that some references to AI in the corpus are incidental and future-oriented, not central to the resolution of the dispute before the Court.

Finally, *Hurbain v. Belgium, 2023 [21]* provides a useful neighboring example. The case concerns digital archives, indexing, searchability, anonymisation, and the "right to be forgotten" online. The doctrinal analysis focuses on the balance between privacy and freedom of expression in the context of online accessibility and the continuing availa-

bility of archived information. The judgment is therefore highly relevant to digital governance, but it is not framed around AI. Its importance for the present study lies in showing that the Court's broader technological jurisprudence often addresses search, retrieval, visibility, and long-term digital dissemination without conceptualizing these questions in terms of AI. This doctrinal context helps qualify the distant-reading patterns: not every technologically salient cluster of terms should be read as evidence of a stable judicial framing of AI itself.

Taken together, these cases support a narrower and more defensible conclusion than a broad claim that the Court frames AI in uniformly threat-based terms. Close reading suggests instead that explicit references to AI emerge unevenly across the corpus. In some cases, such as *Călin Georgescu*, AI-related tools are directly tied to concerns about manipulation, democratic fairness, and institutional neutrality. In others, the relevant doctrinal structure is surveillance, information access, digital memory, or data transmission, and AI appears only marginally or speculatively. Integrating the distant-reading results with close doctrinal analysis, therefore, refines the article's central claim: the corpus does not yet reveal a stable ECtHR doctrine of AI, but it does indicate that when AI is expressly mentioned, it tends to surface in rights-sensitive environments already shaped by concerns about surveillance, informational asymmetry, institutional power, and democratic legitimacy.

6 Conclusion and Reflection

As Richard Jean So said, "All models are wrong" [22]. The analysis reveals a consistent pattern in how AI is portrayed within ECtHR judgments: First, AI as threat: the co-occurrence of "artificial intelligence" with words like "weapons" and "threat" suggests that judges conceptualize AI primarily through a lens of danger and national security. This reflects broader public anxieties about autonomous weapons and uncontrolled surveillance.

Second, AI and rights violations: The presence of "rights," "data," and "privacy" in the proximity of AI points to a legal framing of AI as a potentially rights-infringing technology. In several cases, AI is connected to automated surveillance or profiling, raising alarms about violations of Articles 8 and 6. Third, it reveals a lack of nuanced perspectives: the findings show that positive or opportunity-oriented language (such as "innovation" or "efficiency") is largely absent. The judicial language remains defensive or cautionary, suggesting an early phase of doctrinal development where risk-avoidance dominates.

Finally, we notice ethical undercurrents: some documents implicitly address the ethics of the delegation of decision-making powers to machines, although this is not always articulated directly. This aligns with normative debates in legal theory about the legitimacy of algorithmic governance. The results partially confirmed my hypothesis that the ECtHR's judicial discourse on AI is dominated by themes of risk, control, and rights violations. This emphasis on threat-based language was stronger than expected, while more constructive framings of AI were minimal.

One of the article's central findings is therefore not only how AI is framed, but also how unevenly it appears: sometimes as a central rights issue, sometimes merely as part

of the technological background of the dispute. The digital tools proved highly effective in surfacing these rhetorical patterns. Especially useful were the Context and Links tools, which clarified how AI is embedded in broader legal and emotional frameworks. However, the tools also have limitations: without full close reading, the specific legal reasoning and doctrinal development remain underexplored. This study suggests that distant reading is a valuable approach for legal analysis, particularly in emerging fields where quantitative data is still scarce. It allows scholars to map conceptual terrains and track discursive evolution over time. In retrospect, expanding the corpus to include amicus briefs or academic commentaries cited in the rulings could enrich the analysis. Nonetheless, the current findings contribute meaningfully to our understanding of how legal institutions begin to grapple with AI as both a subject and object of human rights adjudication**.**

The limited number of judgments does not permit broad claims about the Court's jurisprudence as a whole. Instead, it allows a more modest contribution: identifying how AI is currently entering the Court's language, and under which legal and factual conditions it tends to appear.

Acknowledgments: The author thanks the Cheshin Center for Advanced Legal Studies at the Hebrew University of Jerusalem and the Shasha Center for Strategic Studies for their generous support. Many thanks also to Prof. Renana Keydar. There is no conflict of interest to declare.

References

1. Solove, Daniel J. Artificial intelligence and privacy, Fla. L. Rev. 77, 1 (2025). https://doi.org/10.2139/ssrn.4713111
2. Citron, Danielle Keats, Technological due process, Wash. UL Rev, 85, 1249 (2008). Available at: https://openscholarship.wustl.edu/law_lawreview/vol85/iss6/2
3. Heinrichs, Bert, Discrimination in the age of artificial intelligence, AI & society 37.1, 143-154 (2022). https://doi.org/10.1007/s00146-021-01192-2
4. Moretti, Franco. Distant reading. Verso Books, (2013). https://doi.org/10.1093/llc/fqu010
5. Taddeo, Mariarosaria, and Luciano Floridi, How AI can be a force for good, Science 361.6404, 751-752 (2018). https://doi.org/10.1126/science.aat5991
6. Szappányos, M. Artificial Intelligence: Is the European Court of Human Rights Prepared? Acta *Humana* 11(1):93–110 (2023). https://doi.org/10.32566/ah.2023.1.6
7. *Gaughran v. the United Kingdom*, no. 45245/15, European Court of Human Rights (2020, February 13). https://hudoc.echr.coe.int/eng?i=001-200817
8. Gabrielli, G., The Use of Facial Recognition Technologies in the Context of Peaceful Protest: The Risk of Mass Surveillance Practices and the Implications for the Protection of Human Rights. European Journal of Risk Regulation, 1–28. (2025). https://doi.org/10.1017/err.2025.26
9. Rodrigues, Rowena, Legal and Human Rights Issues of AI: Gaps, Challenges and Vulnerabilities, Journal of Responsible Technology 4 (2020): 100005. https://doi.org/10.1016/j.jrt.2020.100005.
10. CoE, Background paper for the Judicial Seminar 2025: Protecting human rights in a world of Artificial Intelligence, algorithms and big data, describing the Framework Convention on AI (May 2024). https://www.echr.coe.int/documents/d/echr/seminar-background-paper-2025-eng

11. *Klass and Others v. Germany*, App. No. 5029/71, 2 Eur. Ct. H.R. Ser. A (1978). https://hudoc.echr.coe.int/eng?i=001-57510
12. *S. and Marper v. the United Kingdom*, Apps. Nos. 30562/04 & 30566/04, Eur. Ct. H.R. (2008). https://hudoc.echr.coe.int/eng?i=001-90051
13. *Roman Zakharov v. Russia*, App. No. 47143/06, Eur. Ct. H.R. (2015). https://hudoc.echr.coe.int/eng?i=001-159324
14. Leslie et al., Artificial intelligence, human rights, democracy, and the rule of law: a primer (COE's CAHAI) (2021). https://doi.org/10.2139/ssrn.3817999
15. Rodrigues, Rowena, Legal and Human Rights Issues of AI: Gaps, Challenges and Vulnerabilities, Journal of Responsible Technology 4 (2020): 100005. https://doi.org/10.1016/j.jrt.2020.100005.
16. Cameron Blevins, Text Analysis of Martha Ballard's Diary (Part 3) (Oct. 19, 2009), https://cblevins.github.io/posts/text-analysis-of-martha-ballards-diary-part-3/ (last visited July 10, 2025).
17. *Călin Georgescu v. Romania, App. No. 37327/24, Eur. Ct. H.R. (Mar. 6, 2025*). https://hudoc.echr.coe.int/eng?i=001-242417
18. *Big Brother Watch and Others v. the United Kingdom*, Applications Nos. 58170/13, 62322/14, & 24960/15, Eur. Ct. H.R. (May 25, 2021). https://hudoc.echr.coe.int/eng?i=001-210077
19. *Youth Initiative for Human Rights v. Serbia*, App. No. 48135/06, Eur. Ct. H.R. (June 25, 2013). https://hudoc.echr.coe.int/eng?i=002-7585
20. *Ships Waste Oil Collector B.V. and Others v. the Netherlands*, Applications Nos. 2799/16, 2800/16, 3124/16, & 3205/16, Eur. Ct. H.R. (Apr. 1, 2025). https://hudoc.echr.coe.int/eng?i=001-242521
21. *Hurbain v. Belgium*, App. No. 57292/16, Eur. Ct. H.R. (Grand Chamber, July 4, 2023). https://hudoc.echr.coe.int/eng?i=001-225814
22. Richard Jean So, All Models Are Wrong, 132 *PMLA* 668 (2017), https://www.jstor.org/stable/27037381.

PYTHEN: A Flexible Framework for Legal Reasoning in Python

Nguyen Ha Thanh[1,2] and Ken Satoh[1]

[1] Center for Juris-Informatics, ROIS-DS, Tokyo, Japan
[2] Research and Development Center for Large Language Models, NII, Tokyo, Japan

Abstract. This paper introduces PYTHEN, a novel Python-based framework for defeasible legal reasoning. PYTHEN is designed to model the inherently defeasible nature of legal argumentation, providing a flexible and intuitive syntax for representing legal rules, conditions, and exceptions. Inspired by PROLEG (PROlog-based LEGal reasoning support system) and guided by the philosophy of The Zen of Python, PYTHEN leverages Python's built-in `any()` and `all()` functions to offer enhanced flexibility by natively supporting both conjunctive (ALL) and disjunctive (ANY) conditions within a single rule, as well as a more expressive exception-handling mechanism. This paper details the architecture of PYTHEN, provides a comparative analysis with PROLEG, and discusses its potential applications in autoformalization and the development of next-generation legal AI systems. By bridging the gap between symbolic reasoning and the accessibility of Python, PYTHEN aims to democratize formal legal reasoning for young researchers, legal tech developers, and professionals without extensive logic programming expertise. We position PYTHEN as a practical bridge between the powerful symbolic reasoning capabilities of logic programming and the rich, ubiquitous ecosystem of Python, making formal legal reasoning accessible to a broader range of developers and legal professionals.

Keywords: PYTHEN · legal reasoning · defeasible reasoning · non-monotonic reasoning.

1 Introduction

The field of AI and Law has long pursued the goal of creating computational models of legal reasoning. A central challenge in this endeavor is capturing the nuanced and defeasible nature of legal argumentation, where rules are seldom absolute and are frequently subject to exceptions and qualifications. Traditional programming paradigms often fall short in representing this complexity in a clear and maintainable way.

Logic programming, with Prolog as its most prominent example, has offered a powerful paradigm for this task. Systems like PROLEG (PROlog-based LEGal reasoning support system) have demonstrated the viability of using logic programming to model sophisticated legal theories, such as the Japanese Presupposed Ultimate Fact (JUF) Theory. These systems excel at handling the burden

of proof and reasoning with incomplete information, which are hallmarks of legal practice.

However, the adoption of Prolog-based systems has been hampered by several significant barriers. First, the steep learning curve of Prolog makes it inaccessible to many researchers and developers, particularly young researchers entering the field of legal AI who may lack formal training in logic programming. Second, the difficulty of integrating Prolog systems into the broader ecosystem of modern software development, which is largely dominated by languages like Python, creates a practical gap between the powerful theoretical models developed in the AI and Law community and the tools available to legal tech developers and practitioners. This creates a significant accessibility barrier that limits the adoption and impact of formal legal reasoning systems.

To address this gap, we propose **PYTHEN**[3], a Python-native framework for defeasible legal reasoning. PYTHEN is designed to be both powerful and easy to use, providing a clear and expressive syntax for defining legal rules and their exceptions. It allows for complex conditions (both `ALL` and `ANY`) within a single rule structure, a feature that enhances its flexibility compared to more rigid logic programming formalisms.

The design philosophy of PYTHEN is deeply influenced by The Zen of Python (PEP 20), which emphasizes principles such as "Beautiful is better than ugly," "Simple is better than complex," and "Readability counts." By adhering to these principles, PYTHEN ensures that legal rules are not only formally correct but also human-readable and maintainable, even for those without extensive training in formal logic.

This paper makes the following contributions:

1. It introduces the PYTHEN framework, detailing its rule and fact structure, and explaining how it leverages Python's built-in `any()` and `all()` functions.
2. It provides a comparative analysis of PYTHEN and PROLEG, highlighting the advantages of PYTHEN's flexible syntax and accessibility.
3. It discusses the potential of PYTHEN in the context of autoformalization and its integration with large language models (LLMs), particularly for converting natural language legal texts to structured rules.
4. It situates PYTHEN within the broader landscape of legal AI research, drawing connections to work from leading conferences and journals like ICAIL, JURIX, and the AI and Law journal, and discussing recent efforts to bridge NLP and formal legal reasoning.
5. It emphasizes the accessibility and democratization aspects of PYTHEN, particularly for young researchers and professionals without extensive logic programming expertise.

By leveraging the ubiquity and flexibility of Python, combined with its philosophical emphasis on simplicity and readability, PYTHEN aims to democratize the development of sophisticated legal reasoning systems, enabling a new generation of legal tech applications and making formal legal reasoning accessible to a broader audience.

[3] `https://github.com/nguyenthanhasia/pythen`

2 Related Work

The development of PYTHEN builds upon a rich history of research in AI and Law, particularly in the areas of logic-based legal reasoning, defeasible argumentation, natural language processing for legal texts, and autoformalization. This section reviews the key influences and situates our work within the current state of the art.

2.1 Logic-Based Legal Reasoning and PROLEG

The idea of representing legal statutes as a set of logical rules dates back to the early days of AI and Law. One of the most influential systems in this tradition is **PROLEG** (PROlog-based LEGal reasoning support system), developed by Ken Satoh and his colleagues [13]. PROLEG is an implementation of the Japanese "theory of presupposed ultimate facts" (JUF theory), a sophisticated model of judicial reasoning that deals with the allocation of the burden of proof under conditions of incomplete information. The system uses a Prolog-based engine to determine whether a legal claim is justified given a set of facts and rules. PROLEG's strength lies in its ability to model the defeasible nature of legal rules, where a conclusion can be overturned by exceptions.

Extensions to PROLEG, such as **ArgPROLEG** [14], have sought to integrate it with formal argumentation frameworks, making the reasoning process more transparent and explainable. These works have demonstrated the power of logic programming for modeling complex legal doctrines. However, their reliance on Prolog has limited their accessibility to the broader legal tech community and to young researchers without formal training in logic programming.

2.2 Defeasible Logic and Argumentation in Law

Legal reasoning is inherently defeasible, a point that has been extensively argued in the literature [10,18]. A legal conclusion is always provisional, subject to defeat by new evidence or arguments. This has led to the development of various formalisms for defeasible reasoning, such as Defeasible Logic [7] and formal argumentation theory [2]. These approaches provide a formal basis for modeling legal disputes as a dialogue between competing arguments, where the ultimate outcome depends on which arguments can withstand attack.

The International Conference on Artificial Intelligence and Law (ICAIL) and the JURIX conferences have been major venues for the presentation of research in this area. The work presented at these conferences has shown the importance of modeling not just the rules themselves, but also the process of argumentation and the allocation of the burden of proof [9]. PYTHEN draws on these insights by providing a clear and explicit mechanism for representing both conditions and exceptions within its rule structure.

2.3 Legal Ontologies and Knowledge Representation

The formalization of legal knowledge through ontologies has been a significant research direction. The LKIF Core Ontology [5] provides a foundational framework for representing basic legal concepts in a machine-readable format. Subsequent work on legal ontologies [16,17] has established methodologies for capturing legal domain knowledge in ways that support reasoning and knowledge reuse.

More recently, legal knowledge graphs have emerged as a powerful approach to representing and reasoning over legal information. Projects such as LYNX [11] demonstrate how semantic web technologies can be applied to legal data across multiple jurisdictions and languages. These approaches complement PYTHEN by providing structured representations of legal concepts that can be integrated with PYTHEN's rule-based reasoning engine.

2.4 Natural Language Processing and Autoformalization of Legal Text

A significant bottleneck in the development of rule-based legal AI systems has been the manual process of formalizing legal texts into a machine-readable format. **Autoformalization**—the automatic translation of natural language into a formal representation—has emerged as a promising solution to this problem [19,6]. The advent of large language models (LLMs) has brought new momentum to this area, with recent work demonstrating the potential of LLMs to translate mathematical and legal texts into formal specifications [1].

Recent efforts have specifically targeted the conversion of natural language legal texts into PROLEG format. The KRAG Framework [15] introduces Soft PROLEG, an extension of PROLEG designed to work with LLMs, using inference graphs to aid in structured legal reasoning. This work demonstrates that LLMs can be effectively guided to produce PROLEG-compatible outputs when provided with clear structural templates and examples.

Work on improving the translation of case descriptions into logical formulas [20] has shown that domain-specific named entity recognition (NER) techniques can significantly improve the quality of autoformalization. These techniques extract key legal entities and relationships from natural language text, which can then be mapped to formal rule structures.

PYTHEN's JSON-based syntax is particularly well-suited for serving as an intermediate representation in autoformalization pipelines. By providing a clear and well-defined target formalism, PYTHEN can serve as the "semantic backbone" for an autoformalization pipeline, where an LLM first generates a draft formalization of a legal text, which is then refined and validated against the PYTHEN rule structure. This hybrid approach, combining the generative power of LLMs with the rigor of symbolic reasoning, represents a promising direction for the future of legal AI [3].

2.5 Legal AI Benchmarks and Evaluation

The emergence of comprehensive benchmarks for legal AI has been crucial for advancing the field. LegalBench [4] provides a benchmark dataset for evaluating legal task specification and reasoning across multiple dimensions. The Cambridge Law Corpus [8] offers a large-scale dataset for legal AI research, with annotations and benchmarks for case outcome prediction and other legal reasoning tasks.

These benchmarks highlight the challenges that legal AI systems must address, including the need for precise knowledge representation, handling of complex legal concepts, and the ability to reason with incomplete or ambiguous information. PYTHEN's design is informed by these challenges and aims to provide a framework that can support the development of systems capable of handling such complexity.

2.6 Hybrid Approaches: Combining Symbolic and Neural Methods

Recent research has explored the integration of rule-based symbolic reasoning with neural network approaches [12]. These hybrid systems leverage the interpretability and logical soundness of symbolic methods with the pattern recognition capabilities of neural networks. PYTHEN is positioned as a tool that can facilitate such hybrid approaches, providing a clear interface between symbolic legal reasoning and neural methods like LLMs.

The integration of PYTHEN with LLMs through autoformalization pipelines represents a modern instantiation of this hybrid approach, where LLMs generate candidate rule structures that are then validated and refined using PYTHEN's formal reasoning engine.

3 The PYTHEN Framework

PYTHEN is designed with the primary goal of making the formalization of legal rules both intuitive for humans and computationally tractable. It achieves this through a simple, JSON-based structure for representing rules and facts, which can be easily created, parsed, and manipulated using standard Python libraries. The design is guided by the principles of The Zen of Python, particularly the emphasis on simplicity, readability, and accessibility.

3.1 Design Philosophy: The Zen of Python

PYTHEN's design is deeply influenced by The Zen of Python (PEP 20), a set of guiding principles for Python development. Key principles that guide PYTHEN's design include:

- **Beautiful is better than ugly**: PYTHEN uses a clean, JSON-based syntax that is visually clear and easy to understand.
- **Simple is better than complex**: Rather than requiring users to learn Prolog syntax, PYTHEN uses familiar Python concepts.

– **Readability counts**: Legal rules in PYTHEN are self-documenting and easy to understand, even for those without formal logic training.
– **Explicit is better than implicit**: PYTHEN makes conditions, exceptions, and reasoning steps explicit and visible.
– **Practicality beats purity**: PYTHEN prioritizes usability and integration with modern tools over strict formal purity.

These principles ensure that PYTHEN is not only formally sound but also accessible and practical for real-world legal tech applications.

3.2 Rule Structure

A PYTHEN rule is a dictionary that defines a legal proposition and the conditions under which it holds true. Each rule consists of the following key-value pairs:

– ``p'': The **proposition** that the rule defines. This is a unique string identifier for the legal concept being modeled (e.g., ``art17_erasure_applicable'').
– ``op'': The **operator** that governs the relationship between the conditions. It can be either ``ALL'' (conjunctive), meaning all conditions must be met, or ``ANY'' (disjunctive), meaning at least one condition must be met. This design is inspired by Python's built-in `all()` and `any()` functions, which are familiar to any Python developer.
– ``conditions'': A list of strings, where each string is a proposition that must be evaluated to determine the truth of the main proposition `p`. These can be references to other rules or to basic facts.
– ``exceptions'': A list of strings representing propositions that, if true, will defeat the main proposition, even if its conditions are met. This provides a direct mechanism for modeling defeasibility.

The use of ``ALL'' and ``ANY'' operators directly mirrors Python's built-in functions, making the semantics immediately clear to Python developers. This design choice significantly enhances accessibility for developers who may not be familiar with formal logic notation.

Below is an example of a rule from a PYTHEN rule-base, modeling a part of Article 17 of the GDPR (the "right to be forgotten"):

```
{
    "p": "art17_erasure_applicable",
    "op": "ANY",
    "conditions": [
        "no_longer_necessary",
        "consent_withdrawn",
        "object_to_processing",
        "processing_unlawful",
        "child_data_collected"
    ],
```

```
    "exceptions": [
        "freedom_of_expression",
        "legal_obligation",
        "public_interest_archiving_research",
        "legal_claims"
    ]
}
```

In this example, the proposition `art17_erasure_applicable` is true if `ANY` of its conditions are met (e.g., if consent is withdrawn), provided that none of the `exceptions` (e.g., the data is needed for a legal claim) are true.

3.3 Fact Base

The fact base is a simple list of strings, where each string represents a basic proposition that is considered to be true for a given case. These are the foundational inputs to the reasoning process.

Example Fact Base:

```
[
    "objection_to_direct_marketing",
    "data_collected_from_child",
    "consent_was_basis",
    "consent_is_withdrawn",
    "data_not_needed_for_purpose"
]
```

3.4 Reasoning Mechanism

The PYTHEN reasoner is a goal-driven engine that works backward from a target proposition. To determine whether a proposition `p` holds, the engine applies an evaluation strategy over rules, conditions, and exceptions. In the current implementation, the following steps are performed:

1. **Exception Evaluation**: The engine first checks whether any proposition listed in the `exceptions` of `p` can be derived. If at least one exception is found to hold, the reasoning process for `p` terminates and `p` is rejected. This exception-first evaluation constitutes a decision-oriented strategy that allows the reasoner to efficiently compute a final judgement in defeasible settings.
2. **Condition Evaluation**: If no exception is derived, the engine evaluates the `conditions` according to the specified operator `op`.
   - If `op` is ‘‘`ALL`’’, the engine recursively attempts to establish that all propositions in the `conditions` list hold (corresponding to Python's `all()` semantics).
   - If `op` is ‘‘`ANY`’’, the engine recursively attempts to establish that at least one proposition in the `conditions` list holds (corresponding to Python's `any()` semantics).

3. **Base Case**: The recursion terminates when a proposition is found in the fact base, in which case it is considered true, or when no rule is available to derive it, in which case it is considered false.

It is important to note that the choice of evaluation order affects the structure of the reasoning process and its explanatory trace, but not the logical outcome of the derivation under standard assumptions of defeasible reasoning. While legal practitioners often reason by first establishing the applicability of general rules and subsequently considering exceptions, the exception-first strategy adopted here prioritizes efficient judgement computation. Alternative evaluation orders, including general-rule-first and explanation-oriented strategies, can be supported without changing the underlying rule representation.

3.5 Evaluation Strategy and Computational Considerations

An important design choice in defeasible legal reasoning systems concerns the evaluation strategy used to determine whether a proposition holds. While different evaluation orders—such as general-rule-first or exception-first—are often logically equivalent in terms of their final outcomes, they may differ significantly in their computational behavior and explanatory characteristics.

In practice, the computational cost of reasoning over general rules and over exceptions may vary substantially depending on the structure of the rule base, the depth of rule dependencies, and the availability of factual information. General-rule reasoning may involve the evaluation of multiple conjunctive or disjunctive conditions, while exception reasoning may terminate early if a single defeating condition can be established. As a result, a fixed evaluation order may be suboptimal in terms of efficiency across different legal domains and cases.

From a system-oriented perspective, evaluation strategy can therefore be viewed as an execution policy rather than a property of the rule representation itself. PYTHEN explicitly separates the representation of legal rules from their evaluation order, allowing alternative strategies to be explored without modifying the underlying rule base. This design enables the incorporation of computational considerations, such as the relative difficulty of general-rule reasoning versus exceptional reasoning, into the reasoning process.

More adaptive strategies are also possible. For example, general-rule reasoning and exception reasoning can be executed concurrently or in parallel, allowing the reasoning process to terminate as soon as a decisive result is obtained. In such settings, dynamic allocation of computational resources between competing reasoning paths can serve as a practical approximation of computational hardness prediction.

From a jurisprudential perspective, the distinction between general rules and exceptions is not absolute, but rather a methodological device for organizing issues and structuring fact-finding. Legal practice suggests that the preferred order of examination depends on context and legal domain. In civil law, anticipatory judgments based on defenses may be theoretically possible, but they are rarely adopted in practice due to the close interdependence between facts underlying

claims and defenses. In criminal law, by contrast, the order of judgment may carry normative significance, making certain anticipatory evaluations inappropriate, as determinations regarding unlawfulness or culpability have independent legal meaning. These observations further support the view that evaluation order should be treated as a context-sensitive strategy rather than a fixed semantic property of legal rules.

4 Comparison with PROLEG

While PYTHEN is inspired by the pioneering work of PROLEG, it introduces several key differences in its design and philosophy, aimed at increasing flexibility, accessibility, and practical usability. This section provides a comparative analysis of the two frameworks.

4.1 Syntax and Expressiveness

PROLEG, being based on Prolog, inherits its syntax, which is powerful but can be opaque to those not trained in logic programming. A PROLEG rule is typically expressed as a Prolog clause, which, while formally precise, does not always map intuitively to the way a lawyer might articulate a rule. This creates a significant barrier to adoption, particularly for young researchers and legal professionals without formal training in logic.

PYTHEN, in contrast, uses a simple, human-readable JSON structure. This has multiple significant advantages:

1. **Accessibility**: The JSON format is universally understood and can be easily generated and parsed by virtually any programming language. This lowers the barrier to entry for developers and legal professionals who may not be familiar with Prolog or formal logic programming. This is particularly important for young researchers entering the field of legal AI.
2. **Flexibility in Conditions**: A key innovation in PYTHEN is the explicit `''op''` field, which allows a rule's conditions to be either conjunctive (`''ALL''`) or disjunctive (`''ANY''`). In PROLEG, achieving disjunctive conditions typically requires writing multiple, separate rules for the same proposition. PYTHEN's approach allows for a more compact and, in many cases, more natural representation of legal rules that involve alternative paths to a conclusion.
3. **Integration with Python Ecosystem**: By using Python's `any()` and `all()` semantics, PYTHEN creates an intuitive bridge between formal logic and Python programming, making it immediately familiar to the millions of Python developers worldwide.

4.2 Exception Handling

Both PROLEG and PYTHEN are designed to model the defeasible nature of legal rules, but they adopt different design perspectives regarding the representation and evaluation of exceptions. PROLEG handles exceptions through negation

as failure and rule ordering within a Prolog-based evaluation framework. This approach is closely tied to legal-theoretical considerations and provides well-defined semantics, but it may result in interactions between rules that are not immediately visible at the level of individual rule definitions.

PYTHEN, in contrast, adopts an explicit representation of defeasibility by associating each rule with a dedicated ‘‘`exceptions`’’ list. This design emphasizes transparency at the knowledge-representation level: the conditions under which a proposition can be defeated are localized within the rule itself, making defeasible relationships directly observable. Importantly, PYTHEN treats the representation of exceptions as orthogonal to the choice of evaluation strategy. Different execution orders or reasoning strategies can be applied without modifying the rule base, allowing the same representation to support both decision-oriented and explanation-oriented reasoning.

4.3 Ecosystem and Integration

Perhaps the most significant practical difference lies in the ecosystems of the two frameworks. PROLEG is embedded in the Prolog ecosystem, which, while powerful for symbolic reasoning, is relatively isolated from the mainstream of modern software development.

PYTHEN, being a native Python framework, has immediate access to the vast and rich Python ecosystem. This includes:

- **Data Science and NLP Libraries**: Seamless integration with libraries like Pandas, NLTK, and spaCy for pre-processing legal documents and extracting facts. This facilitates the development of end-to-end legal AI systems that combine NLP with formal reasoning.
- **Machine Learning Frameworks**: The ability to connect with PyTorch and TensorFlow for building hybrid models that combine symbolic reasoning with machine learning, enabling the development of more sophisticated legal AI systems.
- **Web Frameworks**: Easy integration with Django and Flask for building interactive legal tech applications and APIs, making it practical to deploy legal reasoning systems in real-world applications.
- **LLM Integration**: Direct access to libraries for interacting with large language models like GPT-4 and Claude, facilitating the development of autoformalization pipelines that convert natural language legal texts into PYTHEN rules.

This ease of integration makes PYTHEN a more practical choice for building end-to-end legal AI systems in a modern technology stack, and significantly lowers the barrier to adoption for young researchers and legal tech developers.

4.4 Accessibility and Democratization

A critical advantage of PYTHEN over PROLEG is its accessibility to researchers and developers without extensive formal logic training. The legal AI field is

rapidly growing, and many young researchers are entering from backgrounds in NLP, machine learning, or software engineering rather than formal logic or Prolog. PYTHEN's design explicitly addresses this gap by using familiar Python concepts and JSON syntax.

Furthermore, PYTHEN's integration with LLMs through autoformalization pipelines makes it possible for researchers to work with formal legal reasoning without manually writing Prolog code. This democratization of formal legal reasoning is a significant contribution to making the field more accessible and inclusive.

4.5 A Comparative Example

Consider a simplified rule: "A contract is voidable if the person was a minor OR was mentally incapable, UNLESS the contract was for necessities."

In **PROLEG**, this might be represented by separate rules:

```
contract_voidable(C) :- minor(P), party_to(P, C).
contract_voidable(C) :- incapable(P), party_to(P, C).
exception(contract_voidable(C),for_necessities(C)).
```

In **PYTHEN**, this can be represented in a single, more intuitive rule:

```
{
    "p": "contract_voidable",
    "op": "ANY",
    "conditions": ["minor", "incapable"],
    "exceptions": ["for_necessities"]
}
```

This simple example illustrates how PYTHEN's syntax can lead to a more compact and readable formalization of legal rules, particularly those involving disjunctive conditions. Moreover, a developer without Prolog experience can immediately understand the PYTHEN rule, while the Prolog version requires knowledge of Prolog syntax and semantics.

5 Applications and Use Cases

The flexibility and accessibility of PYTHEN open up a wide range of potential applications in legal tech and computational law, particularly for young researchers and developers entering the field.

5.1 Autoformalization and LLM-Powered Systems

PYTHEN is well-suited to serve as the target formalism for autoformalization pipelines. An LLM can be prompted to translate a piece of legal text (e.g., a clause from a contract or a section of a statute) into a PYTHEN rule structure.

The structured nature of PYTHEN makes it easier to validate and debug the output of the LLM, and the resulting rule base can be executed with the PYTHEN reasoner to ensure logical consistency. This creates a powerful synergy between the generative capabilities of LLMs and the formal rigor of symbolic reasoning.

Young researchers can leverage pre-trained LLMs to rapidly prototype legal reasoning systems without needing to manually write formal rules, significantly lowering the barrier to entry for the field.

5.2 Compliance and Regulatory Analysis

Companies in highly regulated industries, such as finance and healthcare, can use PYTHEN to model complex regulatory requirements. For example, a rule base could be created to determine whether a particular financial transaction complies with anti-money laundering (AML) regulations, or whether a proposed use of patient data complies with HIPAA. The clear, JSON-based rules would be auditable by compliance officers, and the reasoning engine could be integrated into business workflows to provide real-time compliance checks.

5.3 Legal Education and Training

PYTHEN can be a valuable tool for legal education and training. Law students and young researchers could learn about the logical structure of legal rules by creating their own PYTHEN rule bases for specific areas of law. This would provide them with a hands-on understanding of concepts like conditions, exceptions, and the burden of proof. The interactive nature of the framework would allow them to test different factual scenarios and see how they affect the legal outcome. The simplicity of PYTHEN's syntax makes it accessible to students without extensive programming background.

5.4 Contract Analysis and Management

By formalizing the clauses of a contract into a PYTHEN rule base, it becomes possible to automatically analyze the contract for potential issues, such as conflicting clauses or missing provisions. For example, a rule base could be created to check if a software license agreement contains adequate data protection clauses. This could be integrated into contract lifecycle management (CLM) systems to provide automated risk assessment.

6 Discussion and Future Work

PYTHEN shows that defeasible legal reasoning can be operationalized in a way that is both practical and compatible with contemporary AI systems. By embedding formal reasoning directly into Python, PYTHEN aligns legal reasoning with data-driven and model-centric workflows, including large language models.

Future work will focus on scaling and orchestration. First, we will study resource allocation strategies for legal reasoning pipelines, including when and how to invoke symbolic reasoning versus statistical or neural components. Second, we will investigate divide-and-conquer approaches that decompose complex legal problems into smaller, independently solvable subproblems, improving efficiency and modularity. Third, we will explore agentic AI settings in which multiple specialized agents—such as text interpretation, rule induction, and defeasible reasoning agents—coordinate using PYTHEN as a shared reasoning substrate. These directions aim to position PYTHEN as a core component in hybrid, multi-agent legal AI systems.

7 Conclusion

This paper introduced PYTHEN, a Python-based framework for defeasible legal reasoning that emphasizes simplicity, extensibility, and practical integration. By lowering the barrier to formal legal reasoning and enabling tight coupling with modern AI workflows, PYTHEN supports the development of more transparent, scalable, and reliable legal AI systems. From a theoretical perspective, PYTHEN also provides a concrete platform for studying meta-reasoning issues such as conflict management, theory revision, and control of reasoning resources in hybrid and agent-based legal AI architectures.

Acknowledgements

This work was supported by 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 by JST as part of Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE), Grant Number JPMJAP25B2.

References

1. Ariai, F., Mackenzie, J., Demartini, G.: Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges. ACM Computing Surveys **58**(6), 1–37 (2025)
2. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence **77**(2), 321–357 (1995)
3. Fuchs, S., Dimyadi, J., Witbrock, M., Amor, R.: Intermediate representations to improve the semantic parsing of building regulations. Advanced Engineering Informatics **62**, 102735 (2024)
4. Guha, N., Nyarko, J., Ho, D., Ré, C., Chilton, A., Chohlas-Wood, A., Peters, A., Waldon, B., Rockmore, D., Zambrano, D., et al.: Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. Advances in neural information processing systems **36**, 44123–44279 (2023)

5. Hoekstra, R., Breuker, J., Di Bello, M., Boer, A., et al.: The lkif core ontology of basic legal concepts. LOAIT **321**, 43–63 (2007)
6. Mensfelt, A., Cucala, D.T., Franco, S., Koutsoukou-Argyraki, A., Trencsenyi, V., Stathis, K.: Towards a common framework for autoformalization. arXiv preprint arXiv:2509.09810 (2025)
7. Nute, D.: Defeasible logic. In: International Conference on Applications of Prolog. pp. 151–169. Springer (2001)
8. Östling, A., Sargeant, H., Xie, H., Bull, L., Terenin, A., Jonsson, L., Magnusson, M., Steffek, F.: The cambridge law corpus: A dataset for legal ai research. Advances in Neural Information Processing Systems **36**, 41355–41385 (2023)
9. Prakken, H., Sartor, G.: Formalising arguments about the burden of persuasion. In: Proceedings of the 11th international conference on artificial intelligence and law. pp. 97–106 (2007)
10. Prakken, H., Sartor, G.: Law and logic: A review from an argumentation perspective. Artificial intelligence **227**, 214–245 (2015)
11. Rodríguez-Doncel, V., Montiel-Ponsoda, E.: Lynx: Towards a legal knowledge graph for multilingual europe. Law Context: A Socio-Legal J. **37**, 175 (2020)
12. Sadowski, A., Chudziak, J.A.: On verifiable legal reasoning: A multi-agent framework with formalized knowledge representations. In: Proceedings of the 34th ACM International Conference on Information and Knowledge Management. pp. 2535–2545 (2025)
13. Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., Takano, C.: Proleg: an implementation of the presupposed ultimate fact theory of japanese civil code by prolog technology. In: JSAI international symposium on artificial intelligence. pp. 153–164. Springer (2010)
14. Shams, Z., De Vos, M., Satoh, K.: Argproleg: A normative framework for the juf theory. In: JSAI International Symposium on Artificial Intelligence. pp. 183–198. Springer (2013)
15. Thanh, N.H., Satoh, K.: Krag framework for enhancing llms in the legal domain. arXiv preprint arXiv:2410.07551 (2024)
16. Valente, A.: Types and roles of legal ontologies. In: Law and the Semantic Web: Legal ontologies, methodologies, legal information retrieval, and applications, pp. 65–76. Springer (2005)
17. Valente, A., Breuker, J., et al.: A functional ontology of law. Towards a global expert system in law pp. 112–136 (1994)
18. Walton, D.: Argumentation methods for artificial intelligence in law. Springer (2005)
19. Wu, Y., Jiang, A.Q., Li, W., Rabe, M., Staats, C., Jamnik, M., Szegedy, C.: Autoformalization with large language models. Advances in neural information processing systems **35**, 32353–32368 (2022)
20. Zin, M.M., Nguyen, H.T., Satoh, K., Nishino, F.: Addressing annotated data scarcity in legal information extraction. In: JSAI International Symposium on Artificial Intelligence. pp. 77–92. Springer (2024)

Legal-Proofing LLMs: Investigating Legal Applications of Prover Verification Models

Madeleine Pelli

Emory University mpelli@emory.edu

Abstract. Large Language Models (LLMs) have shown remarkable results on legal tasks, but their reasoning remains trapped inside the "black box," raising credibility concerns in high-stakes domains like the legal sphere. While broad research on obtaining better results from LLMs in legal applications is ongoing, a lack of a verifiable component to these models leaves the precision-based legal industry hesitant to accept models that may be unreliable, even to a small degree. This paper explores a novel approach that bridges natural-language legal reasoning with formal verification by leveraging *prover LLMs*, inspired by recent successes in automated theorem proving. Proof-verification frameworks such as Apple's Hilbert and DeepSeek-Prover have demonstrated that combining informal reasoning with formal proof checking can dramatically improve reliability in mathematical problem solving [11,9]. I propose an adaptation of this paradigm to legal reasoning. I outline a dual LLM architecture comprising a general-purpose legal "reasoner" and a formal logic "prover," augmented by a verifier and legal knowledge context. This system can translate legal problems into a form amenable to formal proof, verify each step for correctness, and thereby produce interpretable, verifiable legal conclusions. I describe the design of such a system, situate it in the context of prior legal AI efforts, and present initial experiments showing that a prover-augmented model achieves higher accuracy on certain legal benchmarks than standard LLMs. These results indicate that applying formal theorem-proving techniques to legal reasoning is a promising direction to increase the rigor and trustworthiness of AI in the legal domain.

Keywords: legal reasoning · theorem proving · verification · large language models · prover LLMs

1 Introduction

Modern legal practice involves complex reasoning over statutes, regulations, and precedents expressed in natural language. The prospect of AI systems that can reason "like a lawyer" holds great promise for improving access to justice and legal decision support. Large Language Models have recently been applied to legal tasks with some success [6,8,5], yet significant challenges remain. Even state-of-the-art LLMs tend to falter on advanced legal reasoning problems, struggling with domain-specific knowledge, multi-step logical deductions, and accuracy in application of rules [5,7,3,4]. Critically, their outputs lack verifiability:

an LLM might produce a confident-sounding legal conclusion, but without an explicit proof or logical justification, it is difficult to trust such an answer in real courtroom settings. This absence of transparency and formal correctness limits the credibility of black-box LLMs for broader application in legal use, where precision can cost millions.

Academic research into how to bridge this gap in legal reasoning has not given up on this endeavor, however. Attempts at formalizing legal logic into computer-parseable syllogisms have been attempted for decades, and some standard practices have been found. An early example is the representation of the *British Nationality Act* as a Prolog logic program by Sergot et al [10]. This seminal work demonstrated that entire statutes can be encoded as formal rules, enabling automated inference that is provably sound and explainable. Such logic-based approaches offer clear advantages in terms of transparency and the ability to explain and modify laws. However, manual formalization of law is labor-intensive and these systems historically lacked the flexibility and linguistic capabilities of modern LLMs.

Meanwhile, in the field of automated theorem proving, a new generation of prover LLMs has emerged. Models like HILBERT [11] and DeepSeek-Prover [9] have achieved state-of-the-art results on challenging mathematical benchmarks by combining natural language reasoning with formal verification. HILBERT, for example, solves 99.2% of problems on the MiniF2F math benchmark and 70.0% of Putnam competition problems, dramatically outperforming prior methods by integrating an LLM "reasoner" with a Lean theorem prover and recursive proof checking. DeepSeek-Prover-V2 (henceforth DSP for brevity) similarly leverages a two-phase reasoning (informal and formal) to set new records on formal math datasets. These successes suggest that a hybrid approach, using the creative problem-solving ability of LLMs together with the rigor of formal proof systems, can significantly improve the reliability of AI reasoning.

I posit that such prover-augmented LLM frameworks can be transformative for legal reasoning tasks, an area where both natural language understanding and logical consistency are paramount. To date, however, there has been no systematic application of formal theorem-proving LLMs in the legal domain. In this paper, we take a first step toward bridging that gap. I outline a multi-tiered legal reasoning agent inspired by HILBERT and DSP, consisting of a general-purpose legal reasoning model and a specialized prover model that work in tandem. By encoding legal problems into formal representation and verifying each deductive step, such a system can ensure that its conclusions follow rigorously from given laws and facts. This approach could dramatically reduce hallucinations and logical errors, providing the kind of step-by-step justification a human judge or attorney would expect. It also offers a pathway to integrate vast natural-language legal corpora (cases, statutes) with formal reasoning engines, potentially yielding AI that not only *reads* legal text but also *proves* legal conclusions from it.

In the following sections, I survey related work at the intersection of AI and law, delineate the design of our proposed prover-augmented legal reasoning model, and present preliminary experiments. I evaluate a proof-of-concept using

DSP on several LegalBench tasks to gauge the viability of this approach [5]. My results indicate that even without domain-specific tuning, the prover-based model can outperform a general LLM on many legal reasoning problems, especially those requiring strict logical inference. Finally, I will discuss the broader implications of formal proof-guided LLMs for legal AI, highlighting how this paradigm could enhance the transparency, accuracy, and ultimately the credibility of automated legal reasoning systems.

2 Related Work

Before the rise of LLM research, legal AI researchers pursued formal systems that could represent statutes and apply them transparently. A canonical example is Sergot et al.[10], which translated large portions of the British Nationality Act into a Prolog (logic-verifier programming language) program. They proved that legal rules can be executed mechanically and that a logic-based encoding can preserve the structure of legislation closely enough to be auditable and maintainable, while also producing step-by-step explanations of derived conclusions. In parallel, case-based reasoning systems (e.g., Ashley's *HYPO* [2]) modeled legal argument through a comparison of precedents, offering another structured route to interpretable conclusions.

Yet building and updating hand-crafted representations of the law is expensive, as the nuanced, evolving character of legal language complicates comprehensive and precise coverage. This lack of scalability motivated the shift toward statistical NLP and machine learning, which can absorb large legal corpora without requiring explicit rule-by-rule encoding. The resulting tradeoff, however, is familiar: learned models may be powerful, but they often struggle to provide the kind of explicit, verifiable reasoning that lawyers demand.

The recent wave of legal benchmarks has made this tradeoff measurable. Guha et al. [5] introduced LegalBench, a large suite of expert-authored English tasks spanning statutory reasoning, case analysis, and legal text understanding. Their evaluations show that strong general-purpose models can perform well on many subtasks, but also leave substantial headroom on multi-step reasoning and difficult application questions. Complementing this, Fei et al. [4] proposed LawBench for Chinese law and emphasized a civil-law setting where applying codified rules is central; their results likewise suggest that higher-order application remains challenging even for top models. Across benchmarks and jurisdictions, a consistent picture emerges: LLMs handle surface-level understanding and some forms of retrieval effectively, but they remain untrustworthy where precise rule application, exception handling, and logically consistent inference are required. What is missing is a mechanism that can *enforce* correctness on a model, and outline where its logic first falters.

In response, a growing literature explores hybrid methods that combine neural language understanding with logical constraints. Kant et al. [8] offers a representative view: they argue that trustworthy legal AI requires repeatability, interpretability, and verification, and they survey methods that structure model

reasoning through explicit logical decomposition and rule-guided pipelines. One prominent theme is to break complex legal questions into smaller logical subqueries and recombine them, aiming to reduce leaps of inference and improve consistency. Another theme is controlled or semi-structured representations, such as domain-specific controlled natural languages for contracts, that preserve legal semantics while making downstream reasoning more systematic. A third theme couples LLM generation with logic engines like Prolog or Lean 4, using the symbolic component as a validator that can catch contradictions, missing premises, or unsupported inferences.

These approaches all share an implicit design philosophy: legal reasoning improves when we impose *structure* and *checks* on model outputs. Our proposal adopts the same philosophy but pushes it further by asking whether legal reasoning can be grounded in formal proof construction rather than only rule-guided prompting or post-hoc validation, so that the model's reasoning can be verifiable step-by-step.

2.1 Prover LLMs in mathematics as a template for "legal proofing"

The strongest evidence that proof-guided generation can transform legal LLM reliability comes from recent advances in formal theorem proving. DeepSeek-Prover [9] separates the problem solving process into two modes of thought: Chain-of-Thought (CoT) and non-Chain-of-Thought (non-CoT). Chain-of-Thought mode consists of an informal reasoning component which decomposes problems and proposes proof sketches. In non-CoT mode, a specialized prover component solves sub-goals in Lean 4 and a verifier checks correctness. Similarly, Hilbert [11] frames theorem proving as an agentic workflow that orchestrates an informal reasoner, a formal prover, a verifier, and a retriever of relevant theorems. These systems matter for our purposes because they show a concrete route to narrowing the gap between fluent natural-language reasoning and formally checkable inference: when verification is in the loop, the system can iteratively repair failures, decompose hard steps, and converge on proofs that are not merely persuasive but correct by construction.

Although these frameworks were developed for mathematics, their architectural lesson is domain-general: pair a model that can interpret and plan in natural language with a component that can express and verify reasoning in a formal system, and use verifier feedback to discipline the overall process. Our work adapts this template to law by asking whether we can build a "Legal Hilbert" that retrieves relevant authorities (statutes, regulations, precedents or their formalizations), proposes a structured legal argument in natural language, and then encodes that argument into a formal representation where each step can be verified. If successful, this would address a central limitation surfaced by legal benchmarks: not only improving accuracy, but producing conclusions that are audit-able, defensible, and resistant to hallucination.

3 Proposed Framework: A Dual LLM for Legal Proofs

To my knowledge, there remains no end-to-end system that brings the recent prover-LLM paradigm into legal reasoning in a way that is both practically useful and formally verifiable. I therefore outline an idealized framework, what I will refer to as a *Legal-Proofing LLM* (LP LLM, for brevity), by borrowing the central architecture from mathematical theorem-proving systems and adapting it to a legal context.

3.1 System Architecture and Components

LP would be built around two tightly coupled models and two supporting modules. The first component would consist of a *legal reasoner* LLM which operates in natural language; it would read the user's question and fact pattern, identifies any legally-relevant issues, and sketches a structured route to an answer (for example, recognizing when analysis should proceed element-by-element, as in contract formation or a statutory test). This mirrors the role played by the informal reasoning component in theorem-proving agents, which produces high-level proof plans and decompositions that a formal system can attempt to discharge. The second component would be a *prover* LLM which would use a symbolic language (Lean-style proofs, logic programs, or another legal "logic") so that the reasoning can be expressed as precise claims about rules and facts. It would focus on translating the reasoner's proposed steps into formal sub-goals and work to prove them in a logically-sound manner.

The remaining two modules enforce correctness and supply grounding. A *formal verifier* (e.g., a proof assistant or a logic inference engine) accepts or rejects proof steps, supplying concrete feedback when an attempt fails, which can be routed back to the models for correction and refinement. This verifier forces the system to confront missing premises, invalid implications, or misapplied rules. Finally, a *legal knowledge retriever* supplies legal reasoning authorities such as relevant statutes, regulations, and cases. Conceptually, it plays the same role that theorem retrieval plays in systems like Hilbert: it turns a generic LLM's reasoning capability into inference specific to the legal field. In the long run, one could imagine building a law-analog of `mathlib`, seeded by classic formalizations like the *British Nationality Act* [10] logic program and expanded through incremental formal encodings and extraction methods for tools like LP.

3.2 Algorithmic Approach

Operationally, LP proceeds as an iterative proof construction process. DeepSeek-Prover has the clearest algorithmic method that would serve well in legal analysis due to its recursive nature, breaking down each stance into smaller and smaller lemmas, until single components can be known to be true. The first mode that would be relevant is the Chain-of-Thought mode, using the reasoner outlined in the previous sub-section; this mode would lay out a sketch of how to solve a legal task and recursively break down each problem into sub-problems to check,

verifying that these sub-problems are feasible to solve before moving into the second non-CoT mode; this mode breaks down each recursed lemma into a feasible proof using the prover LLM. This proof would be verifiable using a language like Lean 4.

Emphasizing legal formalism in this model is itself a practical design choice; the right encoding for a given task will largely depend on the target domain: highly rule-bound areas (eligibility determinations, compliance checks, benefit calculations) are often amenable to crisp encodings, while open-textured standards (reasonableness, proportionality, balancing tests) may resist full formalization. LP is therefore best understood as a formalized solver that hopes to help across the spectrum of tasks: even partial formalization could substantially improve reliability by shrinking the space where black-box hallucinations can hide.

4 Experiments

To test whether prover-style reasoning can transfer to law, I ran preliminary experiments using DeepSeek-Prover on a variety of LegalBench benchmarks. The goal was not to assume an ideal legal prover system, but to probe a concrete question: how well do formal prover models handle legal reasoning problems without any fine-tuning, and how do they compare to strong conventional LLM baselines that run on the same queries?

4.1 Experimental Setup

I evaluated **DeepSeek-Prover-V2 (671B)** [1] as a proxy for a prover-augmented legal model. Although it is trained for Lean-based mathematical proof generation rather than legal doctrine, its emphasis on structured decomposition and stepwise correctness makes it a useful stand-in for my proposed paradigm. **GPT-4** served as a high-performing general-purpose model that demonstrated strong results on LegalBench benchmarks, but lacks formal checking.

My benchmark tasks were drawn from **LegalBench** [5]. I focused on English-only tasks with objectively scorable outputs—multiple choice, binary decisions, or classification labels—so that performance could be measured automatically without subjective grading. I also prioritized tasks that require rule application or multi-step reasoning rather than simple retrieval, including statutory yes/no determinations, contract clause interpretation, and other structured legal QA formats. After filtering, I tested the models on roughly 25 distinct tasks, sampling up to 100 queries per task when available.

All models were evaluated in a zero-shot (no-exposure) setting first. Each prompt consisted of the task description and a query, with an instruction to output only the final answer. DeepSeek-Prover required additional prompt shaping to cast legal questions as proof-like objectives (e.g., framing the task as proving whether the conclusion is “Yes” or “No” given stated facts and rules), while

Table 1: Performance Comparison on LegalBench Subsets.

Model	Issue	Rule	Conclusion	Interpretation	Rhetorical
DeepSeek-Prover-v2	**98.6**%	**82.8**%	**97.3**%	**80.9**%	N/A
GPT-4	82.9%	59.2%	89.9%	75.2%	**79.4**%

GPT-4 was prompted more directly. I then ran a small three-shot condition (three pre-exposures) to test whether showing even a limited number of exposures improved reliability on the prover model. Accuracy was computed by exact match against the benchmark. I limited my experiment to three pre-exposures to see how a "bare" prover model operates on legal tasks.

5 Results

Aggregated results by reasoning category appear in Table 1, with per-task breakdowns in Table 2 (0-shot) and Table 3 (3-shot). Two caveats frame how these numbers should be read. First, I only tested a minimal exposure regime (n=0 and n=3 exposures), which likely leaves the prover-oriented model well below its ceiling. Second, DeepSeek-Prover was never fine-tuned to produce *legal* proofs; it is a math-trained prover used here as a proxy, so future legal finetuning would also likely increase performance.

Even under these constraints, the prover-based model (DSP) leads in the most logic-intensive categories. In Table 1, DSP achieves the highest accuracy in all four of the reasoning types it attempted, with particularly large margins on issue spotting and rule selection, precisely the stages that resemble element-by-element proof checking in legal analysis. Interpretation, however, has the most promising results, as it was the category with the most number of tasks (23), and still outperformed the GPT baseline on all fronts. In the per-task analysis, DSP outperforms GPT across many baselines.

Where DSP underperforms, the errors are typically consistent with missing domain knowledge rather than incoherent logic. The trademark distinctiveness task (Abercrombie) is a clear example: GPT-4 likely benefits from memorized doctrinal categories and examples, while DSP, lacking legal fine-tuning, misapplies or misidentifies the relevant standard. Contract entailment tasks show a different dynamic: in zero-shot, DSP can appear behind in performance, but with only three exposures added (Table 3), it improves sharply on many NDA-style NLI subtasks, suggesting that small amounts of task-specific grounding help the prover model map legal language to the intended decision boundary. This supports a central claim of the paper: a prover-oriented reasoning engine has strong latent structure, but it needs legal semantics in order to reach full capacity.

Overall, the results should be read as a proof-of-concept rather than a final system comparison. I find it not insignificant that a math-trained prover model,

evaluated with minimal prompting and without an actual legal proof environment, nonetheless performs strongly on rule-application-heavy legal tasks. To see potentially further significance, a combination of a legal-specific verifier, stronger legal grounding through retrieval or formal rule libraries, and training a prover explicitly for legal-logic lemmas would likely be a fruitful next step.

6 Discussion

This preliminary experiment suggests that using proof-style verification in legal reasoning LLMs can improve reliability in exactly the places where legal users care most: element-by-element rule application, improved consistency, and the ability to justify an output. My hope is that models forced into this disciplined workflow are less likely to produce plausible but unsupported rationales. This mirrors what has been observed in mathematical reasoning: without a verifier in the loop, even strong LLMs can drift into confident errors, whereas verification pressure grounds model reasoning.

Beyond raw accuracy, the central payoff is the ability to audit the prover model. A prover-augmented legal system can, in principle, expose its reasoning as a checkable chain: which rule was applied, what premises were required, where exceptions were considered, and why the conclusion follows. That quality addresses one of the main barriers to legal adoption—the black-box character of LLM outputs.

At the same time, my current evidence is preliminary and comes with clear constraints. The prover model I tested was not trained on legal doctrine, so some failures likely reflect missing domain knowledge rather than weakness in verification-style reasoning; a full system would need retrieval and a formal rule library to close those gaps. My choice of benchmark tasks and prompts also emphasized exact-format categorical answers, which does not measure the quality of explanations or performance on open-ended legal reasoning. And the lack of a more modern LLM (akin to GPT-5 or DeepSeek-R1) alongside GPT-4 is due to primarily personal research funding limitations; however, this deficiency will be corrected in future testings of the LP paradigm. Finally, my experiment has certainly not yet demonstrated end-to-end formal legal verification: there is no mature proof-assistant library for law comparable to `mathlib`, and selecting an appropriate legal formalism (proof assistant, logic programming, or a restricted logic) remains a key engineering and research choice. While the LP aspirations outlined in Section 3 are not yet tangible, there already exists a clear research path for its potential realization.

6.1 Future Directions

Several concrete research directions follow naturally from this work. The most immediate is building a domain-specific formal knowledge base (an executable library of statutes, definitions, and exceptions) for an area of law that is already

highly codified and comparatively determinate, such as select portions of immigration or tax law. A focused repository of formal rules (in Lean, Prolog, or a custom legal DSL), developed in collaboration with legal experts, would enable true end-to-end evaluation: the system could retrieve authorities, translate them into formal premises, and produce verifiable proofs for questions that fall squarely within that domain.

In parallel, the prover component itself should be adapted to law. Our experiments rely on a math-oriented prover largely for availability; a dedicated legal prover LLM would likely need fine-tuning on legal reasoning patterns and formal representations of legal rules. A key obstacle is the scarcity of "gold" formal proofs in law. This suggests a hands-on strategy: generate proof-like supervision from existing legal QA datasets by eliciting structured, element-by-element justifications (from experts or carefully constrained model pipelines) and treating those derivations as training targets for the prover and verifier loop.

Finally, the framework needs evaluation and governance that matches the domain. Testing on realistic legal problems—bar-style questions, multi-issue hypotheticals, or carefully framed advice scenarios—would reveal whether the system scales beyond single-step tasks and where it breaks (for example, whether it handles rule application well but struggles with open-textured interpretation). Alongside this, ethical and practical questions should be treated as first-class research problems: how to present proofs in lawyer-friendly form, how to keep formal rule libraries synchronized with changing law, how the system should communicate uncertainty or "not proven" outcomes, and what validation standards are required before deploying a mock LP in real-world legal environments.

7 Conclusion

I have presented a novel approach to legal AI that integrates large language models with formal proof verification, drawing inspiration from recent advances in theorem-proving LLMs. My approach aims to bridge the gap between natural language legal corpora and formal logical reasoning systems by using a two-tiered LLM framework: a general legal reasoner and a formal prover, supported by a verifier and retriever. This architecture is designed to ensure that legal conclusions generated by AI are not only correct, but accompanied by a verifiable chain of reasoning, addressing one of the most critical requirements for AI in the legal domain: trust and transparency.

Through rewriting and polishing the provided content, and integrating insights from a range of contemporary papers, I highlighted how this work builds on and differs from prior efforts. Benchmarks like LegalBench and LawBench have demonstrated both the potential and the shortcomings of current LLMs on legal tasks [5,4]. My approach takes a further step by proposing a full embedding of an LLM within a formal proof loop.

The experimental results, albeit preliminary, give a glimpse of the potential of LP models. DeepSeek-Prover-V2 outperformed a similarly-sized black-box model on many legal reasoning tasks, especially where logical structure and multi-step

deduction were key. It did so despite no adaptation to the legal domain, which bodes well for future improvements. I envision that a fully realized LP LLM could serve as a dependable assistant for lawyers, capable of checking the consistency of arguments, finding applicable rules, and even generating first drafts of arguments that a human can then review. In education, such a system could help train law students by providing automated feedback on whether their arguments logically follow from the law (acting as a sort of Socratic tutor that always checks each step).

Ultimately, I hope to narrow the long-standing divide between computational logic approaches to law and statistical NLP. By harnessing the power of both, I aim to create legal AI systems that are not only intelligent in an abstract sense, but also aligned with the rigorous reasoning standards of legal practice. This cross-pollination between CoT and non-CoT can lead to a new generation of legal AI that is both cutting-edge and practically useful. The path forward will require collaboration between computer scientists, legal scholars, and domain experts, but the reward will be systems that truly reason like a lawyer, backed by logic strong enough to withstand scrutiny and inspire professional confidence.

References

1. My experimental model source location. `https://huggingface.co/deepseek-ai/DeepSeek-Prover-V2-671B`
2. Ashley, K.D.: Modelling Legal Argument: Reasoning with Cases and Hypotheticals. Ph.D. thesis, University of Massachusetts (1988)
3. Chlapanis, O., Galanis, D., Androutsopoulos, I.: Lar-echr: A new legal argument reasoning task and dataset for cases of the european court of human rights. In: Proceedings of the Natural Legal Language Processing Workshop 2024 (2024)
4. Fei, Z., Shen, X., Zhu, D., Zhou, F.: Lawbench: Benchmarking legal knowledge of large language models. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (2024)
5. Guha, N., Nyarko, J., Ho, D.E., Ré, C.: Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. arXiv preprint arXiv:2308.11462 (2023), `http://arxiv.org/abs/2308.11462`
6. Hu, Y., Yu, Y., Gan, L., Wei, B.: Evaluating test-time scaling llms for legal reasoning: Openai o1, deepseek-r1, and beyond. arXiv preprint arXiv:2503.16040 (2025), `https://arxiv.org/abs/2503.16040`
7. Joshi, A., Paul, S., Sharma, A., Goyal, P., Ghosh, S., Modi, A.: Il-tur: Benchmark for indian legal text understanding and reasoning. arXiv preprint arXiv:2407.05399 (2024), `http://arxiv.org/abs/2407.05399`
8. Kant, M., Nabi, S., Kant, M., Scharrer, R., Megan, M., Nabi, M.: Towards robust legal reasoning: Harnessing logical llms in law. arXiv preview arXiv:2502.17638 (2025), `http://arxiv.org/abs/2502.17638`
9. Ren, Z., Shao, Z., Song, J., Xin, H.: Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. arXiv preprint arXiv:2504.21801 (2025), `http://arxiv.org/abs/2504.21801`
10. Sergot, M., Sadri, F., Kowalski, R., Kriwaczek, F., Hammond, P., Cory, H.: The british nationality act as a logic program. Communications of the ACM (1986)

11. Varambally, S., Voice, T., Sun, Y., Chen, Z.: Hilbert: Recursively building formal proofs with informal reasoning. arXiv preprint arXiv:2509.22819 (2025), http://arxiv.org/abs/2509.22819

A Appendix

Table 2: DeepSeek-Prover (DSP) vs Baseline Model on Legal Tasks (0-shot setting). Accuracy (%) of each model is reported.

Task	**Samples**	**DSP (%)**	**GPT-4 (%)**
hearsay	94	92.6	83.8
personal_jurisdiction	50	86.0	91.4
proa (professional responsibility)	95	93.7	99.0
consumer_contracts_qa	100	99.0	93.6
abercrombie (trademark distinctiveness)	95	37.9	85.3
corporate_lobbying	100	84.0	81.7
canada_tax_court_outcomes	100	98.0	98.9
privacy_policy_entailment	100	73.0	85.5
insurance_policy_interpretation	100	47.0	69.6
citation_prediction_classification	100	63.0	71.3
contract_qa (NDA QA)	80	98.8	96.2
contract_nli_confidentiality	82	50.0	96.3
contract_nli_explicit_id	100	26.0	82.4
contract_nli_inclusion_verbally	100	65.0	90.7
contract_nli_limited_use	100	86.0	86.6
contract_nli_no_licensing	100	50.0	92.5
contract_nli_notice_disclosure	100	74.0	97.2
contract_nli_permissible_acq	100	61.0	96.1
contract_nli_permissible_copy	87	26.4	80.4
contract_nli_permissible_dev	100	60.0	98.5
contract_nli_post_possession	100	33.0	94.6
contract_nli_return_of_info	66	59.1	95.6
contract_nli_sharing_employees	100	90.0	94.6
contract_nli_sharing_third_parties	100	71.0	93.3
contract_nli_survival	100	70.0	94.0
cuad_affiliate_license_licensee	100	92.0	90.9
cuad_affiliate_license_licensor	88	53.4	92.0
cuad_anti_assignment	100	70.0	91.4
cuad_audit_rights	100	96.0	97.9
cuad_cap_on_liability	100	71.0	95.6

Table 3: DSP Results with 3-shot prompting, compared to GPT-4. GPT-4 results are from [5] for the same tasks.

Task	Samples	DSP (3-shot)	GPT-4
hearsay	94	**97.8%**	83.8%
personal_jurisdiction	50	**99.3%**	91.4%
proa (professional responsibility)	95	93.3%	**99.0%**
consumer_contracts_qa	100	**98.3%**	93.6%
abercrombie	95	53.7%	**85.3%**
corporate_lobbying	100	**86.0%**	81.7%
canada_tax_court_outcomes	100	97.3%	**98.9%**
privacy_policy_entailment	100	82.0%	**85.5%**
insurance_policy_interpretation	100	49.3%	**69.6%**
citation_prediction_classification	100	50.3%	**71.3%**
contract_qa	80	98.8%	96.2%
contract_nli_confidentiality_of_agreement	82	75.2%	**96.3%**
contract_nli_explicit_identification	100	68.0%	**82.4%**
contract_nli_inclusion_of_verbally_conveyed_information	100	63.3%	**90.7%**
contract_nli_limited_use	100	**98.7%**	86.6%
contract_nli_no_licensing	100	82.7%	**92.5%**
contract_nli_notice_on_compelled_disclosure	100	93.3%	**97.2%**
contract_nli_permissible_acquirement_of_similar_information	100	94.0%	**96.1%**
contract_nli_permissible_copy	87	56.3%	**80.4%**
contract_nli_permissible_development_of_similar_information	100	86.7%	**98.5%**
contract_nli_permissible_post-agreement_possession	100	70.3%	**94.6%**
contract_nli_return_of_confidential_information	66	77.3%	**95.6%**
contract_nli_sharing_with_employees	100	92.0%	**94.6%**
contract_nli_sharing_with_third-parties	100	79.3%	**93.3%**
contract_nli_survival_of_obligations	100	83.3%	**94.0%**
cuad_affiliate_license-licensee	100	**95.7%**	90.9%
cuad_affiliate_license-licensor	88	75.8%	**92.0%**
cuad_anti-assignment	100	**97.7%**	91.4%
cuad_audit_rights	100	93.7%	**97.9%**
cuad_cap_on_liability	100	**98.0%**	95.6%

Providing Open Access Legal Risk Advice: Reflections on Building an Expert System for Micro-Entities

Stuart Weinstein[0000-0003-3225-9642]

Aston Law School, Aston University, Aston Triangle, Birmingham B4 7ET UK
PhD Student, University of Oslo, Faculty of Law, Department of Private Law, Norwegian Research Center for Computers and Law, Karl Johans gate 47, 0162 Oslo, Norway
s.weinstein@aston.ac.uk

Abstract. This paper examines the development of the Online Legal Risk Advisory System (OLRAS), a rule-based expert system designed to support legal risk management for micro-entities (small businesses meeting at least two of the following criteria: turnover of £1,000,000 or less, a balance sheet total of £500,000 or less, and ten or fewer employees). OLRAS provides free, accessible guidance to help users identify and reflect on contractual risks, addressing both economic and technical barriers to legal support. The paper situates OLRAS within the broader context of open-access legal technology and examines the challenges of encoding legal expertise in forms usable by non-specialists. It highlights the scalability of such systems and reflects on key design constraints, including usability, regulatory boundaries, and the representation of legal knowledge. The paper concludes by outlining the limitations of the approach and identifying directions for future research.

Keywords: Expert Systems, Juris-Informatics, Micro-Entities, OLRAS, Open Access to Law, Legal Technology.

1 Introduction

1.1 Overview

This paper explores the development of the Online Legal Risk Advisory System (OLRAS), an expert system designed to support legal risk management. OLRAS (**https://www.visirule.co.uk/visirule-demos/olras**) helps micro-entities identify and assess legal risks so they can make informed decisions about their contractual rights and responsibilities. For the purposes of this paper, legal risk in contracts refers to the possibility that contractual terms or omissions may lead to adverse legal or commercial outcomes, such as unenforceability, financial loss, or exposure to liability. In practice, such risks often arise from unclear, imbalanced, or incomplete contractual provisions. OLRAS serves as a complementary legal application that assists micro-entities in evaluating and managing potential legal risks when entering into commercial contracts. A micro-entity is defined as a small business meeting at least two of the following criteria: turnover of £1,000,000 or less, a balance sheet total of £500,000 or less, and ten or

fewer employees [1]. A detailed account of the knowledge engineering process, including rule extraction and validation, is provided in Weinstein [2] and is beyond the scope of this reflective paper.

1.2 Legal Challenges of Micro-Entities

A comprehensive survey involving 10,228 small businesses in England and Wales, completed in July 2021, indicated that trading issues such as those arising from the purchase or sale of goods or services between individuals and businesses or with other businesses constitute more than half of the legal challenges encountered [3]. Of the small businesses in England and Wales operating, micro-entities are particularly disadvantaged across various metrics, being the least likely to seek professional assistance, possessing the lowest legal capabilities, and experiencing less favourable outcomes [3]. As of 2025, approximately 5.7 million small businesses (99% of all businesses) operate in the UK, with micro-entities accounting for over 5.4 million, thereby constituting 95% of businesses [4].

1.3 Structure of Paper

The paper proceeds as follows: Section 2 introduces OLRAS as a decision-support system; Section 3 explains its operational logic; Section 4 presents an illustrative use case; Section 5 discusses design, legal, and economic implications; Section 6 concludes.

2 What is OLRAS?

2.1 OLRAS as a Decision Support Tool

OLRAS was developed as a decision-support tool intended to prompt micro-entities to consider common sources of legal risk in commercial contracting. It does not provide bespoke legal advice, nor does it automate legal decision-making. Instead, it structures such considerations into a guided question-and-answer process, encouraging users to reflect on contractual issues that are frequently overlooked by businesses operating without legal counsel.

OLRAS is based on English contract law and has been developed as a prototype system, with limited testing in controlled or demonstrative settings.

The system has been deployed in a demonstrative online environment and has not yet undergone large-scale real-world evaluation.

2.2 Why Use a Rule-Based Expert System?

The decision to adopt a rule-based expert system approach was deliberate. Rule-based systems offer transparency and explainability, attributes that are particularly important when users are non-lawyers and when legal accountability remains with the user

rather than the system [5, 6]. In contrast to data-driven or probabilistic systems, rule-based approaches allow legal reasoning to be articulated in relatively plain language, helping users understand why particular risks are highlighted.

2.3 Open-Access Deployment

A defining characteristic of OLRAS is its open-access deployment. By making legal risk guidance freely available online, the project sought to address the economic exclusion of micro-entities from conventional legal services. Once developed, the cost of providing guidance to additional users is *de minimis*, illustrating how open access legal technologies can scale access to legal knowledge in ways that traditional professional services cannot.

3 How OLRAS Works?

3.1 Step One – User Interaction

OLRAS operates as an interactive, question-driven system that guides users through a structured assessment of potential legal risks in commercial contracting.

The process begins when a user accesses the system and is presented with a series of plain-language questions relating to common contractual issues. These questions are designed to prompt reflection on key aspects of an agreement, such as payment terms, liability provisions, termination rights, and dispute resolution mechanisms. Users respond based on their understanding of the contract or the terms being negotiated.

As the user progresses, the system processes each response using a predefined set of legal rules. These rules encode typical risk scenarios derived from legal expertise and are applied conditionally based on the user's answers. Rather than performing probabilistic analysis, OLRAS follows a rule-based logic, where specific combinations of responses trigger corresponding risk assessments.

3.2 Step Two – The System Generates Output Highlighting Areas of Concern

Based on this evaluation, the system generates outputs that highlight potential areas of concern. These outputs are presented as qualitative risk indicators accompanied by brief explanations, helping users understand why a particular issue may be significant. In some cases, the system may also suggest areas where further clarification or professional advice should be sought.

Importantly, OLRAS does not provide definitive legal advice or automated decision-making. Instead, it supports users in identifying and reflecting on potential risks, enabling more informed decision-making when entering into or reviewing contractual arrangements.

4 Illustrative Use Case of OLRAS

The following example illustrates a typical user interaction with OLRAS.

4.1 Scenario

To demonstrate how OLRAS operates in practice, consider a micro-entity entering into a commercial supply agreement with a larger counterparty.

A small business is offered a contract to supply goods to a larger company. The contract includes standard terms drafted by the counterparty, and the micro-entity seeks to understand potential legal risks before agreeing.

4.2 User Interaction

Upon accessing OLRAS, the user is guided through a sequence of structured questions. These include, for example:

- Whether payment terms are clearly defined and time-bound
- Whether liability is limited or excluded in favour of the counterparty
- Whether termination rights are mutual or one-sided
- Whether there are penalties or indemnities imposed on the supplier

The user responds based on their reading of the contract. Where uncertainty arises, the system encourages the user to review the relevant clauses before proceeding.

4.3 System Processing

As responses are entered, OLRAS evaluates them against its underlying rule set. For instance, if the user indicates that liability is broadly excluded for the counterparty but not for themselves, the system identifies this as a potential imbalance. Similarly, the absence of clear payment timelines may trigger a risk relating to cash flow and enforceability.

These evaluations are not probabilistic but are based on predefined conditional logic linking specific contractual features to recognised risk patterns.

4.4 Output

At the conclusion of the interaction, the system presents a set of risk indicators. These are expressed qualitatively (e.g., “potential concern”, “heightened risk”) and are accompanied by brief explanations. For example, the system may highlight that one-sided termination rights could expose the user to unexpected contract termination without remedy, or that unclear payment terms may increase the likelihood of delayed or disputed payments.

The output is designed to prompt reflection rather than prescribe action. Users are encouraged to consider whether the identified issues require renegotiation, clarification, or professional legal advice.

4.5 Interpretation

Through this process, OLRAS supports the user in recognising contractual risks that might otherwise be overlooked. Rather than replacing legal judgment, the system structures the user's attention and provides a framework for more informed decision-making in situations where professional legal support may not be readily accessible.

5 Discussion

5.1 Legal and Design Constraints on Open Access

The OLRAS project illustrates the constraints faced by open-access legal tools. Here, "open access" is understood in a broader sense, encompassing not only access to legal information, but also the provision of accessible, user-oriented legal guidance for non-lawyers. One significant challenge arises from copyright restrictions associated with legal standards and model texts. Although standards such as ISO 31022:2020 provide valuable guidance on legal risk management, their proprietary status limits direct reuse in open systems. As a result, developers must reinterpret and operationalise underlying principles rather than reproducing authoritative text [7, 8].

Usability presents an equally important constraint. Research on legal readability and legal capability emphasises that legal tools are only effective if they are accessible to their intended users [9, 10]. In developing OLRAS, the ambition to create a comprehensive legal risk framework had to be balanced against the risk of overwhelming users with limited legal literacy. This required simplifying legal concepts while retaining sufficient meaning to support decision-making.

In retrospect, greater use of co-design methodologies involving micro-entity users throughout development may have improved clarity and relevance [11]. The experience highlights the importance of participatory design in open-access legal technologies, particularly where users lack formal legal training. Systems such as OLRAS must therefore balance doctrinal completeness with usability, ensuring that guidance remains both intelligible and practically useful.

5.2 Economic Implications of Open Access Legal Technology

From an economic perspective, OLRAS illustrates how open access legal technologies can address gaps in legal service provision. Bespoke legal advice is often unaffordable for micro-entities, even where contractual risks may have significant commercial consequences. Open-access systems offer an alternative model in which guidance can be delivered at scale without a corresponding increase in cost.

At the same time, regulatory constraints (particularly those relating to the unauthorised practice of law) shape the permissible scope of such systems. OLRAS was therefore designed as a decision-support tool rather than a provider of legal advice. This positioning reflects broader tensions in legal technology between expanding access to legal support and maintaining professional regulatory boundaries [12].

5.3 Reflection and Future Directions

The OLRAS project has attracted criticism for relying on a rule-based expert system at a time when machine learning and large language models dominate legal technology discourse. While data-driven approaches offer powerful capabilities, they also raise concerns regarding explainability, bias, and reliability, particularly in legal contexts [13, 14]. The development of OLRAS suggests that rule-based systems continue to have value where transparency and user understanding are paramount.

At the same time, the project highlights the limits of expert systems in addressing the full complexity of legal risk. Legal risk is inherently contextual and often resists precise quantification, particularly for small organisations lacking structured data. Its assessment depends significantly on professional judgment and organisational context [15]. OLRAS therefore offers qualitative guidance rather than definitive assessments, supporting user reflection rather than automated decision-making.

More broadly, the project highlights the importance of framing. Presenting OLRAS as a supportive tool, rather than a substitute for legal advice, was essential in managing both user expectations and regulatory risk.

If generative AI were to be incorporated into OLRAS in future iterations, it could enhance user interaction by providing natural language explanations or acting as an interface layer. However, this would introduce challenges relating to reliability, explainability, and legal accountability, which are particularly significant in the context of open-access legal guidance.

6 Conclusion and Future Research

The development of OLRAS provides a useful lens through which to examine both the promise and the limits of open-access legal risk advisory systems. The project demonstrates that rule-based expert systems can play a meaningful role in extending legal capability to micro-entities that are otherwise excluded from professional legal services. At the same time, it underscores that such systems are not substitutes for legal representation, nor are they capable of resolving the inherent indeterminacy and contextual nature of legal risk.

It is therefore important to be clear about the scope of the contribution offered here. This article does not claim that OLRAS is empirically validated as an effective legal intervention, nor that its design choices are optimal or generalisable across domains. Rather, the value of the project lies in what it reveals about the practical constraints of translating legal expertise into open-access, user-facing systems: the tension between completeness and usability, the influence of copyright and regulatory frameworks, and the importance of framing and expectation management when offering legal guidance without professional intermediation.

From a juris-informatics perspective, these reflections reinforce a recurring insight within the field: that legal information systems are socio-technical artefacts whose success depends as much on institutional context, user capability, and normative legitimacy as on formal correctness or computational sophistication. As generative AI and

data-driven approaches continue to reshape legal technology discourse, there remains a place for transparent, rule-based systems that prioritise explainability and user trust, particularly in access-to-justice contexts.

Future research should therefore focus less on replacing such systems with more complex automation, and more on understanding how different forms of legal technology can be combined responsibly. This includes systematic user studies, participatory co-design with underserved communities, and careful exploration of hybrid approaches that preserve accountability while enhancing usability. The experience of OLRAS suggests that sustainable open-access legal advice will ultimately be judged not by technical novelty, but by its ability to support informed decision-making for those who would otherwise navigate legal risk alone.

Acknowledgments. The author acknowledges with gratitude feedback received from Professor Tobias Mahler of the University of Oslo, Clive Spenser and Alan Westwood of Logic Programming Associates, Ltd. (https://www.lpai.uk/) and the reviewers of the JURISIN 2026 Organizing Committee. Research funding to develop OLRAS was provided by Aston University. The views expressed herein are solely those of the author.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this paper.

References

1. Companies House: Accounts guidance for micro-entities. UK Government. https://www.gov.uk/annual-accounts/microentities-small-and-dormant-companies, last accessed 2026/1/11
2. Weinstein, S.: OLRAS: Closing the E-justice gap faced by “lawyer-less” micro-entities when they are negotiating commercial contracts against larger “lawyered-up” parties. In: Jacob, K., Schindler, D., Strathausen, R., Waltl, B. (eds.) Liquid Legal – Sustaining the Rule of Law: Artificial Intelligence, E-Justice, and the Cloud, pp. 63–93. Springer Nature, Switzerland, Cham (2025).
3. Legal Services Board: Small business legal needs: Wave four survey. https://legalservicesboard.org.uk/wp-content/uploads/2022/05/20220406-Small-business-legal-needs-FINAL.pdf, last accessed 2026/1/11
4. Murray, A.: Business statistics. House of commons library research briefing, 3 December 2025. London: House of Commons Library. https://researchbriefings.files.parliament.uk/documents/SN06152/SN06152.pdf, last accessed 2026/1/11
5. Waterman, D.A., Paul, J., Peterson, M.: Expert systems for legal decision making. Expert Systems 3(4), 212–226 (1986).
6. Leith, P.: The rise and fall of the legal expert system. International Review of Law, Computers & Technology 30(3), 94-106 (2016).
7. International Organization for Standardization (ISO): ISO 31022:2020: risk management — guidelines for the management of legal risk. Geneva: ISO. https://www.iso.org/standard/69295.html, last accessed 2026/1/11
8. Ola, K.: Theories of open access. Journal of Open Access to Law **6**(1) (2018). doi: 10.63567/mkqsmn32

9. Curtotti, M., Weibel, W., McCreath, E., Ceynowa, N., Frug, S., Bruce, T.: Citizen science for citizen access to law. Journal of Open Access to Law 3, 57 (2015).
10. Hagan, M.: The justice is in the details: Evaluating different self-help designs for legal capability in traffic court. Journal of Open Access to Law 7, 1 (2019).
11. Pope, H.J., Treni, A.: Sharing knowledge, shifting power: Rebellious legal design. Journal of Open Access to Law **9**(1) (2021) doi: 10.63567/qpdygf18.
12. Walters, E.: Re-regulating UPL in an age of AI. Georgetown Law Technology Review 8, 316 (2024).
13. Porębski, A.: Machine learning and law. In: Brożek, B., Kanevskaia, O., Pałka, P. (eds.) Research Handbook on Law and Technology, pp. 450–467. Edward Elgar, Cheltenham, UK (2023).
14. Riehl, D.A.: Evolution of the data-driven lawyer. University of St Thomas Law Journal 20, 368 (2024).
15. Ostrander, R.: The legal function's role in the risk management framework. Bank for International Settlements Legal Experts' Meeting. https://www.newyorkfed.org/newsevents/speeches/2024/ost240419, last accessed 2026/1/11

A Method for Detecting Incorrect Correspondences in Automatically Predicted Legislative Article Mappings

Taiyo Maehara[1][0009−0004−3021−3688], Tomoya Sano[2][0000−0002−3229−3149], and Yoichi Takenaka[3][0000−0003−3286−2265]

[1] Kansai University, Suita Osaka, Japan
suisougakuperc@gmail.com
[2] Center for Digital Humanities and Social Sciences, Nagoya University, Nagoya Aichi, Japan
tomoya@law.nagoya-u.ac.jp
[3] Faculty of Informatics, Kansai University, Takatsuki Osaka, Japan
takenaka@kansai-u.ac.jp

Abstract. In legal research and legislative practice, tracing the legislative history of provisions is indispensable; however, tracking and verifying correspondences between old and new provisions for each amendment at the level of individual articles or sentences incurs a substantial human cost. Moreover, even when the textual similarity between provisions is high, they may not correspond institutionally. It is difficult to determine correspondences based on a single criterion automatically. Therefore, this study proposes an alerting method that prioritizes and presents points that are likely to be erroneous in the outputs of existing old–new correspondence estimation methods. The proposed method performs correspondence estimation at the sentence level in the Law Standard XML, and, focusing on the local monotonicity of article numbers or sentence order in legislative amendments, represents local inconsistency as features using the mean difference of predicted destination-number gaps between adjacent sentences, and detects error candidates via linear discriminant analysis. Trained on amendment data of the Commercial Code and evaluated on amendment data of the Family Law from Civil Code, the method identified 41 cases corresponding to errors of the existing method among 69 alerts, achieving a precision of 0.594, a recall of 0.263, and an F1 value of 0.365. These results indicate that the proposed method is effective not for exhaustively detecting errors, but as support that prioritizes candidates to be checked under limited human verification cost.

Keywords: Natural Language Processing · Legal Informatics · Japanese Law · Sentence-BERT

1 Introduction

In legal research and legal practice, legislative histories of statutory provisions constitute indispensable materials. The interpretation of statutes must be con-

ducted in light not only of the current text but also of the purposes and background of their enactment and amendments. Accordingly, for legal scholars, understanding how the statutes they study reached their current provisions through amendments is a starting point of research. In legislative and administrative drafting, legislative histories are routinely checked when statutes are revised [1]. However, tracking and verifying legislative histories and correspondences at the provision level tends to rely on manual work and entails substantial human cost. Therefore, reducing the burden of compiling and checking legislative histories has become a practical challenge in both research and practice.

A legislative history of provisions refers to information that organizes, at the level of individual provisions, the correspondence relationships between old and new provisions that arise with each amendment. A comparable material is the old-new correspondence table; however, such tables merely present the "before and after" of each amendment, and to trace the continuity and evolution of provisions across multiple amendments at the level of individual provisions, it is necessary to integrate fragmented information across amendments.

In recent years, although access to statutory data has become easier, the environment for comprehensively grasping provision-level legislative histories is still far from sufficient. For example, e-Gov Law Search (e-Gov 法令検索) [4] manages historical information. It does not, however, provide the histories that enable tracking correspondences between provisions for the period before the service launch in 2017 [2]. In addition, the Japanese Law Index (日本法令索引) [5] provides information at the statute level, and obtaining provision-level correspondences requires consulting multiple materials. Furthermore, there have been attempts to digitize provision-level legislative histories of past statutes, but the target statutes are limited in scope. Overall, existing public tools and individual expert efforts alone remain insufficient to streamline the grasping and verification of provision-level legislative histories.

Moreover, correspondences in a legislative history are not uniquely determined, because what counts as an appropriate linkage depends on the compiler's standpoint and purpose. When a legal institution is comprehensively replaced, the change may be treated, as a matter of legislative form, as a repeal and enactment, and amendment materials may not explicitly indicate relationships between the old and new provisions. For example, provisions on inheritance based on the Japanese family system in the Civil Code, which were deleted by the 1947 Civil Code amendment, are close in wording to general inheritance provisions [5]. However, because the underlying institutions of the two sets of provisions are fundamentally different, they are not treated as corresponding provisions in amendment materials prepared from a strict administrative standpoint. Nevertheless, from the perspective of legal practitioners, such correspondences may still be useful as references. Thus, the appropriate granularity and scope differ by standpoint: administrative practice emphasizes strict determination of correspondences, whereas legal practice emphasizes referability. Therefore, auto-

[4] https://laws.e-gov.go.jp/
[5] https://hourei.ndl.go.jp/

matically determining correspondences under a single criterion is difficult, and support that accommodates purpose-dependent criteria is needed.

Maehara et al. proposed methods to estimate old-new comparison tables from provision similarity, provision length, and related features [5, 4]. However, to use the estimated results in practice, humans must inspect numerous correspondence candidates and remove errors, leaving the problem of verification cost. In addition, because similarity in meaning does not necessarily imply correspondence, methods that set a similarity threshold and uniformly align provisions cannot be applied. As a result, in settings where high reliability is required, extensive manual verification is unavoidable. Therefore, rather than automatically fixing correspondences, support that presumes human verification and presents verification priorities is required.

In this study, we propose a method that, given correspondence estimates produced by existing methods, detects correspondences that are likely to be erroneous and presents priorities for human verification. Specifically, focusing on the structural continuity of article numbering in legislative amendments, we issue warnings when estimated correspondences are inconsistent with this structure. This provides clues for error correction for standpoints that demand strict determination of correspondences, and, for standpoints that emphasize referability, it encourages consideration of provision relationships that are less likely to appear in old-new comparison tables. The novelty contributions of this study are summarized in the following three points.

1. We formulated the problem with the primary goal of supporting verification by presenting error candidates for existing estimation results, rather than automatically determining correspondences.
2. We designed features of local inconsistency based on local continuity of article numbering and used them to detect erroneous correspondences.
3. Through cross-domain evaluation from the Commercial Code to the Civil Code (family law), we confirmed the effectiveness of the proposed method.

2 Existing Method

In this study, as a basic framework for automatic alignment between new and old provisions, we adopt the correspondence estimation method based on provision similarity proposed in prior work [5, 4]. In this section, we summarize the existing framework that underlies the proposed method.

Notation.

Let $\mathcal{I}$ be a set of provision identifiers and $\mathcal{T}$ be a set of provision texts. In this paper, we represent a statute as a “mapping from provision identifiers to provision texts” and model a statute as a finite partial map

$$L : \mathcal{I} \rightharpoonup \mathcal{T}$$

We define the domain of L as

$$\mathrm{dom}(L) = \{\, i \in \mathcal{I} \mid L(i) \text{ is defined} \,\}$$

and for $i \in \mathrm{dom}(L)$, we regard $L(i) \in \mathcal{T}$ as the provision text corresponding to the identifier i.

We also define the set of provisions contained in L as

$$\mathrm{Art}(L) = \{(i,\, L(i)) \mid i \in \mathrm{dom}(L)\} \subseteq \mathcal{I} \times \mathcal{T}$$

and call each element $a \in \mathrm{Art}(L)$ a provision. We denote the identifier and text of a provision a by the projections

$$\mathrm{id}(a) \in \mathcal{I}, \qquad \mathrm{text}(a) \in \mathcal{T}$$

Problem Setting.

Given a new statute L^{new} and an old statute L^{old}, we estimate correspondences between their provisions. Let $\mathrm{Art}(L^{\mathrm{new}})$ be the set of provisions in the new statute and $\mathrm{Art}(L^{\mathrm{old}})$ be the set of provisions in the old statute.

The existing method outputs a bipartite graph whose nodes are the new and old provisions:

$$\hat{B} = (\mathrm{Art}(L^{\mathrm{new}}), \mathrm{Art}(L^{\mathrm{old}}), \hat{E})$$

Here,

$$\hat{E} \subseteq \mathrm{Art}(L^{\mathrm{new}}) \times \mathrm{Art}(L^{\mathrm{old}})$$

is the set of estimated correspondences that are represented as edges. For evaluation, we use a correspondence dataset created by legal experts as the ground truth.

Baseline Framework (Two-phase Binary Classification).

For each provision a, we obtain a d-dimensional embedding representation $\mathbf{h}(a) \in R^d$ using an embedding method such as Sentence-BERT. We define the similarity between a new provision $a_n \in \mathrm{Art}(L^{\mathrm{new}})$ and an old provision $a_o \in \mathrm{Art}(L^{\mathrm{old}})$ as $\mathrm{sim}\big(\mathbf{h}(a_n), \mathbf{h}(a_o)\big)$.

For each new provision a_n, let $a_o^{(j)}(a_n) \in \mathrm{Art}(L^{\mathrm{old}})$ denote the j-th old provision when $\mathrm{Art}(L^{\mathrm{old}})$ is sorted in descending order of $\mathrm{sim}\big(\mathbf{h}(a_n), \mathbf{h}(a_o)\big)$. We then define the top-k similarity scores as

$$\mathrm{sim}_{(j)}(a_n) = \mathrm{sim}(a_n, a_o^{(j)}(a_n)) \qquad (j = 1, \ldots, k)$$

which satisfy $\mathrm{sim}_{(1)}(a_n) \geq \mathrm{sim}_{(2)}(a_n) \geq \cdots \geq \mathrm{sim}_{(k)}(a_n)$.

In this study, we use these top similarity scores (e.g., $k = 5$) and the provision length, among others, as candidate features, and specify the set of features actually used as an experimental condition.

Phase 1. For each new provision a_n, a binary classifier f_1 predicts whether a corresponding old provision exists:

$$y_1(a_n) = f_1(\phi_1(a_n)) \in \{0, 1\}.$$

Here, $\phi_1(a_n)$ is a feature vector composed of the above top similarity scores $\big(\mathrm{sim}_{(1)}(a_n), \ldots, \mathrm{sim}_{(k)}(a_n)\big)$, the provision length, $|\mathrm{text}(a_n)|$, and other statistics.

Phase 2. Only when $y_1(a_n) = 1$, we choose the most similar old provision $a_o^{(1)}(a_n)$ as a candidate and determine whether it corresponds using a binary classifier f_2:

$$y_2(a_n) = f_2(\phi_2(a_n, a_o^{(1)}(a_n))) \in \{0, 1\}.$$

Here, $\phi_2(a_n, a_o^*(a_n))$ is a feature vector, and it may include as candidate features the similarity of the candidate pair $\mathrm{sim}(a_n, a_o^*(a_n))$, the top similarity scores $\big(\mathrm{sim}_{(1)}(a_n), \ldots, \mathrm{sim}_{(k)}(a_n)\big)$, and the provision length, among others.

The set of estimated correspondences output by the existing method is given by

$$\hat{E} = \{(a_n, a_o^{(1)}(a_n)) \mid a_n \in \mathrm{Art}(L^{\mathrm{new}}),\ y_1(a_n) = 1,\ y_2(a_n) = 1\}.$$

The estimated output for the correspondence is

$$\hat{B} = (\mathrm{Art}(L^{\mathrm{new}}), \mathrm{Art}(L^{\mathrm{old}}), \hat{E}).$$

Baseline Assumption

In the existing method, for each new provision $a_n \in \mathrm{Art}(L^{\mathrm{new}})$, one old provision candidate is selected and the method determines whether the candidate corresponds. Thus, the estimated correspondence from new provisions to old provisions can be represented as a partial map

$$g : \mathrm{Art}(L^{\mathrm{new}}) \rightharpoonup \mathrm{Art}(L^{\mathrm{old}})$$

where each new provision has at most one corresponding old provision. On the other hand, different new provisions are allowed to correspond to the same old provision, and hence g is not injective in general. In this case, the estimated output of the existing method is given as the graph of

$$g : \hat{E} = \{(a_n, g(a_n)) \mid a_n \in \mathrm{dom}(g)\}$$

If we regard $g(a_n) = a_o^{(1)}(a_n)$ (when $y_1(a_n) = y_2(a_n) = 1$), then $\hat{E}$ above coincides with the graph of g.

Evaluation protocol

In previous studies, experiments on the Japanese Commercial Code and the Family law in the Civil Code have shown that this framework can align provisions with high accuracy when using an expert-created correspondence dataset as the ground truth ($E^\star$). In this study, we focus on the predictions in the second phase of this existing framework and propose a new method to judge their correctness.

3 Proposed Method

In this study, we modify and extend the existing method in the following two respects.

1. Instead of aligning at the provision level, we replace it with sentence-level alignment using the tag of the Law Standard XML [3, 7].
2. Based on the estimated correspondences $\hat{E}$ output by the existing method, we estimate, for each new sentence, the "possibility that the decision of the existing method is incorrect" and output it as an alert.

Sentence-level setting

Experiments in the existing method have shown that provision-level old-new alignment cannot handle splits and merges of provisions. Therefore, because it is necessary to perform correspondence at a finer granularity, the proposed method conducts correspondence at the sentence level.

For each statute L, let $\mathrm{Sent}(L)$ be the set of sentences contained in L. Each sentence $s \in \mathrm{Sent}(L)$ is assigned a unique number $\mathrm{sentid}(s) \in \{1, \ldots, |\mathrm{Sent}(L)|\}$ according to its order of appearance in the statute as a whole. We write the text of a sentence s as $\mathrm{text}(s)$.

Given a new statute L^{new} and an old statute L^{old}, the estimated correspondences obtained by applying the existing method at the sentence level are represented as

$$\hat{B}_{\mathrm{sent}} = \big(\mathrm{Sent}(L^{\mathrm{new}}),\ \mathrm{Sent}(L^{\mathrm{old}}),\ \hat{E}_{\mathrm{sent}}\big)$$

where $\hat{E}_{\mathrm{sent}} \subseteq \mathrm{Sent}(L^{\mathrm{new}}) \times \mathrm{Sent}(L^{\mathrm{old}})$ is the set of estimated correspondences that is represented as edges. Hereafter, for simplicity of notation, we write $\hat{E}_{\mathrm{sent}}$ as $\hat{E}$ unless otherwise stated.

Alerting task

In this study, for the outputs of the existing method, we produce the following two types of alerts at the level of sentences on the new-statute side.

– **False-positive alert (FP):** among new sentences that the existing method included in $\hat{E}$ as "having a correspondence", those for which the correspondence may be incorrect. We denote the set by

$$\mathcal{A}_{\mathrm{FP}} \subseteq \mathrm{Sent}(L^{\mathrm{new}}).$$

– **False-negative alert (FN):** among new sentences that the existing method judged as "having no correspondence", those for which a correspondence may exist. We denote the set by

$$\mathcal{A}_{\mathrm{FN}} \subseteq \mathrm{Sent}(L^{\mathrm{new}}).$$

Similarity ranking (reused from the baseline)

As in the existing method, for each sentence s we obtain a d-dimensional embedding representation $\mathbf{h}(s) \in R^d$ using an embedding method such as Sentence-BERT. We define the similarity between a new sentence $s_n \in \mathrm{Sent}(L^{\mathrm{new}})$ and an old sentence $s_o \in \mathrm{Sent}(L^{\mathrm{old}})$ as $\mathrm{sim}\big(\mathbf{h}(s_n), \mathbf{h}(s_o)\big)$. For each new sentence s_n, let $s_o^{(j)}(s_n) \in \mathrm{Sent}(L^{\mathrm{old}})$ denote the j-th old sentence when $\mathrm{Sent}(L^{\mathrm{old}})$ is sorted in descending order of $\mathrm{sim}\big(\mathbf{h}(s_n), \mathbf{h}(s_o)\big)$. We define the top-$k$ similarity scores as

$$\mathrm{sim}_{(j)}(s_n) = \mathrm{sim}\big(s_n, s_o^{(j)}(s_n)\big) \qquad (j = 1, \ldots, k).$$

In this study, we use the top similarity scores ($k = 5$) and the sentence length, among others, as candidate features, and specify the set of features actually used as an experimental condition.

Neighbor operators

We focus on the structural characteristic of continuity in article numbering. In Japanese legislative amendments, it is often observed that the numbering of old and new provisions corresponds in a largely continuous rule. For example, in the Civil Code amendment from Act No.61 of 1947 to Act No.222 of 1947, the successor of Article 827 of the old statute becomes Article 779 of the new statute, and the successor of Article 828 of the old statute becomes Article 780 of the new statute, indicating that the correspondences evolve almost monotonically (Fig.1). A similar tendency has also been observed in amendments to other statutes, such as the Commercial Code. Based on this empirical observation, we assume that when the predicted correspondences do not evolve monotonically across adjacent sentences of the prediction target, the decision of the existing method may be erroneous.

In what follows, we define quantities for using the predicted outcomes of adjacent sentences as explanatory variables for producing alerts.

For each statute L, we define the adjacent sentences of a sentence $s \in \mathrm{Sent}(L)$ as

$$\mathrm{prev}(s) = \begin{cases} s' & \text{if } s' \in \mathrm{Sent}(L) \text{ and } \mathrm{sentid}(s') = \mathrm{sentid}(s) - 1, \\ \bot & \text{otherwise,} \end{cases}$$

$$\mathrm{next}(s) = \begin{cases} s' & \text{if } s' \in \mathrm{Sent}(L) \text{ and } \mathrm{sentid}(s') = \mathrm{sentid}(s) + 1, \\ \bot & \text{otherwise} \end{cases}$$

where $\bot$ denotes endpoints.

Proposed features based on local consistency

Based on the most similar candidate in the existing method, we denote the destination candidate (old sentence) for a new sentence s_n by $s_o^{(1)}(s_n)$, and define the index of the destination candidate in the old statute as

$$\hat{m}(s_n) = \mathrm{sentid}\big(s_o^{(1)}(s_n)\big).$$

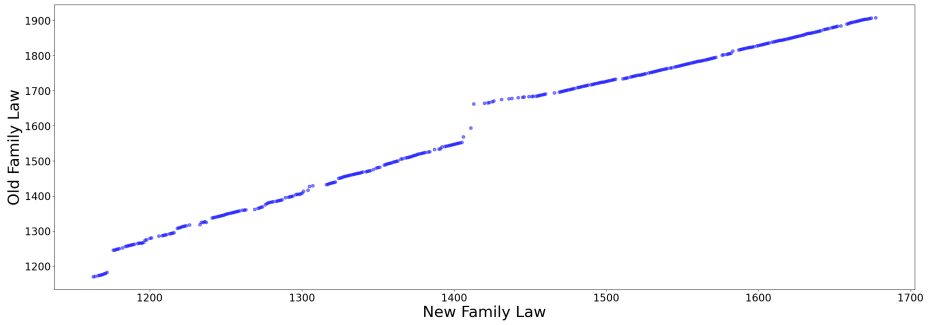


Fig. 1. Correspondence table of new and old sentences in the 1947 amendment to the Civil Code (family law)

To capture continuity of destination candidates across consecutive new sentences, we define local-consistency features as follows:

$$\mathrm{front}(s_n) = \begin{cases} \hat{m}(s_n) - \hat{m}(\mathrm{prev}(s_n)) & \text{if } \mathrm{prev}(s_n) \neq \perp, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathrm{back}(s_n) = \begin{cases} \hat{m}(\mathrm{next}(s_n)) - \hat{m}(s_n) & \text{if } \mathrm{next}(s_n) \neq \perp, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mathrm{whole}(s_n) = \frac{1}{2}\big(\mathrm{front}(s_n) + \mathrm{back}(s_n)\big).$$

These are intended to quantify local discontinuities in the sequence of old-side destination indices for the new-sentence sequence.

Alert classifiers

In this study, we learn a binary classifier that estimates, for each new sentence s_n, the possibility that the decision of the existing method is incorrect, and outputs alerts. Let $\phi_{\mathrm{base}}(s_n)$ be the feature vector consisting of the features used by the existing method for the Phase 2 decision. Let the local-consistency features introduced in this study be

$$\psi(s_n) = \big(\mathrm{front}(s_n),\, \mathrm{back}(s_n),\, \mathrm{whole}(s_n)\big).$$

We construct the concatenated feature vector as

$$\mathbf{x}(s_n) = \big[\phi_{\mathrm{base}}(s_n);\ \psi(s_n)\big].$$

4 Experimental conditions

We use Sentence-BERT [6] to generate distributed representations. As a pre-trained model, we use `sonoisa/sentence-bert-base-ja-mean-tokens-v2`[6].

The datasets used in this experiment are shown in Table 1. The 1938 amendment to the Commercial Code was a large-scale revision accompanied by the renumbering of articles, which required substantial alignment work. In contrast, the 1947 amendment to the Civil Code was a period in which the concept of the family system was substantially changed, resulting in prominent substantive changes.

Table 1. 本研究の対象

Training	Old	Commercial Code, Act No. 79 of 1937	634 articles
	New	Commercial Code, Act No. 72 of 1938	851 articles
Test	Old	Civil Code, Act No. 61 of 1947	422 articles
	New	Civil Code, Act No. 222 of 1947	320 articles

In the initial prediction phase, we use Linear Discriminant Analysis (LDA), the similarity to the provision with the second-highest similarity, and the difference in character length, which showed good performance in prior work.

In the proposed method, we also use LDA as the model, and add whole(s_n), which achieved a good score in preliminary experiments, to the feature set used in the prediction phase. The ground-truth labels used for training are defined as follows: after predicting correspondences in the Commercial Code using the existing method, cases for which an alert should be issued, i.e., those from which False Positives and False Negatives can be derived, are labeled True, whereas correct cases for which no alert should be issued are labeled False.

5 Result

First, Table 2 shows the confusion matrix of the prediction results produced by the conventional method, which serves as the premise of the proposed method. Figure 2 visualizes the prediction results as a dot matrix. In the figure, the blue and gray regions indicate correctly predicted parts. While the method correctly predicts many parts where the correspondences evolve continuously, it can be seen that many errors occur in the portions where the progression is discontinuous in Fig.1.

Based on the prediction results produced by the existing method, the proposed method outputs 69 alerts (Table 3). Among the 69 alerts, 41 correctly identified cases where the existing method was incorrect.

Table 4 shows the confusion matrix when we define issuing an alert as Positive. In the table, Actually Positive indicates that the prediction of the existing

[6] https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2

Table 2. Prediction results of provision correspondences by the existing method

	Predicted Negative	Predicted Positive
Actually Negative	47	80
Actually Positive	76	312

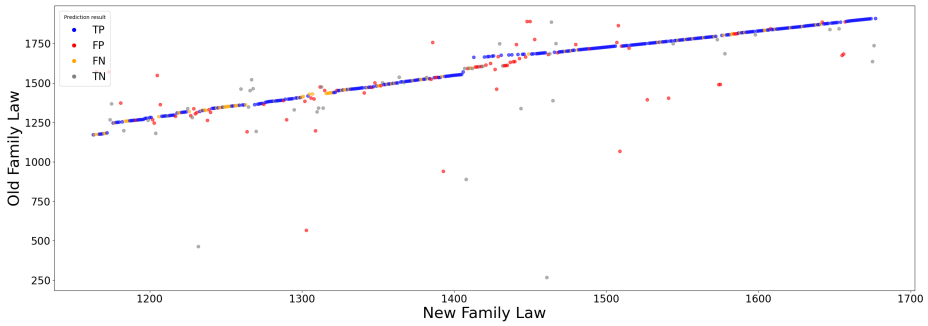


Fig. 2. Correspondence table of new and old sentences in the 1947 amendment to the Civil Code (family law)

method is incorrect and that an alert of either $\mathcal{A}_{\mathrm{FP}}$ or $\mathcal{A}_{\mathrm{FN}}$ is required, whereas Actually Negative indicates that the prediction of the existing method is correct. In this setting, Precision, Recall, and F1 were 0.594, 0.263, and 0.365, respectively (Eq. 1). Among the 28 incorrect alerts, 26 were cases where the existing method correctly aligned the sentences, and 2 were cases where it correctly judged that no alignment should be made.

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \qquad \text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \qquad \mathrm{F1} = \frac{2\,\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}} \tag{1}$$

6 Discussion

As shown in Table 4, the proposed method achieved Precision 0.594, Recall 0.263, and F1 0.365 for errors, meaning Actually Positive. This result means that, for the outputs of the existing method, the proposed method can present candidates of sentences to be checked with an accuracy of approximately 60%. The overall error-retrieval rate remains at approximately 30%. Therefore, the proposed method is better suited to prioritizing locations that should be checked under limited human verification cost than to exhaustively detecting errors.

We visualize the correct alerts with stars in Fig. 3 by the proposed method on the dot matrix. It can be seen that the method correctly infers as "incorrect" those erroneous correspondences that are far from the trajectory of the ground-truth correspondences. Conversely, in portions where the correspondences evolve

Table 3. Prediction results of provision correspondences by the existing method

Number of $\mathcal{A}_{\mathrm{FP}}$	Number of $\mathcal{A}_{\mathrm{FN}}$	Total
44	25	69

Table 4. Prediction results of provision correspondences by the existing method

	Predicted Negative	Predicted Positive
Actually Negative	331	28
Actually Positive	115	41

continuously, the method correctly infers that predictions of "no correspondence" are "incorrect".

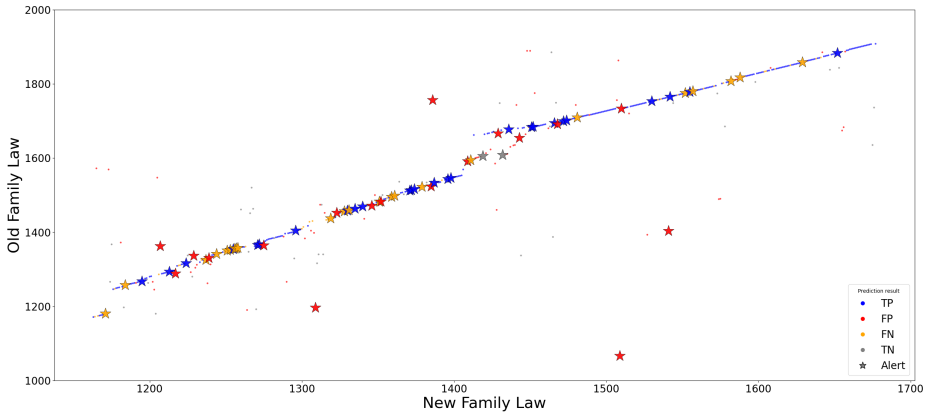


Fig. 3. Prediction results of the existing method and alerts by the proposed method

For actual statutory texts, Table 6 and Table 5 show the original Japanese and the English translations produced by ChatGPT 5.2 Instant. Note that if an entry is given in the form “ x.y.z ”, it denotes Article x, paragraph y, line z. In both $\mathcal{A}_{\mathrm{FP}}$ and $\mathcal{A}_{\mathrm{FN}}$ cases, many are formulaic sentences such as provisions on mutatis mutandis application. Since existing provision-alignment methods rely on textual similarity, they may fail to detect cases like those identified here, where the wording itself is highly similar. The proposed method appropriately detects such cases in which computers are prone to make mistakes, and thus can be effective as support when compiling provision-level legislative histories from an administrative standpoint.

Among the incorrect alerts, the two cases in which the existing method correctly judged that no alignment should be made are not located in regions where correspondences evolve continuously, but rather in regions where discontinuities occur. In Table 7, although the Japanese wording differs slightly, the English

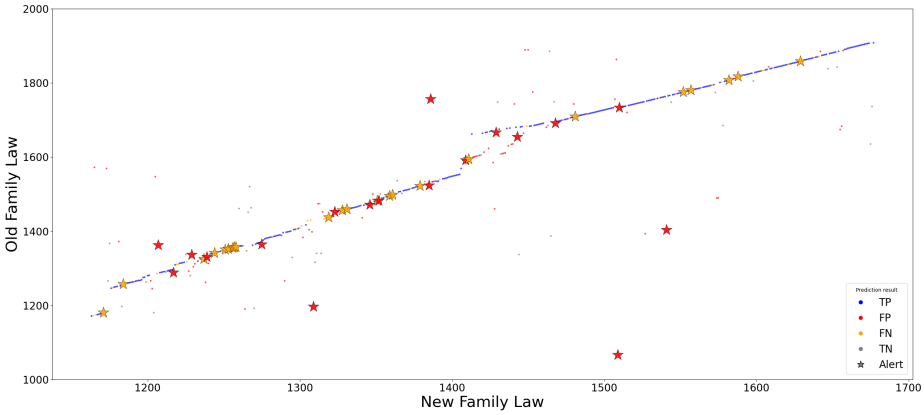


Fig. 4. Prediction results of the existing method and alerts by the proposed method (Only correct alert)

		Text	English translation by ChatGPT 5.2 Instant
ex.1 new law	753.1.1	未成年者が婚姻をしたときは、これによつて成年に達したものとみなす。	If a minor enters into marriage, the minor is deemed to have attained the age of majority.
old law	837.1.1	成年に達したる者は養子を為すことを得	A person who has attained the age of majority may adopt a child.
ex.2 new law	957.2.1	第七十九条第二項、第三項及び第九百二十八条乃至第九百三十五条の規定は、前項の場合にこれを準用する。但し、第九百三十二条但書の規定は、この限りでない。	The provisions of Article 79, paragraphs (2) and (3), and Articles 928 through 935 shall apply mutatis mutandis to the case set forth in the preceding paragraph; provided, however, that the proviso to Article 932 shall not apply.
old law	863.2.1	七百七十二条第二項、第三項及ひ第七百七十三条の規定は前項の場合に之を準用す	The provisions of Article 772, paragraphs (2) and (3), and Article 773 shall apply mutatis mutandis to the case set forth in the preceding paragraph.

Table 5. Extracted new-old sentence correspondences associated with $\mathcal{A}_{\mathrm{FP}}$

		Text	English translation by ChatGPT 5.2 Instant
ex.1 new law	728.2.1	夫婦の一方が死亡した場合において、生存配偶者が姻族関係を終了させる意思を表示したときも、前項と同様である。	The same shall apply when, upon the death of either spouse, the surviving spouse manifests an intention to terminate the relationship by affinity.
old law	729.2.1	夫婦の一方か死亡したる場合に於て生存配偶者か其家を去りたるとき亦同し	The same shall apply when, upon the death of either spouse, the surviving spouse leaves that household.
ex.2 new law	847.2.1	保佐人又はその代表する者と準禁治産者との利益が相反する行為については、保佐人は、臨時保佐人の選任を家事審判所に請求しなければならない。	With regard to any act in which the interests of a curator or a person represented by the curator conflict with those of a quasi-incompetent person, the curator shall petition the Family Court for the appointment of a temporary curator.
old law	909.2.1	保佐人又は其代表する者と準禁治産者との利益相反する行為に付ては保佐人は臨時保佐人の選任を親族会に請求することを要す	With regard to any act in which the interests of a curator or a person represented by the curator conflict with those of a quasi-incompetent person, the curator shall request the Family Council to appoint a temporary curator.

Table 6. Extracted new-old sentence correspondences associated with $\mathcal{A}_{\mathrm{FN}}$

sentences are exactly the same. In these two cases, the underlying institutions targeted by the old and new statutes differ, and therefore, from an administrative standpoint, these are sentences that should not be linked. However, from the perspective of legal practitioners, such correspondences may still be useful as references.

7 Summary

In this study, we proposed a method for detecting errors in the correspondences between new and old provisions predicted by a computer. Among the 69 alerts by the proposed method, we obtained 41 correct alerts. In this study, a correct alert was defined as one that is issued when the existing method is wrong about whether to align new and old texts, using strict correspondences (e.g., those adopted in administrative practice) as the ground truth. Therefore, these 41 alerts are useful when one aims to link provisions under a strict notion of correspondence. Moreover, under the present definition, even alerts that are not counted as correct may still be valuable for standpoints that emphasize referability, such as legal practitioners, because they can draw attention to correspondences that are not included in formal materials issued by administrative bodies.

In future work, we will provide the accumulated results as a GUI and develop a tool that comprehensively supports the task of compiling legislative histories of provisions.

		Text	English translation by ChatGPT 5.2 Instant
ex.1 new law	886.2.1	前項の規定は、胎児が死体で生まれたときは、これを適用しない。	The provisions of the preceding paragraph shall not apply if the child is born dead.
old law	968.2.1	前項の規定は胎児か死体にて生まれたるときは之を適用せす	The provisions of the preceding paragraph shall not apply if the child is born dead.
ex.2 new law	891.1.3	被相続人の殺害されたことを知つて、これを告発せず、又は告訴しなかつた者。但し、その者に是非の弁別がないとき、又は殺害者が自己の配偶者若しくは直系血族であつたときは、この限りでない。	A person who, knowing that the decedent was murdered, did not report or file a complaint thereof; provided, however, that this shall not apply if such person lacks the capacity to discern right from wrong, or if the murderer was the person ' s spouse or lineal blood relative.
old law	969.1.3	被相続人の殺害せられたることを知りて之を告発又は告訴せさりし者但其者に是非の弁別なきとき又は殺害者か自己の配偶者若くは直系血族なりしときは此限に在らす	A person who, knowing that the decedent was murdered, did not report or file a complaint thereof; provided, however, that this shall not apply if such person lacks the capacity to discern right from wrong, or if the murderer was the person ' s spouse or lineal blood relative.

Table 7. New-old sentence correspondences for which an alert was issued despite the existing method judging "no correspondence"

Acknowledgments. This work was partially supported by JSPS KAKENHI Grant Number 23K01052, and the Kansai University Fund for Domestic and Overseas Research Support Fund, 2025.

References

1. Dai-ichi Hoki Co., Ltd.: Survey and analysis on the current state and issues of legislative drafting affairs and the potential of issue resolution through a new editor system (Mar 2024), original title in Japanese: 法制事務の現状及び課題並びに新エディタシステムによる課題解決の可能性の調査・分析
2. of Internal Affairs, M., Communications: Publication of e-gov law search (e-laws legal data) (2017), https://www.soumu.go.jp/main_content/000492195.pdf, original title in Japanese: e-Gov 法令検索（e-LAWS の法令データ）の公開
3. of Internal Affairs, M., Communications, Agency, D.: About the law standard xml schema (version 3.0) (2020), original title in Japanese: 法令標準 XML スキーマについて (Version 3.0)
4. Maehara, T., Sano, T., Takenaka, Y.: Automating the creation of legislative article histories in japanese commercial law: A method for identifying corresponding articles before and after amendments. pp. 114–129 (2025). https://doi.org/10.1007/978-981-96-7071-0_8
5. Maehara, T., Takenaka, Y., Sano, T.: An examination of consistency in correspondence between old and new provisions of the amended civil code promulgated by the national diet of japan. In: Proceedings of the 31st Annual Meeting of the Association for Natural Language Processing. Japan (2025), original paper written in Japanese

6. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
7. Sano, T., Toyama, K., Masuda, T.: Proposal of a legal standard xml schema based on modern japanese acts and imperial orders. Journal of the Japan Society for Digital Archive **6**(s3), s226–s229 (2022). https://doi.org/10.24506/jsda.6.s3_s226

Automating Evaluation and Optimization of Prolog Literals for Traffic Rule Formalization

Wachara Fungwacharakorn[1[0000−0001−9294−3118]], May Myo Zin[1[0000−0003−1315−7704]], and Ken Satoh[1[0000−0002−9309−4602]]

Center of Juris-Informatics, ROIS-DS, Tokyo, Japan
{wacharaf, maymyozin, ksatoh}@nii.ac.jp

Abstract. Formalization is a foundational task in AI and Law. While large language models can assist in automating formalization by generating multiple plausible rule candidates, determining which candidate most accurately represents the original legal rule remains challenging. To address this issue, this paper introduces two additional stages – evaluation and optimization – into the process of automating the formalization of legal rules into Prolog. The evaluation stage conducts a granular analysis by decomposing rule candidates into individual literals in order to identify their contextual conditions, and then assesses these literals with respect to isomorphism and logical connectivity. Building on these literal-level insights, the optimization stage synthesizes an improved rule candidate. We implement these two stages using a chain-of-instruction approach with Gemini-3.0 and demonstrate its application to the formalization of traffic rules. The results indicate that Gemini-3.0 is capable of performing literal-level evaluation and optimization. However, some inconsistencies remain, particularly in the granularity of conditions and in the optimization of more complex rule structures. These findings suggest that the process should be iterative and incorporate a human-in-the-loop to ensure representational fidelity and logical coherence.

Keywords: Autoformalization · Prolog · Large Language Models · Legal Reasoning · Isomorphism · Traffic Rules

1 Introduction

Formalization is a foundational task in Artificial Intelligence (AI) and Law, focusing on translating legal text into formal representations that machines can reason over. With the rapid development of Large Language Models (LLMs), the automation of this process – often termed *autoformalization* – has emerged as a growing area of research, aiming to bridge natural language legal norms and computational logic systems. Typically, the automation consists of these stages:

1. **Translation**: LLMs translate the law into formal representations. Multiple candidates of translations can be generated at this stage.
2. **Validation**: LLMs check the grammatical and logical correctness of the candidates.

3. **Selection**: If there are multiple valid candidates, the LLM selects the best formalization.

However, selecting the best formalization is difficult in a legal context, as there are multiple qualities to be considered. Two qualities exemplified in this paper are isomorphism (how closely the structure maps to the original law) and cross-literal coherence (how variables in the representation are connected with other parts). Rather than simply selecting one, observing multiple candidates often reveals an improved formalization in addition to the candidates.

To address this issue, in this paper, we introduce two additional stages – evaluation and optimization – into the autoformalization process of legal rules into Prolog. The evaluation stage breaks down rule candidates into individual literals to assess isomorphism and cross-literal coherence at the literal level. Building on these insights, the optimization stage synthesizes an improved formalization.

We implement these two stages using a chain-of-instruction approach [12] with GEMINI-3.0 and demonstrate their application to the formalization of traffic rules recently studied in [11]. The results show that GEMINI-3.0 is capable of performing literal-level evaluation and optimization. However, they also reveal remaining inconsistencies and challenges, particularly in optimizing more complex conditions.

The remainder of this paper is organized as follows. Section 2 provides technical background on Prolog and exemplifies the qualities for legal formalization, alongside a motivating example involving traffic rules. Section 3 introduces the new evaluation and optimization stages. Section 4 describes the implementation of these stages using a chain-of-instructions approach with GEMINI-3.0 and presents the application to the formalization of traffic rules. Section 5 discusses the findings and the implications from the example. Finally, Section 6 concludes the paper and outlines directions for future work.

2 Background and Example

2.1 Prolog

Prolog (PROgrammation en LOGique) is a classic logic programming language. Its fundamental structure is an *atom*, expressed as `p(`$a_1, \ldots, a_n$`)` where `p` is a predicate name and $a_1, \ldots, a_n$ are its arguments. Each atom is typically interpreted as a first-order logic sentence. A *literal* refers to either an atom (a positive literal) or its negation (a negative literal). In Prolog, negation is handled as *negation as failure* (`not`), meaning `not p(`$a_1, \ldots, a_n$`)` is true if `p(`$a_1, \ldots, a_n$`)` cannot be derived.

A Prolog program consists of rules, each of which contains a positive literal as the *head* of the rule and a finite set of (positive or negative) literals as the *body* of the rule. A rule in Prolog is typically expressed as `H :- B1, ..., Bn.`, where `H` is the head and `B1, ..., Bn` is the body. A *fact* is determined as a rule with an empty body, typically expressed as `H.`

Researchers have long been interested in using Prolog to represent legal rules, such as the British Nationality Act [8] and the Canadian Income Tax Act [9]. A Prolog rule is typically used to represent a legal rule by adding a legal conclusion as the head of the rule and adding the conditions as the body. Meanwhile, a Prolog fact is typically used to represent a legal fact obtained from the case. The reason why some conclusion is derived or not can be simulated by using Prolog proof trees. Several adaptations have been introduced for Prolog, such as Logical English [4], which is a controlled natural language based on Prolog, or PROLEG [6], which is a Prolog variation to represent legal rules.

2.2 Formalization Qualities

Formalization is a task to translate natural language expressions into formal representations. Besides grammatical and logical correctness, various formalization qualities have been proposed with some of them specifically focused on formalization of legal rules. Those qualities include:

- **Presumability**: The early work [8] in formalization of legal rules discusses the nuance distinction between strong negations and negation as failure in legal reasoning. This later connects to defeasibility of rules and the switch of burden of proof [7].
- **Isomorphism** [1, 3]: This refers to a structural mapping between the legal text and its formal representation, ensuring that the logic follows the structure of the law itself.
- **Cross-literal Coherence** (termed in this paper): This ensures variables are linked across literals within a rule. A lack of coherence often manifests as *singleton variables* – variables used only once – which can indicate a breakdown in the rule's internal logic. Some works [2] on autoformalization have discussed prevention of singleton variables.

2.3 Motivating Example

In this paper, we focus on formalization of traffic rules into Prolog, as recently studied in [11]. Here is one example of traffic rules in their work:

Example 1. If the driver of a vehicle is driving on a carriageway with a solid white line, then, under section 127 of the Criminal Procedure Code, he must not cross it in order to identify an offender having committed an offence without any harmful consequences.

In [11], the authors explore several approaches instructing LLM to automate formalizations of traffic rules into Prolog from predefined predicates. From their research, there is one manually formalized by human and two other formalization candidates generated and validated by GPT-4. Candidates of formalization for Example 1 are shown below.

Code 1: Candidates of formalization for Example 1

```
"candidate-1": "prohibited(cross(Driver,Line), 'under section 127 of
    the Criminal Procedure Code') :- driving_on(Driver,Lane),
    solid_white(Line), (separated_by_line(Lane, Line, RightSide);
    separated_by_line(LeftSide, Line, Lane)), commit(Offender, Act),
    offence(Act), not(causes(Act, Consequence), harmful(Consequence))
    , do_by(identify(Driver, Offender), cross(Driver,Line))."
"candidate-2": "prohibited(cross(Driver,Line), 'section 127 of the
    Criminal Procedure Code'):- driving_on(Driver,Lane), solid_white(
    Line), (separated_by_line(Lane, Line, RightSide);
    separated_by_line(LeftSide, Line, Lane)), commit(Person, Act),
    offence(Act), not(harmful(Act)), do_by(identify(Driver, Person),
    cross(Driver, Line))."
"candidate-3": "prohibited(cross(Driver,Line)):- driving_on(Driver,
    Lane), solid_white(Line), (separated_by_line(Lane, Line,
    RightSide); separated_by_line(LeftSide, Line, Lane)), do_by(
    identify(Driver, Offender), cross(Driver,Line)), holds_according
    _to(section127CriminalProcedureCode, do_by(identify(Driver,
    Offender), cross(Driver,Line))), not(harmful(_))."
```

The candidates of formalization differ in how certain conditions are represented. For instance, `candidate-1` uses `not(causes(Act,Consequence), harmful(Consequence))`; `candidate-2` uses `not(harmful(Act))`; while `candidate-3` uses `not(harmful(_))`. They are intended to capture the same contextual condition, namely that *the identified offender has committed an offence without any harmful consequences*. Each representation employs negation as failure (`not`), indicating that harmful offences are treated as exceptions to the prohibition. Depending on how the predicates `harmful/1` and `do_by/2` are interpreted, all three representations can be considered correct. However, in terms of isomorphism, `not(causes(Act,Consequence), harmful(Consequence))` is the best as it is more closely reflects the original wording of the rule than others. Meanwhile, in terms of cross-literal coherence, `not(harmful(_))` is the worst as it omits a variable to share other literals.

3 Evaluation and Optimization Stages

In this paper, we introduce two additional stages to the standard autoformalization process, namely evaluation and optimization. The new stages of autoformalization can be depicted in Fig. 1.

After retrieving the candidates from the translation and validation stages, the evaluation stage performs a granular analysis by breaking each rule candidate down into its literals. This stage follows three main steps:

1. **Literal Decomposition**: Each rule body is decomposed into literals to identify the specific contextual condition it aims to represent.
2. **Condition Analysis**: Each identified condition is expressed as a proposition and determined whether it is an *exceptional* condition, which can be

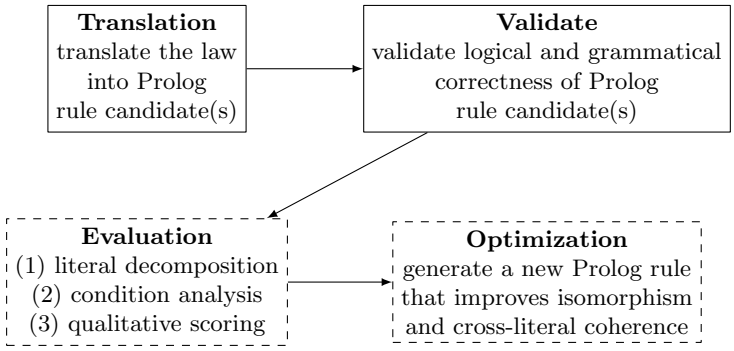


Fig. 1: Overview of evaluation and optimization stages

presumed to be true unless proven otherwise. For the completeness, it also checks for any contextual conditions in the reference rule that have not been represented by any candidates.

3. **Qualitative Scoring**: Each literal is assigned two scores on a scale of $0-5$:
   - *isomorphism score* (i): how well the literal maps to the original wording of the reference rule;
   - *cross-literal coherence score* (c): how effectively variables in the literal are referenced by other literals in the rule, ensuring logical connectivity.

The optimization stage leverages the literal-level evaluation results, particularly the identified conditions and their corresponding scores, to generate a Prolog representation that improves isomorphism and cross-literal coherence, potentially differing from the initial candidates.

4 Implementation and Demonstration

We implement these stages using the chain-of-instructions approach [12] by instructing GEMINI-3.0 with the following prompt template:

Code 2: Prompt Template

```
You are an expert in Computational Law and Prolog programming.
Below are multiple formalizations of the same reference legal rule:

### 1. THE REFERENCE RULE (Ground Truth):
"{reference_rule}"

### 2. CANDIDATE FORMALIZATIONS:
{formalized_candidates}

### 3. YOUR TASK (Literal-Level Decomposition):
For EACH formalization, perform the following deep dive:
```

```
1. **Decomposition**:
   - Extract every Prolog literal (a Prolog atom or a negation of
       Prolog atom) from the rule body.
   - Extract every Prolog variable from the rule body.

2. **Proof Structure Analysis**
   - Identify every condition of the reference rule that has been
       represented by a literal in the formalization.
   - Identify every condition of the reference rule that has NOT been
       represented by a literal in the formalization.
   - Write the condition as a full declarative statement that can be
       classified as either true or false.

3. **Exception Analysis**
   - Assign YES/NO to each condition based on whether it is an
       exceptional condition that is presumed to be true unless the
       opposite is shown.

4. **Isomorphism Analysis**:
   - Assign an isomorphism score (0-5) to each literal based on how
       well it can be mapped to the condition in terms of predicates
       and argument orders.
   For example, "parent(A,B)" is better mapped to "A is a parent of B"
        than "parent(B,A)" which is better mapped to "B is a parent
       of A".

5. **Cross-literal Analysis**:
   - Assign a cross-literal score (0-5) to each literal based on how
       variables are referred to in other literals within the same
       Prolog Rule (it should be referred to in at least one other
       literal).

6. **Synthesis**:
   - Create an optimized Prolog code by considering each condition and
        assign an isomorphism score (0-5) and a cross-literal score
       (0-5) to each literal.
   - Do NOT invent a new predicate

### 4. OUTPUT FORMAT (Strict JSON):
Return exactly this JSON structure:
{{
  "analysis": {{
    "condition": "condition_1",
    "exception": "YES/NO",
     "formalizations":{{
        "candidate-1": {{"literal":"literal_1", "isomorphism_score
            ":(0-5), "cross_literal_score":(0-5)}},
        ... (repeat for other formalizations)
        "optimized": {{"literal":"optimized_literal", "
            isomorphism_score":(0-5), "cross_literal_score":(0-5)}},
```

```
    }},
    ... (repeat for other conditions)
  }},
  "optimized_prolog": "optimized_prolog_code"
}}
```

Table 1 shows an example application to the formalization of the traffic rule in Example 1. Each row presents contextual conditions identified by the model and quality scores assigned to each literal. For example, a contextual condition – *the offence resulted in no harmful consequence*^{ex} – is assessed as an exceptional condition, indicated by a superscript ^{ex}, showing that the condition is presumed to be true unless a harmful consequence is proven.

In representing this condition, the evaluation assigns `not(causes(Act, Consequence), harmful(Consequence))` in `candidate-1` full isomorphism and cross-literal scores ($i = 5, c = 5$). In contrast, `not(harmful(Act))` in `candidate-2` receives an isomorphism score of 3 out of 5 and a cross-literal score of 4 out of 5 ($i = 3, c = 4$). This demonstrates that `candidate-1` better reflects the original wording of the reference rule and exhibits superior cross-literal coherence. These findings suggest that CANDIDATE-1 serves as an optimized Prolog rule.

It can also be seen that the evaluation is occasionally affected by the variable names. For instance, for the condition *an offender has committed an offence*, the evaluation assigns a higher isomorphism score to `commit(Offender,Act), offence(Act)` than to `commit(Person,Act),offence(Act)`, even though both representations are computationally equivalent. This indicates that the evaluation criteria extend to the assessment of variable names.

Nevertheless, the results also reveal several inconsistencies and limitations in the evaluation and optimization stages. For example, Table 2 shows an example result of evaluating and optimizing candidates of formalization for the following traffic rule.

Example 2. If a driver is driving on a carriageway with a solid white line, he must not cross on this line, neither for the purpose of turning left.

Candidates of formalization for Example 2 are shown below.

Code 3: Candidates of Formalization for Example 2

```
"candidate-1": "prohibited(cross(Driver,Line)):- driving_on(Driver,
    Lane), (separated_by_line(Lane, Line, RightLane);
    separated_by_line(LeftLane, Line, Lane)), solid_white(Line),
    do_by(turn_left(Driver), driving_on(Driver, Line)).",
"candidate-2": "prohibited(cross(Driver,Line)):- driving_on(Driver,
    Lane), solid_white(Line), (separated_by_line(Lane, Line,
    RightLane); separated_by_line(LeftLane, Line, Lane)), do_by(
    turn_left(Driver), cross(Driver,Line))."
"candidate-3": "prohibited(cross(Driver,Line)):- driving_on(Driver,
    Lane), solid_white(Line), (separated_by_line(Lane, Line,
    RightSide); separated_by_line(LeftSide, Line, Lane)), not(do_by(
    turn_left(Driver), cross(Driver,Line)))."
```

Table 1: Result of evaluating and optimizing candidates of formalization for Example 1 using GEMINI-3.0

Contextual Condition	candidate-1	candidate-2	candidate-3	Optimized
A driver is driving on a carriageway (lane)	driving_on(Driver, Lane) ($i = 5, c = 5$)	driving_on(Driver, Lane) ($i = 5, c = 5$)	driving_on(Driver, Lane) ($i = 5, c = 5$)	driving_on(Driver, Lane) ($i = 5, c = 5$)
There is a solid white line separating the lanes.	solid_white(Line), (separated_by_line(Lane, Line, RightSide);separated_by_line (LeftSide, Line, Lane)) ($i = 5, c = 5$)	solid_white(Line), (separated_by_line(Lane, Line, RightSide);separated_by_line (LeftSide, Line, Lane)) ($i = 5, c = 5$)	solid_white(Line), (separated_by_line(Lane, Line, RightSide);separated_by_line (LeftSide, Line, Lane)) ($i = 5, c = 5$)	solid_white(Line), (separated_by_line(Lane, Line, RightSide);separated_by_line (LeftSide, Line, Lane)) ($i = 5, c = 5$)
An offender has committed an offence.	commit(Offender, Act), offence(Act) ($i = 5, c = 5$)	commit(Person, Act), offence(Act) ($i = 4, c = 5$)	null ($i = 0, c = 0$)	commit(Offender, Act), offence(Act) ($i = 5, c = 5$)
The offence resulted in no harmful consequences.[ex]	not(causes(Act, Consequence), harmful(Consequence)) ($i = 5, c = 5$)	not(harmful(Act)) ($i = 3, c = 4$)	not(harmful(Act)) ($i = 2, c = 0$)	not(causes(Act, Consequence), harmful(Consequence)) ($i = 5, c = 5$)
The crossing of the line is performed in order to identify the offender.	do_by(identify(Driver, Offender), cross(Driver, Line)) ($i = 5, c = 5$)	do_by(identify(Driver, Person), cross(Driver, Line)) ($i = 5, c = 5$)	do_by(identify(Driver, Offender), cross(Driver, Line)) ($i = 5, c = 5$)	do_by(identify(Driver, Offender), cross(Driver, Line)) ($i = 5, c = 5$)

Table 2: Result of evaluating and optimizing candidates of formalization for Example 2 using GEMINI-3.0

Contextual Condition	candidate-1	candidate-2	candidate-3	Optimized
A driver is driving on a carriageway (specifically a lane)	driving_on(Driver, Lane) ($i = 5, c = 5$)	driving_on(Driver, Lane) ($i = 5, c = 5$)	driving_on(Driver, Lane) ($i = 5, c = 5$)	driving_on(Driver, Lane) ($i = 5, c = 5$)
The carriageway has a solid white line.	solid_white(Line) ($i = 5, c = 5$)	solid_white(Line) ($i = 5, c = 5$)	solid_white(Line) ($i = 5, c = 5$)	solid_white(Line) ($i = 5, c = 5$)
The line separates the driver's current lane from an adjacent side (left or right).	(separated_by_line(Lane, Line, RightLane);separated_by_line(LeftLane, Line, Lane)) ($i = 5, c = 5$)	(separated_by_line(Lane, Line, RightLane);separated_by_line(LeftLane, Line, Lane)) ($i = 5, c = 5$)	(separated_by_line(Lane, Line, RightLane);separated_by_line(LeftLane, Line, Lane)) ($i = 5, c = 5$)	(separated_by_line(Lane, Line, _); separated_by_line(_, Line,Lane)) ($i = 5, c = 5$)
The action is performed for the purpose of turning left.	do_by(turn_left(Driver), driving_on(Driver, Line)) ($i = 2, c = 5$)	do_by(turn_left(Driver), cross(Driver, Line)) ($i = 4, c = 5$)	not(do_by(turn_left(Driver), driving_on(Driver, Line))) ($i = 1, c = 5$)	null ($i = 0, c = 0$)

A comparison between the two tables reveals some inconsistencies. The evaluation in Table 1 identifies the condition as *there is a solid white line separating the lanes*, whereas the evaluation in Table 2 identifies the similar condition as two separate components: *the carriageway has a solid white line* and *the line separates the driver's current lane from an adjacent side (left or right)*. Moreover, the optimization in Table 1 uses the variables `LeftSide` and `RightSide`, which appear as singleton variables (variables used only once in a rule). In contrast, the optimization in Table 2 uses the anonymous variable (`_`). Furthermore, Table 2 shows that the final condition – *the action is performed for the purpose of turning left* – is a difficult condition to optimize. Consequently, the optimization stage simply omits this condition in the resulting rule.

5 Discussion

As shown in our demonstration, while Gemini-3.0 demonstrates capabilities in identifying exceptional conditions and evaluating literal-level quality, it still exhibits inconsistencies in condition granularity and variable optimization. These variations indicate that a single-pass translation often falls short of the precision required for legal reasoning, aligning with prior studies on autoformalization in general [5, 10]. This suggests the process should be iterative. Moreover, the interaction between formal representation (Prolog) and informal but human-readable representation (e.g., Logical English) is shown to be crucial in our demonstration. Therefore, the proposed evaluation and optimization stages serve as a critical feedback loop, allowing humans to refine representations through successive literal-level analyses.

6 Conclusion and Future Work

In this paper, we explore a pipeline for autoformalization of legal rules into Prolog, focusing on two additional stages: evaluation and optimization. By decomposing rule candidates into individual literals, this pipeline enables a more granular assessment of legal formalization qualities such as isomorphism and cross-literal coherence. Our demonstration with Gemini-3.0 shows that the model can perform a literal-level evaluation and optimization.

However, there are still limitations, such as inconsistencies in condition identification and the omission of complex conditions in optimization. It indicates that fully automated formalization remains a challenge. Future work will focus on stabilizing the granularity of these evaluations and exploring iterative prompting strategies for evaluating and optimizing formalizations with human-in-the-loop.

Acknowledgements

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 appreciate all comments from the reviewers.

References

1. Bench-Capon, T.J., Coenen, F.P.: Isomorphism and legal knowledge based systems. Artificial Intelligence and Law **1**(1), 65–86 (1992)
2. Coppolillo, E., Calimeri, F., Manco, G., Perri, S., Ricca, F.: LLASP: Fine-tuning large language models for answer set programming. In: Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning. pp. 834–844 (2024)
3. Karpf, J.: Quality assurance of legal expert systems. In: Pre-proceedings of the third international conference on Logica, Informatica, Diritto, CNR, Florence. pp. 411–440 (1989)
4. Kowalski, R., Dávila, J., Calejo, M.: Logical english as a programming language for the law. In: Proceedings of Programming Languages and the Law 2022 (2022)
5. Mensfelt, A., Cucala, D.T., Franco, S., Koutsoukou-Argyraki, A., Trencsenyi, V., Stathis, K.: Towards a common framework for autoformalization. In: Proceedings of the Second International Workshop on Next-Generation Language Models for Knowledge Representation and Reasoning. NeLaMKRR 2025 (2025)
6. Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., Takano, C.: PROLEG: an implementation of the presupposed ultimate fact theory of japanese civil code by prolog technology. In: JSAI international symposium on artificial intelligence. pp. 153–164. Springer (2010)
7. Satoh, K., Tojo, S., Suzuki, Y.: Formalizing a switch of burden of proof by logic programming. In: Proceedings of the first international workshop on Juris-informatics (JURISIN 2007). pp. 76–85. Miyazaki Japan (2007)
8. Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., Cory, H.T.: The British Nationality Act as a logic program. Communications of the ACM **29**(5), 370–386 (Apr 1986)
9. Sherman, D.M.: A prolog model of the income tax act of Canada. In: Proceedings of the 1st international conference on Artificial intelligence and law. pp. 127–136. New York, NY, USA (1987)
10. Vandevelde, S.: On the role of domain experts in llm-based knowledge formalization. In: Proceedings of the Second International Workshop on Next-Generation Language Models for Knowledge Representation and Reasoning. NeLaMKRR 2025 (2025)
11. Zin, M.M., Borges, G., Satoh, K., Fungwacharakorn, W.: Towards machine-readable traffic laws: Formalizing traffic rules into PROLOG using LLMs. In: Proceedings of the Twentieth International Conference on Artificial Intelligence and Law. pp. 327–336 (2025)
12. Zin, M.M., Satoh, K., Borges, G.: Leveraging llm for identification and extraction of normative statements. In: Legal Knowledge and Information Systems, pp. 215–225. IOS Press (2024)

A Dataset and Benchmark for Resolving Legal Cross-References in Japanese Export Control Regulations

Rafal Rzepka[1][0000−0002−8274−0875], Shinji Muraji[2], and Akihiko Obayashi[3]

[1] Faculty of Information Science and Technology, Hokkaido University, Kita-ku, Kita 14, Nishi 9, Sapporo, Japan
rzepka@ist.hokudai.ac.jp
http://kabura.info
[2] Graduate School of Information Science and Technology, Hokkaido University, Kita-ku, Kita 14, Nishi 9, Sapporo, Japan
shinjimuraji@ist.hokudai.ac.jp
[3] Faculty of Engineering, Hokkaido University, Kita-ku, Kita 13, Nishi 8, Sapporo, Japan
obayashi@eng.hokudai.ac.jp

Abstract. This paper introduces a simple, manually annotated dataset for resolving complex cross-references within Japanese export control regulations. While automatically constructing a complete legal knowledge graph is desirable, it requires meticulously labor-intensive expert verification. A significant challenge in this domain is the prevalence of non-contiguous citations. Specifically, references that define a range of articles while explicitly excluding subsets (e.g., "Articles 10 through 15, excluding Article 12"). To address the scarcity of resources for this task in Japanese, we also develop a synthetic data generation pipeline to enable the fine-tuning LLMs. We conduct a comprehensive evaluation of standard paradigms like a rule-based system, Retrieval-Augmented Generation (RAG), or fine-tuned Large Language Models (LLMs) with graph support. Our experimental results demonstrate that all methods are far from high performance which is a critical requirement for legal compliance tasks. We plan to release this dataset as a challenging benchmark to stimulate further research into robust legal citation resolution.

Keywords: Cross-Referencing · Citation Extraction · Export Control Regulations · Benchmark Dataset · Japanese Language

1 Introduction

The digitization of legal frameworks has transitioned from a passive archival activity to the creation of active, computable legal infrastructures. In the high-stakes domain of international trade compliance, in our research within the scope of Japanese export control regulations such as the Foreign Exchange and Foreign Trade Act (FEFTA), the precise, algorithmic resolution of legal citations is not

merely a matter of efficiency but of national security and regulatory adherence. The complexity of these regulations, characterized by deep hierarchical nesting and intricate cross-referencing strategies, presents a unique challenge to the field of Natural Language Processing (NLP). The state-of-the-art in this domain has shifted from purely rule-based regular expressions and heuristic parsing to sophisticated neuro-symbolic architectures that blend the semantic flexibility of LLMs with the rigorous structural integrity of logic programming. Japanese export control regulations are basically enumerated lists of controlled goods and technologies, often referred to as "List Controls." These lists are not static "A is B" type statements but are embedded within a complex network of relations outside a given paragraph. A single determination of whether a specific dual-use technology requires an export license often necessitates traversing a reference path that spans multiple documents, resolving dynamic ranges (e.g., "Items (b) through (d) in Category 2", "excluding the technology specified in Item 3", etc.), and interpreting semantic definitions that modify the scope of the control. The failure to accurately expand such reference range, such as missing a specification mentioned in another passage or failing to exclude a repealed provision – can result in severe legal penalties for exporters. Consequently, the automated systems designed to assist in this process must achieve near-perfect recall and precision, a requirement that traditional information retrieval methods struggle to meet. XML files provided by Japanese government via e-Gov site[4] do not follow international legal schemas like Akoma Ntoso (LegalDocML)[5], hence automatic methods might become helpful in the process of adding references to a wide range of regulatory documents in Japanese language.

2 Background and Challenges in Legal Reference Mapping

In our HokBot project [7, 12], we have been trying to develop an online assistant that helps researchers to determine if their work is regulated by export control laws. With only one expert on board, the data creation and expansion [8] is difficult, and we have tried to implement various neural approaches [9, 11] trying to minimize risks of hallucinations by suppressing the functionality. Currently we are experimenting with LLMs, which have their own problems when deployed to the public [10] and cannot be trusted as more graph-like approaches. To solve the problem, creating computer-readable version of export control regulations is preferable and Japanese government partially provides them via e-Gov mentioned above, but another problem we faced is that they have no cross references, while legal regulations often use cross-reference expressions citing specific parts of the document set (e.g. "Category 2, Items (b)–(d)"). Constructing an exhaustive list or graph of these references automatically requires not only an understanding of the document's hierarchy and scope, but also the ability to

[4] https://www.e-gov.go.jp
[5] http://akomantoso.info

parse natural-language expressions so that all related parts are correctly linked to the source sentence. Early computational approaches relied on rule-based patterns and regular expressions to detect citations (e.g. "Art. 156" or "paragraph (3)"). For example, Adedjouma et al. [1] use GATE[6] plus hand-crafted regex patterns and a first-order logic engine (CrocoPat [2]) to interpret cross-references in Luxembourg law. Similarly, Gheewala et al. [5] built a regex-based extractor in GATE for legal citations. These systems can expand simple references (e.g. "articles 14, 61, 91 or 95" to "[article 14, article 61, ... article 95]") and ranges ("paragraphs 99 to 102" to "[paragraph 99, paragraph 100, paragraph 101, paragraph 102]"), but require extensive manual pattern engineering. They also often fail on complex or ambiguous expressions like "paragraphs 1 to 3 in the previous paragraph except items described in article 4" which requires managing knowledge of hierarchical structure of the whole document set. Additionally, there are logical expressions that decide on which parts must be linked, for example exceptions (*moshikuwa, matawa*/or, *nozoku*/except of) or conjunctions (*oyobi, narabini*/and). For this research we construct manual list of references and investigate how this burdensome process could be mechanized to address updates and applications to other legal domains. This paper is structured as follows: in Section 3 we present research related to automatic approaches to the cross reference expressions and in Section 4 we describe the created dataset for legal citation resolution in Japanese. From Section 5 to 8, we describe baselines for different standard approaches. After discussing the results in Section 9, we conclude the paper (Section 10).

3 Related Work

3.1 Machine Learning and Heuristic Pipelines

More recent work applies supervised learning to reference detection and resolution. Tran et al. (2014) introduce a four-step ML pipeline for Japanese law: mention detection (using CRFs), contextual information extraction, candidate retrieval, and antecedent determination [18]. On the Japanese National Pension Law corpus they reported 80.06% F1 at mention detection and 67.02% end-to-end F1 and in follow-up research they achieve 91.6% detection and 88.5% final F1 [17]. Their novelty lies in resolving sub-document references (identifying the exact clause or phrase cited, not just the target document). However, their statistical model still uses regex in the resolution step and requires a sizable annotated corpus (JNPL) for training. Other ML pipelines follow similar multi-stage frameworks: for instance, Sannier et al. [13] (extended version of Adedjouma's work described in the previous section) formalize legal schema and apply NLP patterns to both detect and resolve references. They validate patterns derived from Luxembourg regulations on Ontario law, showing portability

[6] GATE (General Architecture for Text Engineering) is an open-source Natural Language Processing (NLP) framework used to develop, manage, and execute complex text-processing rules [4].

across languages, but their approach remains largely rule-driven, relying on the legal text's structure and large handwritten grammars.

3.2 Neural and Transformer Approaches

Recently, end-to-end neural models have been applied to citation extraction. Thuy et al. (2023) propose a joint Transformer encoder–decoder that simultaneously extracts reference mentions and their relations in legal documents [16]. Their model achieved an F1 of 99.4% on a Vietnamese legal corpus (61K references), suggesting that a properly trained transformer can excel at this task in a well-defined setting. However, such models typically need large, task-specific datasets, which Japanese export control texts lack. Moreover, pure neural models may still struggle with the logical consistency of ranges and nested structures unless explicitly constrained. Blair-Stanek et al. (2024) explicitly tested large LLMs (GPT-4, Claude) on "basic legal text handling" tasks and found that off-the-shelf LLMs perform poorly without fine-tuning [3]. They created a benchmark of lawyer-level tasks (e.g. locating a clause by reference) and showed fine-tuned models can reach near 100% accuracy, implying that even simple legal-text tasks (like reference mapping) are nontrivial for general LLMs. More recently, Sargeant et al. (2025) have concentrated on legal citation detection in court judgments testing a variety of models showing that BERT family is capable to outperform GPT-4.1 model and regular expression approach performs poorly when compared to Transformers [14]. However, their work is closer to LER (Legal Entity Recognition) task, where a model is trained to discover ranges containing correct answers, while in citation expansion the main difficulty is that related items very often map beyond what is directly stated in the text, or stated but should not be retrieved (like a range of articles with an exception).

The data comprises 134 regulatory cases with total count of 6,695 relevant addresses mapped. The average number of list items per case is 49.96, and the longest list contains 555 items[7].An example of a single entry with its translation is shown in Figure 1.

4 Reference Dataset Preparation

Basic document in this research is based on is so called Consolidated Matrix (*gattai matorikusu*) which refers to an integrated classification table that merges multiple regulatory lists or control categories into a single, unified framework. In the context of export control, it functions as a harmonized matrix that aligns technical specifications, regulatory items, and corresponding legal provisions across different ordinances, enabling consistent mapping and cross-referencing of controlled technologies and items.

[7] The data with more detailed information is available at https://github.com/Language-Media-Lab/ExportControlReferences-ja

4.1 Scope and Selection Criteria

The primary objective was to capture the most logically complex cross- references within the Foreign Exchange and Foreign Trade Act (FEFTA) framework. The Japanese export control system is organized as a hierarchical stack: originating from the FEFTA (containing approximately 73 articles), it is further defined by two Cabinet Orders—the Export Trade Control Order for physical goods and the Foreign Exchange Order for technology and software. Below these lies the Ministerial Order Specifying Goods and Technologies Pursuant to the Provisions of the Appended Table 1 of the Export Trade Control Order and the Appended Table of the Foreign Exchange Order, which contains the granular technical specifications.

4.2 Annotation Process and Guidelines

The dataset was manually constructed by the second author, native speaker of Japanese who have worked for the Hokbot export control expert chatbot project[8] since 2022. The annotation followed a three-step guideline:

- Contextual Mapping: Identifying the source paragraph (pa) and its absolute hierarchical address (ad).
- Exhaustive Extraction: Identifying all mentions of "Items" or "Articles" including those elided by range expressions or semantic modifiers (e.g., "excluding the following...").
- Canonical Resolution: Mapping each mention to its exhaustive set of leaf-node identifiers within the regulatory hierarchy to create the relevant_list (when conditional branches were present, they were listed as separate items).

Task Definition: Hierarchical Address Expansion To clarify the experimental objective, we formally define the task as Hierarchical Address Expansion. This task transcends simple "citation detection" by requiring the system to:

- Extract relative or abbreviated natural language mentions.
- Normalize these mentions into a consistent numeral system (e.g., resolving Arabic vs. Kanji numeral discrepancies).
- Expand root citations into a unique set of leaf-level identifiers using a structured knowledge graph to ensure "legal safety" and near-perfect recall.

Selection of Identifiers While the source documents are provided in XML format via e-Gov, these files lack internal cross-reference metadata or stable XPaths for nested sub-items. Consequently, we utilized canonical hierarchical strings (e.g., "Article 15, Paragraph 1, Item 5") as identifiers. This format ensures the benchmark is interoperable across different model architectures (including rule-based parsers and Large Language Models) that must navigate the semantic prose of Japanese regulations.

[8] https://www.hokudai.ac.jp/research/export-control/, access credentials upon request.

```
{
  "seirei": "2(1)_5(2)",
  "Category": "原子力",
  "RegNum": "輸出令別表第1の2の項の貨物の技術",
  "sekkei":1,
  "seizo":1,
  "shiyo":1,
  "shorei":{
    "第十五条": {
      "第1項": {
        "ad": "貨物等省令第十五条第1項",
        "pa":
            "外国為替令（以下「外為令」という。）別表の二の項（一）の経済産業省令で定める技術は、
            次のいずれかに該当するものとする。",
        "第五号": {
          "ad": "貨物等省令第十五条第1項第五号",
          "pa":
              "第一条第六号（リチウムの同位元素の分離用の装置に限る。）、第二十五号、第二十九号、
              第五十三号又は第五十九号のいずれかに該当する貨物の設計、製造又は使用に係る技術（プ
              ログラムを除く。）"
        }
      }
    }
  },
  "relevant_list": [
    "貨物等省令第一条第1項第二十五号",
    "貨物等省令第一条第1項第二十九号",
    "貨物等省令第一条第1項第五十三号",
    "貨物等省令第一条第1項第五十三号イ",
    "貨物等省令第一条第1項第五十三号ロ",
    "貨物等省令第一条第1項第五十九号"
  ],
  "ref_address": [
    "貨物等省令第十五条第1項第五号"
  ],
  "kisei_komoku": "該当貨物の設計、製造又は使用に係る技術が規制されます。"
},
```


(a) Original Japanese gold set


```
{
  "cabinet_order": "2(1)_5(2)",
  "Category": "Nuclear Energy",
  "RegNum": "Technology for goods in Item 2 of Appended Table 1 of the Export
      Control Order",
  "design": 1,
  "manufacturing": 1,
  "use": 1,
  "ministerial_ordinance": {
    "Article 15": {
      "Paragraph 1": {
        "ad": "Ministerial Ordinance on Trade Goods, etc. Article 15, Paragraph 1",
        "pa": "The technology specified by the Ordinance of the Ministry of Economy,
            Trade and Industry in Item 2 (1) of the Appended Table of the Foreign
            Exchange Order (hereinafter referred to as the 'Exchange Order') shall
            fall under any of the following.",
        "Item 5": {
          "ad": "Ministerial Ordinance on Trade Goods, etc. Article 15, Paragraph 1,
              Item 5",
          "pa": "Technology (excluding programs) related to the design,
              manufacturing, or use of goods falling under any of Article 1, Item 6
              (limited to equipment for the separation of lithium isotopes), Item
              25, Item 29, Item 53, or Item 59."
        }
      }
    }
  },
  "relevant_list": [
    "Ministerial Ordinance on Trade Goods, etc. Article 1, Paragraph 1, Item 25",
    "Ministerial Ordinance on Trade Goods, etc. Article 1, Paragraph 1, Item 29",
    "Ministerial Ordinance on Trade Goods, etc. Article 1, Paragraph 1, Item 53",
    "Ministerial Ordinance on Trade Goods, etc. Article 1, Paragraph 1, Item 53 (a)",
    "Ministerial Ordinance on Trade Goods, etc. Article 1, Paragraph 1, Item 53 (b)",
    "Ministerial Ordinance on Trade Goods, etc. Article 1, Paragraph 1, Item 59"
  ],
  "ref_address": [
    "Ministerial Ordinance on Trade Goods, etc. Article 15, Paragraph 1, Item 5"
  ],
  "regulatory_item": "Technology related to the design, manufacturing, or use of the
      relevant goods is regulated."
}
```


(b) English translation

Fig. 1: An example of the manually crafted regulatory data on export control ("relevant_list" is the list of citations related to paragraph ("pa") under an address ("ad") inputs.

5 Graph-based Baselines with LLMs

Currently, one of the first choices would be to feed a proprietary Large Language Model with the regulations and simply prompt it to extract the citations. However, while LLMs excel at interpreting the semantic nuances and relative references within natural language, they often lack the symbolic reasoning required to exhaustively and accurately map hierarchical regulatory structures. Therefore, for the first baseline to be investigated, we choose a hybrid approach of combining a large language model with a deterministic knowledge graph to decrease the risk of hallucinations and ensure better legal veracity. By using a knowledge graph as an anchor, the system makes sure that every extracted citation is authenticated against a valid address space and expanded with mathematical precision. This separation aims at reproducible and stable framework that maintains higher level of accuracy necessary for the complex nesting characteristic of export control regulations. The implementation of the reference expansion algorithm follows a deterministic two-stage architecture that combines neural extraction with a rule-based expansion engine.

5.1 Initial Step: Prompting

In the initial stage, a large language model is provided with the target legal text and is prompted to extract all mentions of regulatory items and articles while

resolving range expressions and omissions into a discrete, comma-separated list of strings (see Figure 2 in the Appendix). To ensure higher precision in this extraction, the model normalizes various numeric formats, such as converting Arabic numerals into the formal kanji numeral system used in Japanese legislative documents[9]. This step transforms relative mentions like sub-item identifiers into potentially absolute references by prepending the provided contextual article and paragraph headers (prefix "Ministerial Ordinance on Trade Goods, etc." is added as the matched list always contains it). It should be noted that this augmentation, together with the normalization of numbering, increases the likelihood of correct matching with the gold standard and consequently improves performance, albeit at the cost of reduced realism with respect to naturally occurring legal NLP scenarios.

5.2 Second Step: Graph Matching

The second stage processes these extracted strings through a legal knowledge graph represented as a sorted list of all valid hierarchical addresses within the target regulation. For each extracted reference string, the algorithm performs a prefix-based lookup to identify all corresponding leaf nodes in the regulatory hierarchy. This is achieved by utilizing a binary search, specifically through a bisect-left operation on the sorted address list, to locate the starting index of the reference string. The algorithm then iterates forward from this index, collecting every address that begins with the specific reference prefix until an address is encountered that no longer matches. This method ensures that a high-level reference to a specific article automatically expands to include all its nested paragraphs, items, and sub-items. The final output is a unique set of identifiers.

5.3 Results of LLM With Graph Approach

We test three proprietary (`gpt4o`, `gpt4o-mini`, `gpt5`) and local (`gpt-oss-20b`, `gpt-oss-120b`, `Qwen3-14B`) models. The local models are quantized (4bit) versions and are run on Mac Studio M3 (512GB memory) computer via LM Studio (we only use the official recommended model versions). Temperature is set to 0 (except `gpt5`, which does not allow temperature setup), and max completion tokens are set to 8,192 as previously tested size of 500 could not match longer lists of cited articles for local models. Although 500 tokens limit yielded slightly better results for OpenAI models, we report longer max completion results for consistency (we also keep this length in further experiments). `qwen3-30b` model's version is A3B-2507 featuring "significant improvements in general capabilities including instruction following, logical reasoning, text comprehension, mathematics, science, coding and tool usage". The performance of the graph-based approach is reported in Table 1.

[9] This processing must be performed carefully, as the type of numeral used (Chinese characters or Arabic digits) carries important information in Japanese legal numbering.

Table 1: Performance comparison across different models using Macro and Micro averages. Default max token length is 8,192 or 500 (then indicated in brackets).

	Macro-Averaged (Per-Sample)			Micro-Averaged (Global)		
Model	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
`gpt4o-mini`	0.3448	0.3362	0.3207	0.7885	0.1864	0.3016
`gpt4o-mini` (500)	0.3328	0.3268	0.3110	0.7685	0.2086	0.3281
`gpt4o`	0.3808	0.3742	0.3705	0.8987	0.2275	0.3631
`gpt4o` (500)	0.3874	0.3758	0.3724	0.8639	0.2353	0.3699
`gpt5`	**0.5392**	**0.5291**	**0.5234**	0.8791	**0.3546**	0.5054
`gpt5` (500)	0.5388	0.5281	0.5223	**0.8868**	0.3536	**0.5056**
`gpt-oss-20b`	0.4389	0.4367	0.4099	0.7690	0.2667	0.3960
`gpt-oss-20b` (500)	0.4226	0.4048	0.3828	0.7482	0.1693	0.2761
`gpt-oss-120b`	0.5158	0.5113	0.4983	0.8872	**0.3574**	**0.5095**
`gpt-oss-120b` (500)	0.4013	0.4067	0.3918	**0.9172**	0.1698	0.2866
`qwen3-8b`	0.2793	0.3295	0.2743	0.5992	0.1700	0.2648
`qwen3-8b` (5 shots)	0.5343	0.4848	**0.4943**	**0.7447**	**0.3747**	**0.4985**
`qwen3-8b` (500)	0.0866	0.0531	0.0579	0.6976	0.0283	0.0544
`qwen3-14b`	0.4719	**0.4993**	0.4597	0.7283	0.3067	0.4316
`qwen3-14b` (5 shots)	**0.5360**	0.4625	0.4734	0.8143	0.2338	0.3633
`qwen3-14b` (500)	0.2488	0.2301	0.2278	0.5882	0.1049	0.1781
`qwen3-30b`	0.4381	0.4888	0.4302	0.6608	0.3012	0.4138
`qwen3-30b` (5 shots)	0.4949	0.4715	0.4605	0.6012	0.2838	0.3856
`qwen3-30b` (500)	0.1030	0.0598	0.0662	0.3944	0.0453	0.0812

The results indicate that, as expected, the strongest GPT-5 model achieves the highest F-scores, but open-source models perform competitively and do not exhibit a substantial performance gap. In low-resource settings, a natural strategy is to leverage a portion of the dataset for few-shot learning. As shown in Table 1, this approach yields the highest overall scores among the evaluated methods. Interestingly, five randomly selected shots from the gold data are more effective for the 8B and 14B models than for the 30B model.

We also evaluated a BERT-based retrieval approach. However, because this method operates at token-level boundaries, it struggled to handle citations involving ranges or explicit exceptions, which are common in regulatory text. As a result, this approach underperformed on the cross-reference resolution task despite its frequent use in related work – and its description and results have been excluded from this paper.

6 Two Solutions For Data Scarcity Problem

As rule-based model was reported to work well on annotated Japanese National Pension dataset [17], the authors have tried to obtain the corpus and the code, but the creators have not responded (most of the e-mail addresses were discontinued, as the data has been created over a decade ago). As LegalBERT showed

promising results for English, we also wanted to experiment with a Japanese version [6] but the model was not open for public. As the scarcity of data problem has not been solved, we decided to create a rule-based program and to generate synthetic data to allow fine-tuning.

6.1 Rule-based Citation Extractor

To establish a manually-crafted performance baseline, we developed a rule-based extraction program designed to identify hierarchical Japanese regulation references. The system processes natural language text to resolve relative references, ranges, and exclusions by validating candidates against a knowledge base of valid regulation IDs extracted from e-Gov XML file. The extraction logic is formalized in Algorithm 1, presented in the Appendix.

The system constructs full keys by prepending the regulation title (e.g., Ministerial Ordinance on Export Goods and Technologies, Article 1, Paragraph 1, Item 8, b) to the resolved hierarchy. The "b" letter in the example does not exist in the data, the sub-items on this level are represented by Japanese katakana characters order in traditional *iroha* system.

We evaluate this extractor on the whole gold dataset (134 entries). The system achieved precision of 0.7821, a recall of 0.3097, and an F1 score of 0.4437. The lower recall is primarily attributed to non-standard citation formats and references to external tables not present in the knowledge base. While precision remains relatively high, over-extraction in complex range cases occasionally introduces false positives. It became clear that to reach higher coverage achieved by Tran et al. [17], one would need more sophisticated set of rules if the manual approach is chosen.

6.2 Reference Data Synthesis

Although fine-tuning has proven effective e.g. for English court judgments, the scarcity of suitable training data necessitated the generation of synthetic data. In order to generate a datasets resembling export control regulations, we employed a neuro-symbolic verification loop to bridge the gap between probabilistic language modeling and deterministic legal logic. The framework treats the Large Language Model (`gpt-oss-120b`) as a neural proposal engine that generates diverse legal scenarios and candidate cross-references. Because LLMs are prone to structural hallucinations, these outputs are treated as unverified hypotheses. Grounding is provided by a symbolic extractor, a rule-based system that parses the generated text against a formal regulatory ontology. This component identifies all mentioned articles, paragraphs, and items, mapping them to a normalized statutory database. A verification gate then compares the LLM's self-reported labels with the extractor's findings. If a logical discrepancy is detected, such as a reference to a non-existent article, the sample is rejected and the LLM is prompted to self-correct. This algorithm is presented in Algorithm 2 (see Appendix). As our test dataset (100 random samples) contains only 11 cases where empty list is the output (correctness needs to be verified by an expert), we

additionally asked `gpt-oss-120b` to generate 150 negative examples related to export control, but do not mention articles or paragraphs (if article-related kanji characters are found in the generated text, it was automatically excluded).

7 Fine-tuning on Synthetic Data

7.1 Fine-tuning Details

To see if the synthetic training corpus can help models learn how to process citations, we fine-tune two Qwen (30B- and 72B-parameter instruction-tuned) models using Low-Rank Adaptation to specialize high-level reasoning for legal reference resolution while preserving general linguistic competence. LoRA adapters are inserted into all linear projection modules of the top 24 transformer layers, leaving lower layers frozen. Training is performed in the MLX framework optimized for Apple Silicon hardware, with gradient checkpointing enabled to control memory usage and a long context window to support dense legal structures. Inputs are formatted in ChatML with a fixed system instruction requiring exhaustive identification and canonical resolution of all legal cross-references. The hyperparameters used for fine-tuning are as follows:

- LoRA rank $r = 16$ and scaling factor $\alpha = 32$
- dropout probability 0.05
- trainable layers 56–80
- batch size 4
- learning rate 1.0×10^{-5}
- training iterations 1,000
- maximum sequence length 8,192

This configuration concentrates adaptation capacity on abstract semantic composition and symbolic grounding, hoping to enable extraction of complex legal references from Japanese regulatory text.

7.2 Results of Fine-tuning

The results suggest that generated data has not been able to improve the performance. The fine-tuning on smaller model, Qwen-7B, yielded higher results (P=0.425, R=0.348, F1=0.383) than its bigger version, Qwen-32B (P=0.585, R=0.257, F1=0.357). The results show that larger model can be weaker in fine-tuning approach, at least in our task of retrieving article names from export-control regulations.

8 Retrieval-Augmented Generation Framework

Although costly with bigger documents, one popular methods in legal NLP is retrieval-Augmented Generation (RAG). As the next experiment, we build such architecture to ground the reasoning of Large Language Models (LLMs) within the specific constraints of Japanese export control regulations. This framework

shifts the model's operation from a closed-book reasoning task to an evidence-based extraction process by leveraging a structured knowledge base designated as the "Consolidated Matrix". The retrieval phase identifies relevant statutory fragments, such as specific articles and paragraphs, which are then used to populate the model's context window to decrease the costs of feeding a model the whole document. This augmentation allows the LLM to resolve implicit references that are otherwise ambiguous when viewed in isolation.

8.1 The Pipeline

Our implementation divides the knowledge processing task into a high-dimensional vector search component and a generative model. To ground the reasoning within the specific constraints of Japanese export control regulations, we decomposed the "Consolidated Matrix" into granular, semantically distinct chunks. These fragments were encoded into a 1,536-dimensional latent vector space using the `text-embedding-3-small` model.

When a query involving legal text is provided, the system projects it into the same embedding space and performs a k-nearest neighbor search with $k = 5$ to isolate the most relevant legal articles. This step effectively mitigates the inherent "context window" limitations of large language models by filtering the extensive regulatory database into a manageable set of high-probability context windows. Once the relevant statutory context is retrieved, it is concatenated with the original query to form a Prompt-Context Augmented Input. To evaluate the impact of parameter scale on legal reasoning, this composite input was fed into three local models: `qwen3-14b`, `gpt-oss-20b`, and `gpt-oss-120b`.

8.2 RAG Results

Across all evaluated architectures, the RAG pipeline failed to obtain meaningful results for the cross-reference resolution task. The most capable model, `gpt-oss-120b`, achieved a Micro-average F1 of only 0.0133 and a Macro-average F1 of 0.0376. Experimental observations revealed several critical performance bottlenecks inherent to this approach.

First, while the integration of regulatory context is intended to improve factual grounding, it introduces a non-trivial computational overhead that scales with the volume of retrieved data. Our tests showed that expanding the evaluation to one hundred samples resulted in an execution time exceeding sixty minutes, representing a linear bottleneck in long-context processing for legal datasets. Second, the system encountered a "resolution gap" where semantic embeddings struggled to distinguish between logically distinct but linguistically similar article titles. This emphasizes that semantic proximity in a vector space does not necessarily correlate with the precise symbolic requirements of legal information retrieval, highlighting the necessity of hyperparameter tuning and deterministic grounding in legal RAG systems.

9 Discussion

From the results of all examined methods, we can draw a conclusion that the manually crafted dataset is a hard testbed for basic approaches. In many cases precision or recall were distinctively different, showing the inconsistency of all methods. In the specific domain of export control, the "importance" of precision vs. recall depends on the user's goal. In the case of recall (thoroughness), missing a single citation (a false negative) could mean overlooking a critical regulation, potentially leading to illegal exports and heavy fines. In this view, high recall is the priority to ensure "legal safety". In the case for precision (efficiency), if the tool provides too many "extra" irrelevant citations (false positives), the human expert wastes hours manually verifying non-existent links. If the tool's precision is too low, the expert will stop trusting it entirely.

Rather small differences between models' F1 scores shows the difficulty of the task. Often big parameter size of the model does not help, however the maximal number of token has to be carefully chosen[10]. Relatively high performance of rule-based method suggests that these methods could become white-box tools in the toolboxes of black-box Legal NLP agents.

As researchers recently showed [15], even strong models can have problems with reasoning and our dataset can be an example of a hard testbed as it represents a rare task in a rare domain and is written in Japanese language.

10 Conclusion and Future Work

In this paper, we introduce a manually curated dataset derived from Japanese export control regulations, designed to evaluate the ability of NLP methods to identify and resolve all article- and item-level cross-references appearing in regulatory texts. We conduct a series of experiments using a wide range of approaches, including rule-based systems, large language models, parameter-efficient fine-tuning, and retrieval-augmented generation. Although `gpt-5` yields the highest scores, 4-shots even on small `qwen3-8b` model can achieve comparable performance. The results suggest that our dataset can be a challenging benchmark for the Legal Japanese cross-reference resolution task to be tackled by novel methods in the future.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 23K11757 and JST CREST Grant Number JPMJCR20D2.

[10] During error analysis we discovered that there were several cases where 8,192 limit has been crossed, possibly leading to missing output candidates.

References

1. Adedjouma, M., Sabetzadeh, M., Briand, L.C.: Automated detection and resolution of legal cross references: Approach and a study of luxembourg's legislation. In: 2014 IEEE 22nd International Requirements Engineering Conference (RE). pp. 63–72. IEEE (2014)
2. Beyer, D.: Relational programming with crocopat. In: Proceedings of the 28th international conference on Software engineering. pp. 807–810 (2006)
3. Blair-Stanek, A., Holzenberger, N., Van Durme, B.: BLT: Can large language models handle basic legal text? In: Aletras, N., Chalkidis, I., Barrett, L., Goanță, C., Preoțiuc-Pietro, D., Spanakis, G. (eds.) Proceedings of the Natural Legal Language Processing Workshop 2024. pp. 216–232. Association for Computational Linguistics, Miami, FL, USA (Nov 2024). https://doi.org/10.18653/v1/2024.nllp-1.18
4. Cunningham, H.: Gate, a general architecture for text engineering. Computers and the Humanities **36**(2), 223–254 (2002)
5. Gheewala, A., Turner, C., de Maistre, J.R.: Automatic extraction of legal citations using natural language processing. In: WEBIST. vol. 1, pp. 202–209 (2019)
6. Miyazaki, K., Sugawara, Y., Yamada, H., Tokunaga, T.: Construction of bert specialized for japanese legal domain documents (in japanese). In: Proceedings of the 28th Annual Meeting of the Association for Natural Language Processing. pp. 1546–1551 (March 2022)
7. Obayashi, A., Rzepka, R.: Towards interactive advisory system for security export control. In: Proceedings of IJCAI Workshop on Language Sense on Computer, Macau (2019)
8. Obayashi, A., Rzepka, R.: Expanding export control-related data for expert system. In: Proceedings of 26th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Verona, Italy (2022)
9. Rzepka, R., Muraji, S., Obayashi, A.: Expert evaluation of export control-related question answering capabilities of LLMs. In: 2023 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE). pp. 1–6. IEEE Computer Society, Los Alamitos, CA, USA (2023). https://doi.org/10.1109/CSDE59766.2023.10487735
10. Rzepka, R., Muraji, S., Obayashi, A.: Evaluating lightweight embedding guardrails for cost-effective misalignment mitigation in export control dialog systems. In: to appear in: Proceedings of the 10th Linguistic and Cognitive Approaches to Dialog Agents (LaCATODA 2026), AAAI 2026 Workshop, Singapore (2026)
11. Rzepka, R., Obayashi, A.: Effectiveness of security export control ontology for predicting answer type and regulation categories. In: Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence. pp. 156–161. ICAAI '24, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3704137.3704180, https://doi.org/10.1145/3704137.3704180
12. Rzepka, R., Shirafuji, D., Obayashi, A.: Limits and challenges of embedding-based question answering in export control expert system. Procedia Comput. Sci. **192**(C), 2709–2719 (Jan 2021). https://doi.org/10.1016/j.procs.2021.09.041
13. Sannier, N., Adedjouma, M., Sabetzadeh, M., Briand, L.: An automated framework for detection and resolution of cross references in legal texts. Requirements Engineering **22**(2), 215–237 (2017)
14. Sargeant, H., Östling, A., Magnusson, M.: Detecting legal citations in United Kingdom court judgments. In: Christodoulopoulos, C., Chakraborty, T., Rose, C., Peng,

V. (eds.) Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 26810–26836. Association for Computational Linguistics, Suzhou, China (Nov 2025). https://doi.org/10.18653/v1/2025.emnlp-main.1361, https://aclanthology.org/2025.emnlp-main.1361/
15. Song, P., Han, P., Goodman, N.: Large language model reasoning failures (2026), https://arxiv.org/abs/2602.06176
16. Thuy, N.T.T., Diep, N.N., Bach, N.X., Phuong, T.M.: Joint reference and relation extraction from legal documents with enhanced decoder input. Cybernetics and Information Technologies **23**(2), 72–86 (2023)
17. Tran, O.T., Ngo, B.X., Le Nguyen, M., Shimazu, A.: Reference resolution in japanese legal texts at passage levels. In: Knowledge and Systems Engineering: Proceedings of the Fifth International Conference KSE 2013, Volume 2. pp. 237–249. Springer (2014)
18. Tran, O.T., Ngo, B.X., Nguyen, M.L., Shimazu, A.: Automated reference resolution in legal texts. Artificial intelligence and law **22**(1), 29–60 (2014)

```
prompt = f"""
あなたは日本の輸出管理令（貨物等省令）の専門家です。
現在位置: {loc}

【タスク】
テキストに含まれる「号(Item)」や「条(Article)」を抽出し、**カンマ区切りのリスト**で出力してください。

【厳格なルール】
1. 説明や挨拶は不要です。CSVのみ出力してください。
2. 省略された親番号を補完してください（「第一号イ及び口」→「第一号イ，第一号口」）。
3. 範囲を展開してください（「一から三」→「第一号，第二号，第三号」）。
4. 該当なしの場合は「None」と出力してください。

【テキスト】
{text}
"""

prompt = f"""
You are an expert on Japan's Export Control Order (Ministerial Ordinance on Goods, etc.).
Current location: {loc}

[Task]
Extract all referenced Items and Articles appearing in the text, and output them as a comma-separated list.

[Strict Rules]
1. Do not include explanations or greetings. Output CSV only.
2. Restore omitted parent numbers (e.g., "Item 1 (i) and (ro)" → "Item 1 (i), Item 1 (ro)").
3. Expand ranges (e.g., "one to three" → "Item 1, Item 2, Item 3").
4. If no applicable references are found, output "None".

[Text]
{text}
"""
```

Fig. 2: Prompt used for LLM-based approaches (with English translation below).

Algorithm 1 Rule-based Reference Extraction

Require: Natural language text T, Knowledge Base K
Ensure: Set of expanded regulation IDs S_{final}
$S_{range}, S_{in}, S_{black} \leftarrow \emptyset, \emptyset, \emptyset$ ("black" meaning "omission")
$L \leftarrow$ Tokenize T (regex for $j\bar{o}$ (Article), g$\bar{o}$ (Item), etc.)
for each token $t \in L$ **do**
  Update $context\_article, context\_item$ based on t
  $ref \leftarrow$ Resolve t (e.g., $d\bar{o}g\bar{o}$ (the same item) $context\_item$)
  $window \leftarrow$ Text segment following t
  if $window$ contains *-wo nozoku* (except of) **then**
    Add ref to S_{black}
  else if $window$ contains *kara* (from) ... *made* (to) **then**
    Add range $[start, end]$ to S_{range}
  else
    Add ref to S_{in}
  end if
end for
$S_{expanded} \leftarrow$ Expand $(S_{range} \cup S_{in})$ using child nodes in K
$S_{final} \leftarrow (S_{expanded} \setminus S_{black}) \cup S_{in}$
return S_{final}

Algorithm 2 Neuro-Symbolic Verification Loop for Legal Data Generation

Require: N (Target samples), $\mathcal{S}$ (Symbolic Address Database), G (Generator), J (Judge)
Ensure: $\mathcal{D}$ (gold dataset with expanded article and paragraph names)
$\mathcal{D} \leftarrow \emptyset$
while $|\mathcal{D}| < N$ **do**
  $L \leftarrow$ GenerateRandomLogic()
  $S_{gold} \leftarrow$ SymbolicExpand$(L, \mathcal{S})$
  if $S_{gold} = \emptyset$ **then**
    continue
  end if
  $T \leftarrow G$.generate$(L, \text{temp} = 0.3)$ {generating}
  $L' \leftarrow J$.parse$(T, \text{temp} = 0.0)$ {judging}
  if L' is invalid JSON **then**
    $L' \leftarrow$ ast.literal_eval(L') {recovering}
  end if
  $S_{pred} \leftarrow$ SymbolicExpand$(L', \mathcal{S})$
  if $S_{gold} = S_{pred}$ **then**
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(T, L)\}$
    SaveToDisk$(\mathcal{D})$
  else
    LogMismatch(L, L', T)
  end if
end while
return $\mathcal{D}$

A Computational Framework to Uncover Gray Areas in Tax Legislation

Sofía Ocampo[1]✉[0009−0001−4871−1122], Carlos David Sánchez[1][0009−0004−4485−9834], Andrés Leguizamón[1][0009−0005−0040−3656], Una-May O'Reilly[2][0000−0001−6923−8445], and Erik Hemberg[2][0000−0001−8905−1186]

[1] National University of Colombia, Bogotá, Colombia
{socampoq, csanchezn, aleguizamonl}@unal.edu.co
[2] Massachusetts Institute of Technology, Cambridge, MA, USA
{unamay, hembergerik}@csail.mit.edu

Abstract. It is often said that every law has its loophole, meaning that legal rules frequently contain technical flaws, ambiguities, or unspecified details that can be legally exploited to bypass the intent of the law. In the context of taxation, such gray areas contribute to regressivity, unfairness, and substantial revenue losses, costing governments and companies billions of dollars worldwide. Reducing tax disputes increasingly requires more efficient and cost-effective strategies than traditional ex post resolution. In this context, natural language processing models offer a scalable and adaptable opportunity to anticipate gray areas directly from legal texts. However, systematic approaches that leverage these tools to identify potential sources of tax norm abuse at scale remain limited. This paper asks whether language-based computational methods can contribute to the systematic identification of gray areas in tax legislation. To address this question, we propose a methodology to define, identify, and annotate legal provisions according to their potential gray-zone risk. We apply this methodology to Colombian Free Trade Zone (FTZ) legislation and construct, to our knowledge, the first systematic dataset of tax norms annotated for gray-area-related non-compliance risk. Using this dataset, we implement classification tasks based on TF–IDF representations, compare its performance with transformer-based models, and relate it to tax law complexity. The classification achieves around 80% F_1 scores at its best specification.

Keywords: Legal NLP · Computational Legal Analysis · Tax Law Analysis · Legal Text Classification

1 Introduction

A common saying states that every law has its loophole. Colombian Free Trade Zone legislation offers a clear illustration of how loopholes do not disappear with reform, but rather continue to manifest in different contexts over time. In 1958, Free Trade Zones were introduced as a policy instrument to stimulate investment

and employment through the production of manufactured goods in Colombia. Law 105 of 1958 granted tax benefits to firms established in delimited coastal areas, under the expectation that these firms would engage in productive activity [1]. The design of the laws, however, allowed domestic firms to locate in these zones, access preferential treatment, and import goods without carrying out any real production. Subsequent reforms attempted to close this gap. By the mid-1980s, legislation distinguished between commercial and industrial zones, granting tax benefits only to firms that "develop the industrialization process of a product".[1] Because the notion of industrialization remained undefined, firms carried out purely formal transformations, such as labeling or repackaging, to qualify for the benefit. More recent reforms followed the same pattern. Given that 75 percent of firms operating in Free Trade Zones produced for the domestic market [10], Law 2277 of 2022 replaced production requirements with the obligation to submit an export plan reviewed by a public official [2]—shifting legal compliance to individual assessment, which renders access to the benefit dependent on discretionary power. Each reform closed a visible loophole and opened a new one, rooted in legal ambiguity.

Ambiguity, incompleteness, discretionary powers, and differential treatments in law generate gaps where legal interpretation determines tax outcomes. This results in tax controversies that impose high costs on taxpayers and administrations [11]: In 2022, firms involved in transfer pricing disputes reported average expenditures of USD 56.3 million in assessments, USD 24.7 million in penalties and interest, and USD 21.3 million in legal fees [12]. In 2025, 825 surveyed businesses reported at least one recent tax dispute, and over 70 percent indicated an increase in tax audits in recent years as governments seek to close loopholes and curb avoidance [27]. Through these gaps, firms pay less tax, public revenue declines, and fiscal capacity erodes. Such disputes reflect a reactive enforcement paradigm that intervenes after exploitation occurs, rather than preventing it at the drafting stage.

Despite its relevance, the problem remains weakly systematized in the literature. Governments and tax authorities address gray areas on a case-by-case basis, often through audits, litigation, or ad hoc administrative guidance. Computational approaches that attempt to anticipate gray areas directly from statutory language remain in an early stage. While recent studies explore how language models interact with tax law, there is still no systematic framework that links linguistic features of legal texts to the risk of norm abuse at scale.

This paper addresses the following research questions: How can gray areas in tax legislation be defined in a way that is both legally meaningful and computationally tractable? Can linguistic features of statutory texts signal provisions that are prone to exploitative interpretation? To what extent can language-based models recover expert assessments of gray-area risk, and where do they fail? How stable are the predictions when the input is modified through synonym substitution and phrase reordering? What is the relation between tax law complexity and the gray area risk detection?

The paper contributes to the literature in three ways. First, it proposes an operational framework to identify gray-area indicators grounded in legal theory and expert practice. Second, it constructs the first systematic dataset of Colombian tax provisions annotated for gray-area-related risk, using Free Trade Zone legislation as a case study. Third, it evaluates the ability of standard text representations and legal language models to detect these indicators, and relates their performance to legal text complexity. The remainder of this paper is organized as follows: Section 2 provides the theoretical background on legal uncertainty. Section 3 reviews related work. Section 4 details the proposed framework, dataset construction, and empirical results. Section 5 discusses the implications and limitations of the study, and Section 6 presents the conclusions.

2 Background

Legal uncertainty is a structural and unavoidable feature of tax law rather than a drafting defect. While the principle of certainty of taxation functions as a normative ideal, aiming at clarity, predictability, and stability, it cannot be fully realized in practice. As argued by Demin [11], legal uncertainty operates along a continuum and may arise both negatively, through omissions or inconsistencies, and positively, as a regulatory technique that enables flexibility and case-specific judgment in complex economic environments. Core sources of legal uncertainty in tax law include:

Lack of accuracy and clarity: Uncertainty arises from imprecision in legal drafting, limited understandability and accessibility, and incomplete regulation, including gaps in statutory provisions. It also reflects tensions between abstract norms and concrete applications; e.g., *undefined taxable events*, *partial regulatory coverage*.

Instability and inconsistency: Tax law is characterized by chronic instability, inconsistency, and fragmentation; e.g., *multiple amendments* may affect the same tax provision.

Vague or evaluative terms and concepts: *Open-textured* legal concepts lack fixed meaning and require interpretation by taxpayers, administrators, and courts. Their content is shaped through individual decoding and subsequent legal practice; e.g., terms such as *reasonable*, *ordinary* and *necessary expenses*

Open-ended statutory lists: Enumerations that are explicitly non-exhaustive function similarly to vague concepts, allowing authorities and courts to extend their scope with additional elements; e.g., expressions such as *other economically equivalent transactions* and *any similar arrangement*.

Discretion: Discretion arises when the law authorizes officials to depart from general rules based on subjective assessments of specific circumstances; e.g., *approval subject to* the authority's assessment.

In this context, we define **tax avoidance strategies** as the deliberate exploitation of these structural uncertainties—such as ambiguities, incomplete regulations, or discretionary powers—to minimize tax liabilities while remaining

strictly within the formal boundaries of the law, distinguishing it from illegal tax evasion.

3 Related Work

Detecting exploitable ambiguities in legal texts requires understanding how language creates interpretive flexibility that enables norm abuse. We organize prior research into three areas: natural language processing approaches to ambiguity, computational methods targeting tax uncertainty, and alternative techniques using genetic algorithms.

3.1 NLP and Ambiguity Detection

The challenge of computationally disambiguating meaning in text has long been recognized as difficult. Navigli [22] shows that fine-grained *disambiguation*, understood as "the ability to identify the meaning of words in context in a computational manner", achieves only 70% accuracy, establishing fundamental limits for semantic tasks. Bhattacharya et al. [3] probe BERT and GPT-2 representations and find that *ambiguity* is poorly encoded, with performance barely exceeding baselines. They further show that middle layers outperform upper layers, suggesting information loss at higher abstraction levels. Wildenburg et al. [26] deals with *underspecification* and demonstrate that models recognize underspecification at rates above chance, yet fail to identify cases in which underspecified sentences allow open-ended continuations (e.g., "among others").

In legal and technical domains, the picture becomes more complex. Guitton et al. [17] develop human annotation protocols for *open-texture* in EU legal texts, achieving a Cohen's Kappa of $\kappa = 0.70$ after reconciliation, up from an initial baseline of 0.24. This reveals both the difficulty and tractability of annotating abstract legal concepts, even for experts. In a posterior work, they use gpt-3.5-turbo and LLaMa-2-70b-chat on a corpus annotated and obtain 0.84 and 0.67 f1 scores [16]. Ferrari and Esuli [14] use domain-specific word2vec models to identify ambiguous terms across Wikipedia corpora, achieving Kendall's τ up to 0.88 for two-domain comparisons, though non-expert annotators miss domain-specific nuances. Ezzini et al. [13] achieve approximately 80% precision detecting ambiguity in requirements engineering, with domain-specific corpora improving detection by approximately 33%. Kim et al. [20] develop task-specific embeddings for smart speaker command ambiguity, finding that they outperform off-the-shelf embeddings. Taken together, these studies show that domain-specific representations consistently outperform generic ones, that annotation of abstract properties remains challenging, and that distinguishing whether models capture genuine semantic properties or merely surface lexical patterns remains an open question.

3.2 Computational Approaches to Tax Uncertainty

Recent studies show that language models exhibit non-trivial capabilities in analyzing legal texts, while consistently diverging from expert judgment. This divergence suggests that such models are better suited to augment, rather than replace, expert-driven approaches. Blair-Stanek et al. [4] show that OpenAI's `o1` model can generate novel tax strategies. However, their subsequent evaluation [5] reveals that Spearman's ρ for whole-strategy grading ranges only between 0.30 and 0.48 when compared to expert grades, far below the 0.89 achieved for simpler tasks. Guitton et al. [16] find that `gpt-3.5-turbo` detects *open-texture* in GDPR text with $F_1 = 0.84$, yet only 42% of the terms flagged by humans appear in model output, and inter-annotator agreement among LLMs remains low ($\kappa = 0.01$–0.36). Fratrič et al. [15] combine large language models with Prolog-based planning to expose loopholes in multinational structures, though the formalization of legal rules by LLMs remains insufficient without human fine-tuning.

3.3 Alternative Computational Methods: Genetic Algorithms

Hemberg et al. [18] analyze a loophole created by a modification to Section 754 of the U.S. Tax Code and show that "each time the IRS changes the tax code, tax evaders respond by exploiting new ambiguities". They employ co-evolutionary genetic algorithms to model adversarial taxpayer–auditor dynamics, in which tax schemes and audit rules evolve simultaneously. Extending this anticipatory approach to Special Economic Zones, recent work [21] applies Genetic Algorithms to simulated supply chains, demonstrating how structural rate differentials enable complex, compliant-appearing profit-shifting strategies.

3.4 Synthesis and Identified Gaps

In summary, while prior literature has extensively explored the theoretical dimensions of legal uncertainty, treating ambiguity, open-texture, and tax uncertainty as broad, abstract phenomena, a significant gap remains in translating these concepts into verifiable computational tasks. Existing NLP approaches often struggle to capture domain-specific legal nuances, and recent LLM applications in tax law frequently lack grounded, expert-driven frameworks to formally evaluate structural loopholes. This paper bridges this gap by crystallizing these broad, nebulous theoretical concepts into a concrete, systematic, and empirically observable set of gray-area indicators. Our framework advances beyond theoretical discussions of ambiguity to provide a computationally tractable tool for the automated detection of tax norm abuse by operationalizing legal uncertainty into discrete categories that can be consistently identified by both human annotators and computational models.

4 Methodology and Results

4.1 Framework for Gray-Area Risk Classification and Dataset Creation

As mentioned in the background, legal uncertainty has been defined in different ways across fields. We propose an approach that operationalizes specific manifestations of legal uncertainty in ways that are computationally tractable and empirically observable. The translation method is a qualitative operationalization process combining discussions with tax law experts, reconstruction of historical avoidance schemes, and review of the legal uncertainty literature. From these inputs, we mapped abstract theoretical concepts into a set of discrete features that are tangible, precise, and amenable to explicit decision rules, which were then consolidated into the indicators reported in Table 1 and further detailed in the technical appendix (Section 7). Although the presence of an indicator does not constitute avoidance per se, it allows us to systematically identify risk-prone features embedded in tax legislation.

Building on the formal framework introduced above, we collapse the structural components of legal gray areas into a finite set of operational indicators. A more detailed explanation of how the indicator gives rise to the risk of tax avoidance, as well as examples of each category can be found in the technical appendix.

The corpus comprises all legal norms governing Colombia's Free Trade Zone (FTZ) regime from 1958 to 2005. This legislative domain was selected because it has been repeatedly identified as a focal point for tax avoidance strategies and exhibits frequent regulatory reforms that often close existing loopholes while simultaneously creating new ones [10]. Each article was manually annotated by a subject matter expert for the presence of gray area indicators following predefined annotation guidelines and decision rules. `Relevance`, `completeness`, `discretion`, and `differential regime` were treated as predicates (true for presence, false for absence). `Interpretation` was annotated using an ordinal scale reflecting increasing degrees of interpretive openness: low(1), moderate(2), and high(3). More details on the annotation protocol and the full database can be consulted through the technical appendix.

The expert-annotated labels for each indicator exhibit a clear but modest class imbalance, with a systematic overrepresentation of the label `false`. This imbalance motivates the use of macro-averaged evaluation metrics, as they weight all classes equally and prevent overall performance measures from being dominated by the majority class. Figure 1 reports the presence of indicators in training dataset articles.

4.2 Embedding-Based Analysis

To evaluate whether gray-area triggers form distinct patterns beyond surface-level text, we map the provisions into a semantic space—a mathematical representation where texts with similar legal meanings are clustered together. We generate these dense vector representations using MEL [24], a benchmark transformer model

Indicator	Answers	Question
Completeness	Yes/No	Does the article contain a fragment that specifies legal or tax treatment for some cases while omitting other relevant situations?
Differential Regime	Yes/No	Does the article contain a fragment that assigns different tax treatments within the same system based on specific attributes or classifications?
Discretion	Yes/No	Does the article contain a fragment where the application or enforcement of tax rules depends on administrative judgment rather than explicit legal criteria?
Interpretation	Low/Medium/High	Does the article contain a fragment whose wording admits multiple plausible interpretations leading to distinct legal or tax outcomes?
Relevance	Yes/No	Does the article contain a fragment that directly affects tax liabilities or economic operations relevant for tax calculation?

Table 1: Annotation indicators and labeling scheme

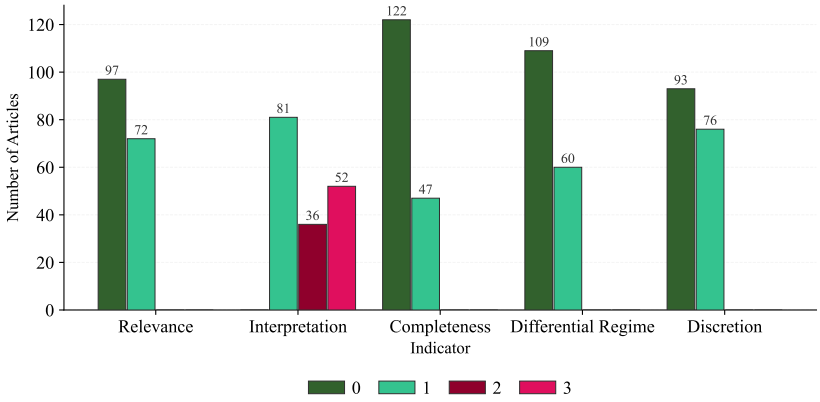


Fig. 1: Presence of indicators in training dataset articles

trained on Spanish legal texts to capture semantic nuances. Unlike bag-of-words approaches, these embeddings are designed to capture deep semantic properties. We then apply UMAP, a dimensionality reduction technique, to project these high-dimensional vectors into a 2D plane for visual inspection. Figure 2 presents these UMAP projections. `Interpretation` and `discretion` show no clustering, suggesting heterogeneous definitions or expert criteria not encoded in surface-level semantics. By contrast, `relevance`, `completeness`, and `differential regime` exhibit partial structure, indicating closer alignment between annotations and latent semantic space.

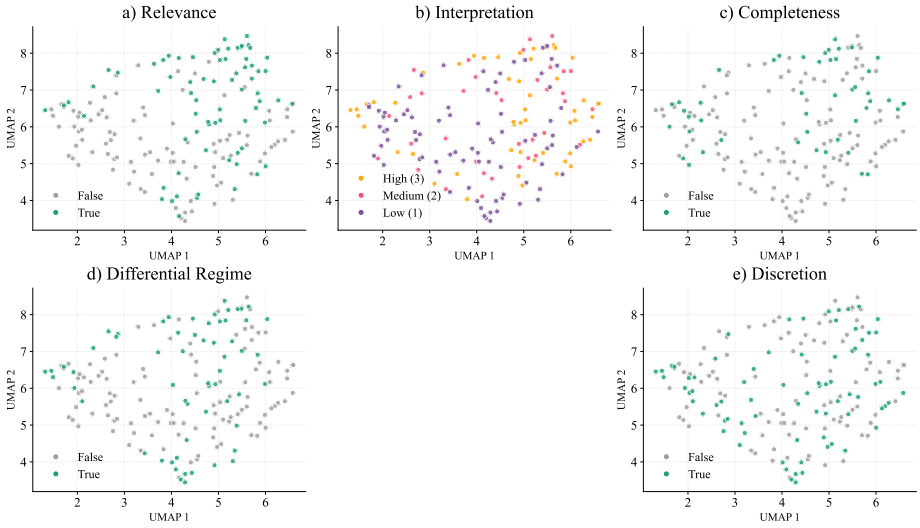
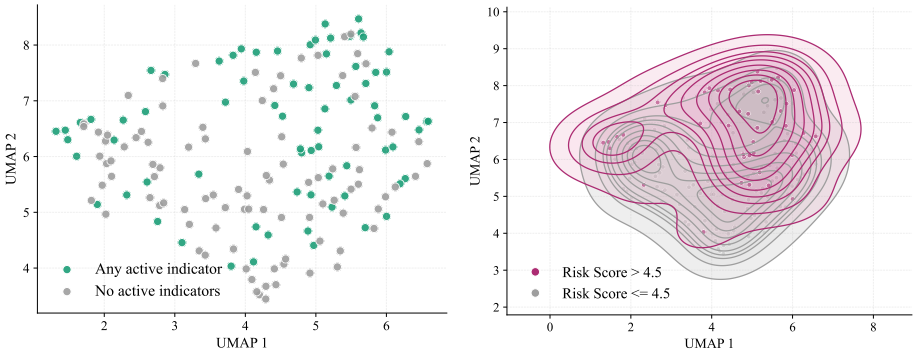


Fig. 2: Scatter plots of embeddings using MEL and UMAP.

We construct an aggregate risk score as a descriptive count of how many gray-area indicators are active in each article. Formally, for article i,

$$R_i = \sum_{k=1}^{K} \mathbb{1}_{\{p_{ik}=\text{true}\}} \tag{1}$$

where $K = 5$ is the total number of indicators, p_{ik} denotes the annotation assigned to indicator k in article i, and $\mathbf{1}\{\cdot\}$ is the indicator function. Thus, R_i ranges from 0 to 5 and can be interpreted simply as the number of active gray-area indicators in article i. Figure 3 compares articles with at least one active indicator and those with higher aggregated risk scores against low-risk articles in the embedding space. Both scatter and KDE plots show substantial overlap across groups, with no clear clustering or density separation for high-risk provisions.



(a) Scatter plot (b) UMAP KDE (MEL) - Risk Score > 4.5

Fig. 3: Global risk assessment of high-risk versus low-risk articles in the embedding space using MEL and UMAP.

4.3 Classification Task

We evaluate whether language models can learn expert-informed assessments of gray-area risk. To do so, we contrast models that capture surface lexical patterns (i.e., exact word frequencies and syntactic markers) with those designed to capture deep semantic properties (i.e., contextual legal meaning). We compare multiple text representation strategies, including TF-IDF as our lexical baseline, Legal-BERT on its light Spanish variant [7], and MEL [24]. Logistic regression is used as the classification model across all configurations, which provides an interpretable and widely adopted baseline in legal NLP applications [8,9]. The train/test proportions were 80 and 20% (135 and 34 articles respectively) Table 2 shows similar performance across models, with F1 scores around 0.7 for most binary indicators. TF–IDF remains competitive and achieves the best performance for `differential regime`, indicating that this indicator is largely captured by surface-level textual features. Domain-specific transformers provide limited gains: Legal-BERT improves `completeness` and `discretion`, while MEL performs best on `interpretation` and `relevance`. Overall, improvements from domain adaptation and Spanish-language training are modest and concentrated in semantically driven indicators, while most predictive signal appears to be lexical.

Table 2: Model performance comparison by indicator

	TF–IDF			Legal-BERT			MEL		
Indicator	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
`Completeness`	0.814	0.702	0.730	0.775	0.809	**0.788**	0.738	0.753	0.744
`Diff. Regime`	0.868	0.890	**0.875**	0.709	0.701	0.704	0.807	0.807	0.807
`Discretion`	0.766	0.754	0.757	0.793	0.788	**0.790**	0.674	0.675	0.674
`Interpretation`	0.262	0.343	0.272	0.331	0.362	0.333	0.455	0.457	**0.447**
`Relevance`	0.757	0.757	0.757	0.749	0.700	0.704	0.800	0.771	**0.779**

`Note:` Identical data were used for all three models ($n = 169$ in each of the 5 categories). All metrics are macro-averaged.

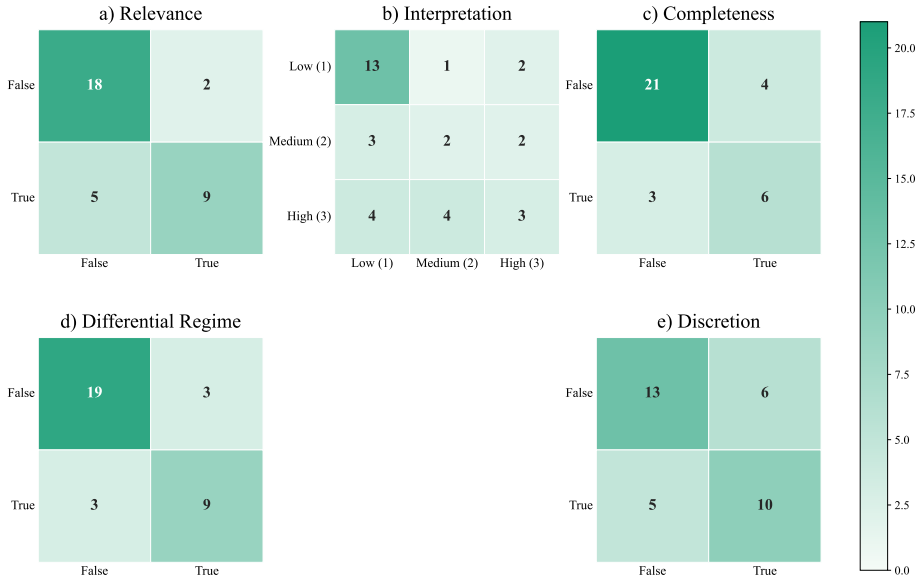


Fig. 4: Confusion matrices for MEL across the five indicators.

4.4 Robustness Analysis

Following [23] and [25], we assess robustness through controlled perturbations designed to separate sensitivity to surface form from sensitivity to core regulatory meaning. We generate two variants of each text. First, phrase reordering shuffles sentence-level units to alter syntactic organization while keeping most lexical material unchanged, testing whether predictions depend on structural arrangement. Second, lexical substitution replaces a fixed proportion of non-stopword tokens with Spanish WordNet synonyms, modifying surface lexical patterns while aiming to preserve the core meaning of the provision. In our annotation setting, these perturbations are treated as approximately label-preserving because indicators such as discretion or completeness depend on the regulatory mandate expressed

by the provision rather than on exact clause order or synonymous lexical choices. Perturbed texts are embedded using MEL, and predictions are obtained from the trained model. Across the 170 perturbed test instances (test size 34 for each of the 5 categories), predictions remain unchanged in all but four cases: two for `discretion` and two for `relevance`, all induced by phrase reordering rather than lexical substitution. This indicates that model decisions are largely invariant to lexical variation and only marginally sensitive to syntactic reorganization, suggesting limited reliance on surface-level cues and weak sensitivity to higher-order structural changes.

4.5 Legal Complexity and Predictive Performance

To examine whether gray-area risk is associated with the textual and structural burden of legal provisions, we construct an exploratory article-level complexity proxy. Building on broader discussions of tax complexity [19], our goal is not to measure readability in the narrow psycholinguistic sense, but to approximate three observable dimensions of legal complexity: textual length, syntactic packing, and intertextual dependency.

Let i denote a legal article. We define W_i as the total number of words, S_i as the number of sentences, $L_i = W_i/S_i$ as the average sentence length, and R_i as the number of explicit normative references. These components are intended to capture, respectively, the overall volume of textual material, the density with which conditions and qualifications are packed into sentences, and the extent to which interpretation depends on cross-references to other legal provisions.

Each component is normalized using min-max scaling:

$$\tilde{X}_i = \frac{X_i - \min(X)}{\max(X) - \min(X)}, \quad X \in \{W, L, R\}. \tag{2}$$

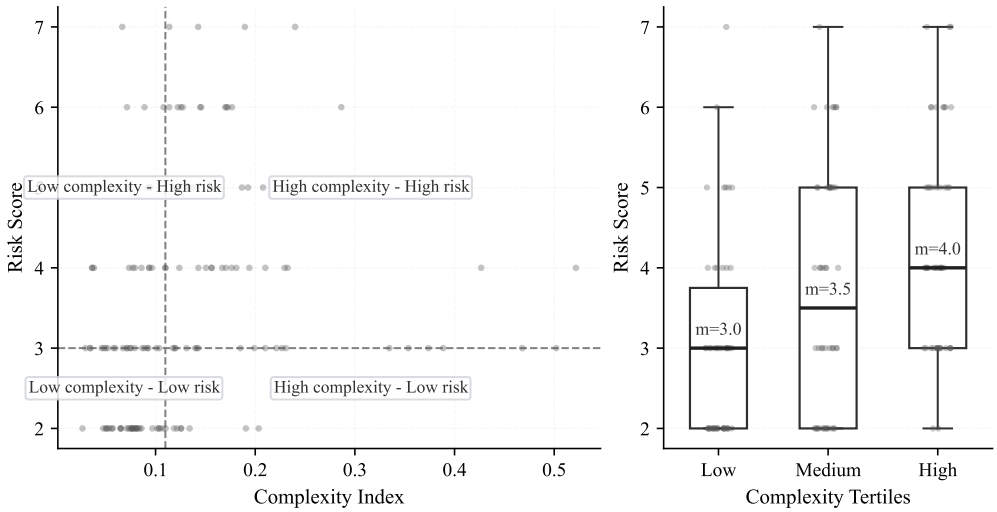
We then define the textual complexity index as the unweighted average:

$$\text{Complexity}_i = \frac{1}{3}\left(\tilde{W}_i + \tilde{L}_i + \tilde{R}_i\right). \tag{3}$$

In the absence of an established theoretical or empirical basis for assigning different weights to these dimensions, we use equal weights to avoid imposing arbitrary assumptions about their relative contribution to article-level legal uncertainty. While future work could explore component-level ablations to isolate their individual contributions, we interpret this linear combination as an exploratory proxy for the overall textual and interpretive burden imposed by the provision rather than a fully validated measure.

Figure 5a shows that articles with lower complexity levels tend to be associated with lower risk scores, while articles with higher complexity are more frequently linked to higher risk scores. The plot partitions the space using the midpoints of the complexity index and the risk score, yielding four quadrants that facilitate a descriptive comparison between low- and high-complexity provisions and their corresponding risk levels. Figure 5b presents the distribution of risk scores across

different levels of legal complexity. The results show that higher complexity levels are associated with higher median risk scores. In addition, the dispersion of the risk score increases with complexity, indicating greater variability in uncertainty among more complex legal provisions.



(a) Complexity index and risk score **(b)** Risk by complexity tertile

Fig. 5: Association between legal complexity and aggregate risk score.

Figure 6 shows the distribution of the complexity index across indicators. Notably, the indicators with weaker classification performance, namely `interpretation` and `discretion`, exhibit a wider dispersion in complexity values. In contrast, indicators with stronger classification results display comparatively lower variability in complexity, particularly for `relevance`, and to a lesser extent for `differential regime` and `completeness`.

5 Discussion

As noted above, this study examines gray-area triggers linked to expert-identified sources of tax norm abuse as a first step toward anticipating exploitative uses directly from statutory language. The analysis is intentionally restricted to uncertainty-based textual features and does not incorporate other mechanisms highlighted in prior work, such as interactions across legal provisions or enforcement dynamics [6].

The empirical results show clear differences across gray-area indicators. The strongest performance is observed for Differential Regime, Relevance, and Completeness. These indicators exhibit structured organization in the embedding space and consistent predictive behavior, indicating that they align with stable linguistic regularities present in statutory text. In contrast, Interpretation and

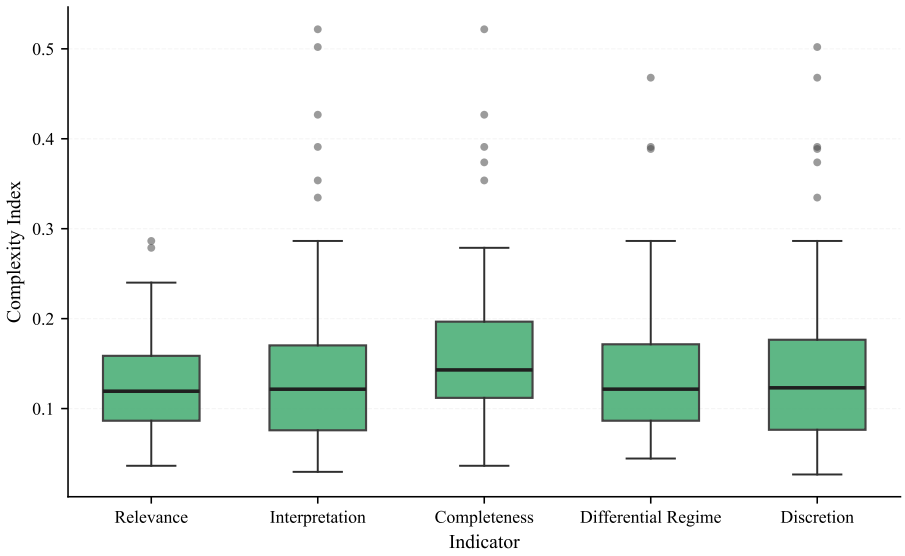


Fig. 6: Complexity index distribution by indicator

Discretion display weak predictive performance and lack discernible structure in vector representations. This pattern suggests that expert assessments for these latter indicators do not rely on textual properties encoded in the statutory language itself.

The absence of structure for Interpretation and Discretion indicates that the exploitability of these gray areas depends on contextual, institutional, or strategic factors external to the written norm. This finding supports prior literature arguing that uncertainty-based textual approaches should be complemented with methods that account for cross-provision interactions and enforcement signals in order to capture these dimensions of norm abuse.

At the aggregate level, the analysis also reveals limitations of the risk score construction. Since the overall risk score is defined as the simple sum of indicators-level indicators, the lack of separation in the embedding space suggests that aggregated risk is not reflected in global linguistic similarity. This result does not undermine the validity of expert annotations, but rather indicates that the criteria underlying risk assessment are not jointly encoded in the textual representations used in this study. We therefore conclude that a linear aggregation of indicators is not well suited to characterize overall gray-area risk, likely because the indicators do not exhibit linear independence and do not define a coherent group of provisions when combined.

6 Conclusion

A folk saying holds that every rule has its loophole. In tax law, the exploitation of such loopholes generates substantial economic costs for both taxpayers and tax

administrations. Addressing this challenge requires scalable approaches capable of anticipating the emergence of gray areas before they are systematically exploited.

This paper contributes to that objective in three ways. First, it proposes an operational framework to identify gray-area indicators grounded in legal theory and expert practice, translating specific manifestations of legal uncertainty into indicators that are computationally tractable and empirically observable. Second, building on this framework, it introduces the first systematic dataset of Colombian tax provisions annotated for gray-area risk, using Free Trade Zone legislation as a case study. The annotation reveals that the indicators Relevance, Completeness, and Differential Regime capture coherent and related groups of provisions, whereas Interpretation and Discretion do not exhibit such structure.

Third, using this corpus, the paper evaluates the ability of standard text representations and legal language models to detect gray-area indicators. The results show that transformer-based architectures, including MEL, do not substantially outperform bag-of-words representations, suggesting that the annotations are primarily grounded in lexical rather than semantic features. In the best-performing specifications, classification achieved F1 scores of 0.88 for Differential Regime, 0.78 for Completeness, 0.79 for Discretion, 0.77 for Relevance, and 0.44 for Interpretation. Additionally, legal text complexity is positively associated with the presence of gray-area indicators and negatively associated with classification performance. This finding underscores the need to treat textual complexity as an explicit dimension in future computational approaches, in order to avoid biased or confounded detection outcomes.

The findings show that models operating on legal text can anticipate gray-area triggers grounded in explicit linguistic structure, including differential treatment, scope delimitation, and normative completeness. In these cases, the statutory language itself provides sufficient signal for systematic identification, which provides a basis for future computational work on the anticipation and detection of tax avoidance strategies.

Acknowledgments. This study was supported by the MIT-Colombia Universidad Nacional de Colombia Seed Fund (MIT Global Seed Funds).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

7 Appendix

Due to space constraints, the complete formalization underlying the framework is provided in an external technical document, available at: technical-appendix.

References

1. Ley 109 de 1985: Por la cual se establece el estatuto de las zonas francas (Dec 1985), https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=77397

2. Ley 2277 de 2022: Por medio de la cual se adopta una reforma tributaria para la igualdad y la justicia social y se dictan otras disposiciones (Dec 2022), https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=199883
3. Bhattacharya, S., Zouhar, V., Bojar, O.: Sentence Ambiguity, Grammaticality and Complexity Probes (Oct 2022). https://doi.org/10.48550/arXiv.2210.06928, http://arxiv.org/abs/2210.06928, arXiv:2210.06928 [cs]
4. Blair-Stanek, A., Holzenberger, N., Durme, B.V.: Can LLMs Identify Tax Abuse? (Aug 2025). https://doi.org/10.48550/arXiv.2508.20097, http://arxiv.org/abs/2508.20097, arXiv:2508.20097 [q-fin]
5. Blair-Stanek, A., Holzenberger, N., Durme, B.V.: Can LLMs Identify Tax Abuse? (Jan 2026). https://doi.org/10.48550/arXiv.2508.20097, http://arxiv.org/abs/2508.20097, arXiv:2508.20097 [q-fin]
6. Blair-Stanek, A., Holzenberger, N., Van Durme, B.: Shelter Check: Proactively Finding Tax Minimization Strategies via AI (Dec 2022), https://papers.ssrn.com/abstract=4327110
7. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The Muppets straight out of Law School. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2898–2904. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261, https://aclanthology.org/2020.findings-emnlp.261/
8. Chhatwal, R., Huber-Fliflet, N., Keeling, R., Zhang, J., Zhao, H.: Empirical evaluations of preprocessing parameters' impact on predictive coding's effectiveness. In: 2016 IEEE International Conference on Big Data (Big Data). pp. 1394–1401 (Dec 2016). https://doi.org/10.1109/BigData.2016.7840747, https://ieeexplore.ieee.org/document/7840747
9. Chhatwal, R., Huber-Fliflet, N., Keeling, R., Zhang, J., Zhao, H.: Empirical evaluations of active learning strategies in legal document review. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 1428–1437 (Dec 2017). https://doi.org/10.1109/BigData.2017.8258076, https://ieeexplore.ieee.org/document/8258076
10. Commission, T.E., OECD, DIAN, Ministry of Finance, C.: Tax Expenditures Report by the Tax Experts Commission: Colombia 2021. Tech. rep., Dirección de Impuestos y Aduanas Nacionales (DIAN) and OECD and Ministry of Finance, Colombia (2021), https://www.dian.gov.co/dian/Documents/Tax-Expenditures-Report-By-Th-Tax-Experts-Commission.pdf
11. Demin, A.V.: Certainty and Uncertainty in Tax Law: Do Opposites Attract? Laws **9**(4), 30 (Dec 2020). https://doi.org/10.3390/laws9040030, https://www.mdpi.com/2075-471X/9/4/30
12. EY Global Tax and Transfer Pricing Team, Tracee J. Fultz, Joel Cooper, Jay Camillo: 2024 EY International Tax and Transfer Pricing Survey. Tech. rep., Ernst & Young (EY) (2024), https://www.ey.com/en_gl/insights/tax/international-tax-and-transfer-pricing-survey
13. Ezzini, S., Abualhaija, S., Arora, C., Sabetzadeh, M., Briand, L.C.: Using Domain-Specific Corpora for Improved Handling of Ambiguity in Requirements. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). pp. 1485–1497 (May 2021). https://doi.org/10.1109/ICSE43902.2021.00133, https://ieeexplore.ieee.org/abstract/document/9402055
14. Ferrari, A., Esuli, A.: An NLP approach for cross-domain ambiguity detection in requirements engineering. Automated Software Engineering **26**(3), 559–598 (Sep 2019). https://doi.org/10.1007/s10515-019-00261-7, https://doi.org/10.1007/s10515-019-00261-7

15. Fratrič, P., Holzenberger, N., Amariles, D.R.: Can AI expose tax loopholes? Towards a new generation of legal policy assistants (Mar 2025). https://doi.org/10.48550/arXiv.2503.17339, http://arxiv.org/abs/2503.17339, arXiv:2503.17339 [cs]
16. Guitton, C., Gubelmann, R., Karray, G., Mayer, S., Tamò-Larrieux, A.: Identifying open-texture in regulations using LLMs. Artificial Intelligence and Law (May 2025). https://doi.org/10.1007/s10506-025-09450-0, https://doi.org/10.1007/s10506-025-09450-0
17. Guitton, C., Tamò-Larrieux, A., Mayer, S., van Dijck, G.: The challenge of open-texture in law. Artificial Intelligence and Law **33**(2), 405–435 (Jun 2025). https://doi.org/10.1007/s10506-024-09390-1, https://doi.org/10.1007/s10506-024-09390-1
18. Hemberg, E., Rosen, J., Warner, G., Wijesinghe, S., O'Reilly, U.M.: Detecting tax evasion: a co-evolutionary approach. Springer Netherlands (Apr 2016), https://dspace.mit.edu/handle/1721.1/105846, accepted: 2016-12-15T22:58:41Z
19. Hoppe, T., Schanz, D., Sturm, S., Sureth-Sloane, C.: The Tax Complexity Index – A Survey-Based Country Measure of Tax Code and Framework Complexity. European Accounting Review **32**(2), 239–273 (Mar 2023). https://doi.org/10.1080/09638180.2021.1951316, https://www.tandfonline.com/doi/full/10.1080/09638180.2021.1951316
20. Kim, J.M., Lee, Y.j., Jung, S., Choi, H.j.: Semantic Ambiguity Detection in Sentence Classification using Task-Specific Embeddings. In: Sitaram, S., Beigman Klebanov, B., Williams, J.D. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track). pp. 425–437. Association for Computational Linguistics, Toronto, Canada (Jul 2023). https://doi.org/10.18653/v1/2023.acl-industry.41, https://aclanthology.org/2023.acl-industry.41/
21. Leguizamon, A., Sanchez, C.D., Ocampo, S., O'Reilly, U.M., Hemberg, E.: Evolutionary Computation for Tax-Minimizing Strategies in Special Economic Zones. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '26). Association for Computing Machinery, New York, NY, USA (2026)
22. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. **41**(2), 10:1–10:69 (Feb 2009). https://doi.org/10.1145/1459352.1459355, https://dl.acm.org/doi/10.1145/1459352.1459355
23. Ribeiro, M.T., Wu, T., Guestrin, C., Singh, S.: Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 4902–4912. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.442, https://aclanthology.org/2020.acl-main.442/
24. Sánchez, D.B., García, N.A., Jiménez, B., Nieto, M.G., Morales, P.M., Salas, N.S., Hernán, C.G., Coll, P.H., Ponsoda, E.M., Ibáñez, P.C.: MEL: Legal Spanish Language Model (Jan 2025). https://doi.org/10.48550/arXiv.2501.16011, http://arxiv.org/abs/2501.16011, arXiv:2501.16011 [cs]
25. Wei, J., Zou, K.: EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks (Aug 2019). https://doi.org/10.48550/arXiv.1901.11196, http://arxiv.org/abs/1901.11196, arXiv:1901.11196 [cs]
26. Wildenburg, F., Hanna, M., Pezzelle, S.: Do Pre-Trained Language Models Detect and Understand Semantic Underspecification? Ask the DUST! (Jun 2024). https://doi.org/10.48550/arXiv.2402.12486, http://arxiv.org/abs/2402.12486, arXiv:2402.12486 [cs]
27. Yates, G.: Business sees surge in tax disputes around the world and inquiries to increase (Feb 2025), https://fairtaxmark.net/business-sees-surge-in-tax-disputes-around-the-world-and-inquiries-to-increase/

Hierarchical Institutions for Contract Invalidity

Huimin Dong[1], Réka Markovich[2], Leendert van der Torre[2,3], and Liuwen Yu[4]

[1] TU Wien, Vienna, Austria
[2] University of Luxembourg, Esch-sur-Alzette, Luxembourg
[3] Zhejiang University, Hangzhou, China
[4] Luxembourg Institute of Science and Technology, Luxembourg

Abstract. Contracts can be modelled from a rights-first perspective as bundles of legal rights, which naturally supports contrary-to-duty (CTD) reasoning: when primary rights are not fulfilled, secondary rights and remedies arise. CTD captures the normative dynamics *within* a contract. In legal practice, however, admissible remedies also depend on an institutional determination *about* the contract — whether it is *valid*, *void*, or *voidable*. In this paper, we introduce a two-level architecture for contract governance, called *hierarchical institutions*, with two coupled state machines: an object-level contract-instance machine for drafting, acceptance, performance, revision, and escalation; and a meta-level institutional machine that, once a dispute is escalated, classifies the contract, may determine avoidance for voidable contracts, and dispatches an admissible repair family. This separation clarifies that the same dispute can be routed to enforcement or compensation when the contract is treated as valid, and to restoration such as restitution or unwinding when the contract is void or avoided. We sketch an agentic reasoning model in which agents update beliefs and intentions under institutional rules to anticipate these outcomes. Two scenarios inspired by Hungarian private law illustrate the framework: usury leading to voidness and gross disparity leading to voidability with avoidance.

1 Introduction

Contracts, understood as bundles of rights, provide a natural point of interaction between agents and institutions. Agents draft and negotiate the rights embodied in a contract, while institutions give legal effect to those rights. Between these points, a contract evolves as a structured artefact that can be performed, violated, repaired, or even deprived of legal effect. In AI&Law, this dynamic is often studied through *contrary-to-duty* (CTD) reasoning: if a primary right in the contract is not fulfilled, which secondary rights become operative, who may trigger them, and how the system returns to a stable normative state.

CTD is well-suited for rights-first reasoning *within* an effective contract [8]. It captures how contractual rights guide performance and how breaches trigger remedy structures. However, a focus on CTD alone can leave a crucial dimension underexplored: reasoning *about* the contract as an institutional procedure. In legal practice, parties do not merely anticipate what happens *if* a right is

violated; they also anticipate what happens *if* the agreement itself is challenged as producing no (or only defeasible) contractual legal effects. This institutional "shadow" influences drafting, negotiation, and escalation decisions, and is therefore central to agreement technologies and agentic systems operating under legal rights [15].

A contract may be *valid*, *void* (no contractual legal effect), or *voidable* (effective unless and until successfully challenged, typically with retroactive consequences). This kind of validity control is not just another CTD branch inside the contract. It is a meta-level institutional operation—performed by courts, regulators, or other legal authorities—that determines whether contractual effects are legally operative and enforceable [13]. When invalidity "blocks" a contract, it blocks contractual normative effects and enforceability, not the parties' ability to act in the world. Parties may already have transferred money, goods, or services; in such cases the legal system may route consequences through restoration mechanisms (e.g., restitution or unwinding) rather than through contractual enforcement or damages.

Institutions are structurally present throughout contractual interaction as background constraints: legal norms shape what agents should do and what they expect others to do. Institutional procedures typically become *procedurally active* when parties face an issue (e.g., a dispute or a validity challenge) and seek authoritative repair. In this paper we focus on this dispute-triggered pathway, while keeping the architecture compatible with additional institutional checkpoints.

Our motivating case study is institutional determination of validity outcomes that cannot be decided "inside" the contract. We illustrate this with two running examples framed by Hungarian private law: usury (formed but classified as void) and gross disparity (formed and classified as voidable; may become retroactively ineffective after successful avoidance).

The contrast can be summarised as follows:

- **CTD / violation reasoning** addresses what happens when agents do not fulfil what the contract requires (e.g., late delivery, non-payment). In many agreement technologies, the handling of such disruption is partly specified by the parties in the contract itself (e.g., damages clauses, revision mechanisms, escalation paths) and partly supplied by default legal remedies [15].
- **Voidness / voidability reasoning** addresses what happens when the legal system does not determine the legal effects of a contract (or part of it), or determines it as defeasible and later terminated retroactively. Here, the handling is primarily imposed by institutional doctrine: denying enforceability, enabling avoidance procedures, and imposing restoration duties such as restitution or unwinding.

This difference matters for agentic systems: reasoning only in CTD terms can treat every disruption as repairable within the contractual framework, while institutional classification (void/voidable) can deny enforceability from the outset or unwind the relationship after performance.

In this paper, our research questions are:

1. How do contractual violations differ from institutional invalidity outcomes and their consequences?
2. How can a legal institution determine an accepted but violated agreement as valid/void/voidable and route repair accordingly, including avoidance and restoration?
3. How can agents anticipate institutional classification and repair when deciding whether to accept, perform, revise, escalate, or comply under a contract?

Our approach is design-oriented and operational. Rather than proposing a new logic, we develop an explicit institutional model that separates what happens *within* a contract from what happens *about* the contract at the institutional level. We model this separation using a hierarchical transition system with two coupled layers: an object-level procedure capturing drafting, acceptance, performance, revision, and dispute escalation, and a meta-level procedure capturing institutional classification of validity, voidness, and voidability, including avoidance and retroactive effects. Within this setting, repair is treated as a central organising principle: both violation and invalidity are understood as perturbations that require restoration, but they differ in trigger, authority, and routing of admissible mechanisms. On the agent side, we provide a lightweight practical reasoning model that allows agents to anticipate institutional classifications and repair outcomes, based on belief and intention update under institutional rules.

The paper is structured as follows. Section 2 introduces the running examples. Section 3 presents the hierarchical institutional architecture, distinguishing object-level contractual operations from meta-level validity and avoidance control, and formalises the corresponding state machines and repair pathways. Section 4 develops the agentic reasoning model and illustrates how agents reason about violation, voidness, and institutional repair when deciding how to act under a contract. Section 5 discusses related work, and Section 6 concludes with directions for future research.

2 Case Study and Motivating Examples

Legal systems classify contracts that are *void* from contracts that are *voidable*, alongside the default case of *valid* contracts. Voidability is typically associated with an avoidance mechanism: an entitled party may attack enforceability through legally prescribed procedure (e.g., notice to the counterparty or court action); if avoidance succeeds, the contract becomes retroactively ineffective and restoration duties may follow (e.g., restitution/unwinding). We use this pattern in the Hungarian Civil Code (Act V of 2013, sections 6:88-6:98) to illustrate why validity control belongs *above* the contract. We use two scenarios to classify two distinct institutional determination patterns:

– **Usury:** an agreement is formed but institutionally determined as *void*.
– **Gross disparity:** an agreement is formed and institutionally determined as *voidable*, i.e., effective unless and until successful avoidance leads to retroactive voidness and restitution/unwinding.

Symbol	Informal meaning
$\mathsf{Void}(k)$	contract k is void (no contractual legal effect)
$\mathsf{Voidable}(k)$	contract k is voidable (effective but defeasible)
$\mathsf{Valid}(k)$	contract k is valid (binding and enforceable)

Table 1. Meta-level validity statuses used in the motivating examples.

Example U: Usurious loan offer (void) A lender and a borrower conclude an agreement k_U whose surrounding circumstances meet a legal ground for usury, represented as $\mathsf{Usurious}(k_U)$. Institutionally, this ground can support classification $\mathsf{Void}(k_U)$: the agreement is treated as producing no enforceable contractual effects, and the system should not proceed as if ordinary breach remedies (enforcement/compensation) were available under the contract. At the agent level, the parties may anticipate this risk already during negotiation and treat it as a reason to reject, revise, or avoid escalation.

Example D: Sale with gross disparity (voidable + avoidance) A seller and buyer conclude an agreement k_D whose circumstances meet a legal ground for gross disparity without gift intention, represented as $\mathsf{GrossDisparity}(k_D)$ and $\mathsf{NotGratuitous}(k_D)$. Institutionally, these grounds can support classification $\mathsf{Voidable}(k_D)$: the contract is effective but defeasible, because an entitled party may exercise an avoidance mechanism that, if successful, leads to retroactive ineffectiveness and triggers restitution/unwinding. At the agent level, this defeasibility affects acceptance and performance incentives, including whether parties attempt private revision or escalate to dispute.

These two examples motivate a hierarchical separation: *object*-level contractual operations (formation and performance) do not by themselves determine a contract's institutional standing; instead, validity outcomes (Valid, Voidable, Void) and their repair consequences must be controlled at a *meta*-level that can block enforcement, enable avoidance, and impose restoration.

3 Hierarchical Institutions

The institutional functions of *contract classification* and *contract repair* constitute the foundation of our architecture of *hierarchical institutions*. Unlike CTD reasoning *within* an effective contract, these functions are performed by legal institutions (courts, regulators, arbitral bodies) and therefore concern authoritative assessments *about* the contract as an institutional object.

Contract classification determines a contract instance's legal status as one of Valid, Voidable, or Void. Contract repair determines which *families* of legally admissible consequences can be ordered to resolve a dispute arising from nonfulfilment or contestation of an accepted contract. Based on these functions, we introduce a two-level architecture of contract governance, called a *hierarchical institution*, coupling two *state machines*: (1) an *object-level* contract-instance

machine modelling the parties' interaction, and (2) a *meta-level* institutional machine governing validity review, (optional) avoidance classification, and repair dispatch.

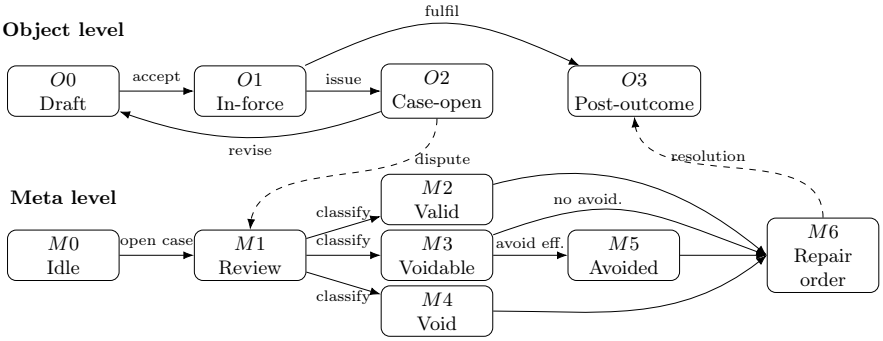


Fig. 1. Compact coupled governance machine: object-level execution and meta-level validity classification and repair control. Dashed arrows denote coupling (dispute escalation and resolution feedback).

3.1 A Two-Lane Hierarchy

Institutions are typically present as *background constraints* on contracting, but they become *procedurally active* once a dispute arises and a case is opened. The two layers of the hierarchy are shown in Figure 1. The object-level lane describes how a contract progresses through formation, performance, disruption handling, and post-outcome steps. The meta-level lane captures how a legal institution determines the contract's institutional status. Depending on that determination, the institution may deny enforceability (Void), treat the contract as defeasible and enable an avoidance pathway (Voidable), or confirm its enforceability (Valid). The coupling between lanes is represented through classificatory *counts-as* norms [9]: institutional determinations make certain facts and acts count as institutional statuses (e.g. Voidable) and admissible repair families (e.g. Restitution), thereby constraining which downstream actions are institutionally admissible for the parties.

We model each contract instance k as evolving in a two-lane transition system for the hierarchy, presented in Table 2. A run interleaves (i) object-level moves by the contractual parties and (ii) meta-level moves by the institution. To make the machine auditable, we distinguish (1) *fact transitions* (logged events such as non-fulfilment, evidence surfacing, and benefit transfers) from (2) *act transitions* (exercises of powers such as acceptance, dispute filing, avoidance, classification, and remedy orders). A governance configuration is a pair $\langle o, m \rangle$ where $o \in \{O0, O1, O2, O3\}$ is the current object-level state and $m \in \{M0, \ldots, M6\}$ is the current meta-level state.

Two transitions explicitly couple the lanes at the level of global configurations: escalation activates the institution, and the institutional order feeds back to object-level execution. Concretely, escalation moves the global configuration from $\langle O2, M0\rangle$ to $\langle O2, M1\rangle$ (opening institutional review), while the dispatched order moves the configuration from $\langle O2, M6\rangle$ to $\langle O3, M0\rangle$ (execution/closure).

3.2 Counts-as Coupling and Repair Routing

The machine becomes operational once we specify how *grounds* count as institutional statuses, and how these statuses constrain admissible repair families. In institutional review ($M1$), the institution establishes *grounds*: legally relevant conditions supported by evidence and doctrine (e.g. $\mathsf{Usurious}(k)$, $\mathsf{GrossDisparity}(k)$, $\mathsf{NotGratuitous}(k)$). These grounds serve as inputs to classificatory counts-as norms producing a validity status.

Using the motivating examples in Section 2, three key counts-as rules for classification are:

$$\mathsf{Formed}(k) \Rightarrow_{\mathsf{ca}} \mathsf{Valid}(k) \qquad (C_{\mathsf{valid}})$$

$$\mathsf{Formed}(k) \wedge \mathsf{Usurious}(k) \Rightarrow_{\mathsf{ca}} \mathsf{Void}(k) \qquad (C_{\mathsf{U}})$$

$$\mathsf{Formed}(k) \wedge \mathsf{GrossDisparity}(k) \wedge \mathsf{NotGratuitous}(k) \Rightarrow_{\mathsf{ca}} \mathsf{Voidable}(k) \qquad (C_{\mathsf{D}})$$

Rule (C_{U}) (voidness, $M1 \to M4$) captures Example U; rule (C_{D}) (voidability, $M1 \to M3$) captures Example D; and (C_{valid}) represents the default classification for formed contracts in the absence of established defeating grounds. Note that $\mathsf{Valid}(k)$, $\mathsf{Void}(k)$, $\mathsf{Voidable}(k)$ are mutually exclusive. A logic of counts-as norms must capture their logical relations to *preserve consistency*; these can be formalised as follows.

$$\mathsf{Valid}(k) \Rightarrow \neg\mathsf{Voidable}(k) \qquad (Ex_1)$$

$$\mathsf{Valid}(k) \Rightarrow \neg\mathsf{Void}(k) \qquad (Ex_2)$$

$$\mathsf{Voidable}(k) \Rightarrow \neg\mathsf{Void}(k) \qquad (Ex_3)$$

where $\Rightarrow$ is the classical implication. In other words, we understand mutual exclusion in the sense of classical logic.

The statuses $\mathsf{Valid}(k)$, $\mathsf{Voidable}(k)$, and $\mathsf{Void}(k)$ are treated as mutually exclusive outputs of institutional classification. In particular, when multiple classification rules could apply, the institutional machine produces a single status upon leaving $M1$ (e.g. by a fixed priority among applicable grounds).

Voidability enables an avoidance pathway ($M3 \to M5$), whereas non-avoidance proceeds directly to repair dispatch ($M3 \to M6$). In the latter case, the entitled contractual party does not exercise the power of avoidance, and the institution recognises that the contract remains valid and that repair should therefore be undertaken. $\mathsf{AvoidAttempt}(i, j, k)$ represents an avoidance exercise by party i toward j (e.g. a legally relevant notice or claim). In our abstract model, $\mathsf{AvoidAttempt}(i, j, k)$ is recorded at the meta level only when the institution determines it as an effective exercise for contract instance k in the given case.

ID	Lane	Name	What it means	Enabled outputs / moves
$O0$	obj	Draft/Negotiate	No formed instance yet (or parties are renegotiating).	Propose terms, revise, reject; may reach acceptance.
$O1$	obj	In-force / Perform	Contract is formed and treated as operating at the object level.	Perform/monitor; may reach fulfilment or detect a perturbation.
$O2$	obj	Perturbation / Case-open	A disruption is detected (non-fulfilment, disagreement, or validity challenge).	Choice point: private revision vs. escalation; (if escalated) await meta-level output.
$O3$	obj	Post-outcome	After fulfilment or after institutional resolution; execute termination/repair plan.	Execute ordered repair (e.g. compensation or restitution), close, or continue cooperation.
$M0$	meta	Idle (background only)	No active proceeding; only background constraints apply.	On escalation, open institutional review.
$M1$	meta	Review (proceeding open)	Institution gathers/assesses grounds and routes the case.	Classify Valid/Voidable/Void; may recognise avoidance; dispatch repair.
$M2$	meta	Classified Valid	Contracts are recognised as enforceable.	Breach-handling admissible: enforcement/compensation (if Violated).
$M3$	meta	Classified Voidable	Contracts are valid but defeasible; avoidance power is potential for entitled party.	Either (i) dispatch repair while still valid, or (ii) transition to avoided.
$M4$	meta	Classified Void	Contractual validities are denied (as an unenforceable contract).	Restoration admissible: restitution/unwinding (where benefits transferred).
$M5$	meta	Avoided (effective avoidance)	Voidability is realised: contract becomes retroactively invalid.	Rescission + restitution/unwinding admissible.
$M6$	meta	Repair dispatched / Order	Institution issues the admissible repair family for the case.	Feeds back to object level: parties execute repair/termination accordingly.

Table 2. Governance states (object lane O and meta lane M).

$$\mathsf{Voidable}(k) \wedge \mathsf{AvoidAttempt}(i,j,k) \;\Rightarrow_{\mathsf{ca}}\; \mathsf{AvoidEffective}(k) \qquad (C_{\mathsf{A}})$$

From → To	Trigger (guard ∧ act)	Effect (informal)
$O0 \to O1$	act: acceptance (e.g. $\mathsf{Accept}(\cdot,\cdot,k)$)	Instance becomes formed/in-force ($\mathsf{Formed}(k)$, $\mathsf{InForce}(k)$).
$O1 \to O3$	fact: fulfilment succeeds	Discharge at object level; move to post-outcome governance.
$O1 \to O2$	fact: perturbation detected (non-fulfilment, disagreement, validity challenge)	Open a case at object level (choice point: revise vs dispute).
$O2 \to O0$	act: private revision agreed	Loop back to negotiation without institutional procedure.
$O2 \to M1$ (coupling)	act: escalation / dispute filing	Institution becomes procedurally active (opens review).
$M1 \to M2$	counts-as classification (no defeating ground established)	Institution recognises enforceability as valid.
$M1 \to M3$	counts-as classification (voidability grounds established)	Institution marks contract defeasible; avoidance becomes relevant.
$M1 \to M4$	counts-as classification (voidness grounds established)	Institution denies contractual enforceability; shift to restoration.
$M3 \to M5$	act: avoidance attempted + institution recognises effectiveness	Retroactive ineffectiveness ($\mathsf{NoEffect}(k)$); rescission/restoration dominates.
$M3 \to M6$	act: repair routing/order while still not avoided	Dispatch admissible repair while the contract remains valid (e.g. enforce/comp if $\mathsf{Violated}$).
$M2/M4/M5 \to M6$	act: repair routing/order issued	Institution dispatches the admissible repair family for the case.
$M6 \to O3$ (coupling)	fact: institutional resolution communicated/entered	Feed back: parties execute repair/termination in post-outcome state.

Table 3. Transitions of hierarchical governance machine (facts vs. acts + coupling).

Intuitively, $\mathsf{AvoidEffective}(k)$ means that the contract becomes retroactively ineffective and the governance pathway shifts to restoration-oriented consequences.

A voidable contract *remains valid unless the power of avoidance is exercised*. This non-avoided situation differs from that of contracts are valid and non-voidable ($\mathsf{Valid}(k)$). It is represented as $\mathsf{Voidable}(k) \wedge \neg\mathsf{AvoidEffective}(k)$. We

thus route breach-driven repair under both Valid and (non-avoided) Voidable:

$$(\mathsf{Valid}(k) \ \vee \ (\mathsf{Voidable}(k) \wedge \neg \mathsf{AvoidEffective}(k))) \wedge \mathsf{Violated}(k) \Rightarrow_{\mathsf{ca}} \mathsf{Route}(k, \mathsf{Enforce/Comp}) \quad (R_{\mathsf{eff}})$$

$$\mathsf{Void}(k) \Rightarrow_{\mathsf{ca}} \mathsf{Route}(k, \mathsf{Restitute/Unwind}) \quad (R_0)$$

$$\mathsf{AvoidEffective}(k) \Rightarrow_{\mathsf{ca}} \mathsf{Route}(k, \mathsf{Rescind} + \mathsf{Restitute}) \quad (R_{\mathsf{A}})$$

The transition from $M6$ back to the post-outcome state $O3$ is a second coupling, running from the meta level to the object level: the institutional order constrains what the parties execute in $O3$.

Trace sketches (Example U and D).
- **U (usury → void):** $O0 \to O1 \to O2 \dashrightarrow M1 \to M4 \to M6 \dashrightarrow O3$ (route: restitute/unwind).
- **D (gross disparity → voidable → avoided):** $O0 \to O1 \to O2 \dashrightarrow M1 \to M3 \to M5 \to M6 \dashrightarrow O3$ (route: rescind+restitute).

3.3 Repair Control as a Meta-level Submachine

Once a contractual interaction enters a perturbed state, repair control is precisely the meta-level segment $M1 \to (M2/M3/M4/M5) \to M6$: the process from review/classification, (optional) avoidance determination, to repair dispatch. See Figure 2 for the detailed architecture.

1. *Perturbation Detector* (supports $O1 \to O2$): monitors operational and structural anomaly signals.
2. *Institutional Gateway* (supports $O2 \to M1$ and $M1 \to M2/M3/M4$): routes the case to the appropriate institutional track.
3. *Repair Dispatcher* (supports $M2/M3/M4/M5 \to M6$): selects an admissible repair family consistent with validity/avoidance outcomes.
4. *Equilibrium Validator*: checks whether the ordered measures restore an acceptable normative balance under institutional constraints.

4 Agentic Reasoning Model

When agents draft and negotiate a contract, they must take the meta level contract governance into account — to anticipate possible post-outcomes (O3) that fulfill the contractual purpose, as well as the potential repairs (M6) when the contract is violated.

We propose a formal representation of practical reasoning that enables agents to *reason within* and *about* contracts, with particular emphasis on the coupling process between the object and meta levels. The model integrates two complementary frameworks in order to bridge the gap between institutional rules and individual practical reasoning:

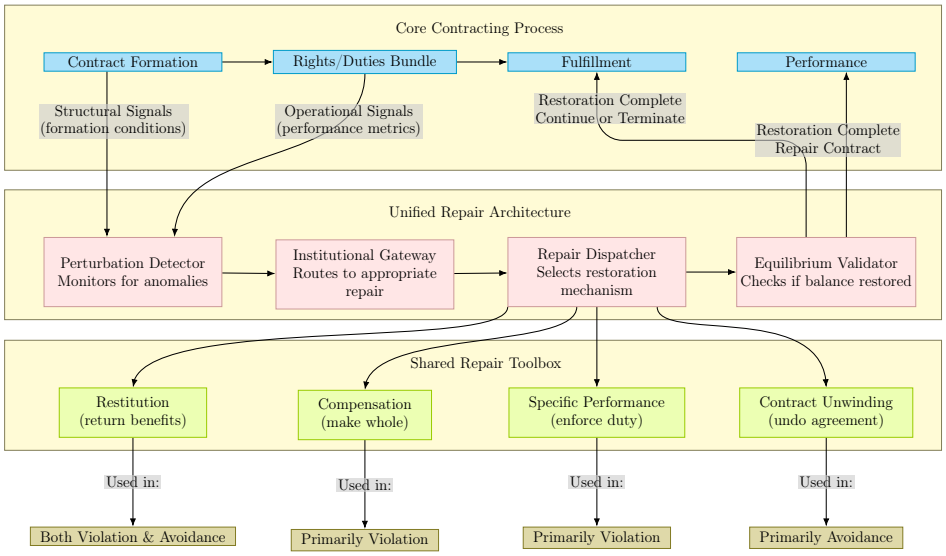


Fig. 2. Repair control as a meta-level submachine that classifies and dispatches admissible repair families.

– Classificatory counts-as norms [9]: These are employed to represent the institutional hierarchy — encompassing both operational procedures and repair controls — grounded in a rights-first perspective [8];
– Agent-based databases: These provide a computational structure to reason about the evolution of legal rights and duties over time [6].

By extending the database approach with counts-as norms, we enable institutional processes of validity classification and their associated repair mechanisms to be analyzed in a systematic, step-by-step operational manner. This synthesis allows agents to reason about their concrete reactions in terms of counts-as norms within their belief and intention databases. In particular, when a contract is counted as activated, breached, or avoided, the framework determines which beliefs and intentions are practically appropriate to adopt.

4.1 The Agentic Machine

We define an *agentic machine* A_i^t (for an agent $i \in \mathsf{Ag}$ at time t) as a structure of three different databases: $\langle \mathcal{B}_i^t, \mathcal{I}_i^t, \mathcal{N}_i \rangle$. This model facilitates agent-level practical reasoning regarding the institutional hierarchy.

Beliefs $\mathcal{B}_i^t$. This database is partitioned into strong and weak beliefs: $\mathcal{B}_i^t = \mathcal{B}_{i,s}^t \cup \mathcal{B}_{i,w}^t$. Strong beliefs support *requiring* (hard) gates — those inputs are assessed by institutions such as whether a contract is accepted or unfair, therefore primary for agents' databases as unchangeable when updating intention; weak beliefs motivate caution and information-seeking — which are derived beliefs from strong beliefs within other databases, such as that a contract is voidable.

This mirrors the previous work [6] in which the belief base is separated from "hard constraints" to "soft evidence".

Intentions $\mathcal{I}_i^t$. An intention base is consistent with explicit timestamps, storing all consistent *individual* and legal actions for agent i [6]. It is possible for the intention base $\mathcal{I}_i^t$ of agent i to store other agents' possible actions [16] like avoidance exercising. We adopt the intention-based revision methods that have been well discussed in the existing work [16, 6].

Normative and institutional model $\mathcal{N}_i$. This contains the institution's *meta-rules* involved in the institutional hierarchy. These are represented as *counts-as norms*. In our loan example, the key norms include counts-as rules (C_{valid}), (C_{D}), and (C_{A}). Agents use them to predict institutional outputs. We do not support that the contractual bounded parties share the same model of the institutional rules and so it is possible that $\mathcal{N}_i \neq \mathcal{N}_j$ when $i \neq j$.

4.2 Practical Reasoning and Logical Closure

Agents' autonomy is governed by the normative boundaries of the institution — the "rules of the game" that structure interaction [14]. Consequently, beliefs are influenced by the social context.

In this model, strong beliefs represent evidence-based truths and facts, while weak beliefs aim at social acceptance [14]. Weak beliefs are derived not only from an agent's internal states but also from the institutional status in which the agent is embedded. When an agent functions as a "group member" (a party to a contract), these weak beliefs constitute public facts that other members are expected to share.

Logical Closure. Our inference mechanism adopts a Searlean methodology [4] regarding social acceptance. While belief data is information-based, the institutional model is rule-based:

$$\text{weak belief} = \text{closure}_{\text{counts-as}}\ (\text{strong belief} \cup \text{intention}).$$

More precisely, weak beliefs are derived from strong beliefs and intention, logically closed under the counts-as norms. We do not specify which logic of counts-as norms $\Rightarrow_{\mathsf{ca}}$ to use, but many are available in the literature [9]. When using $Cn_{\mathsf{ca}}(\Gamma)$ to mean the logical closure of Γ on the specific logic of $\Rightarrow_{\mathsf{ca}}$, and so the weak belief database $\mathcal{B}_{i,w}^t$ is defined as $Cn_{\mathsf{ca}}(\mathcal{B}_{i,s}^t \cup \mathcal{I}_i^t \cup \mathcal{N}_i)$. We use $\mathcal{B}_{i,w}^t \vDash \varphi$ to indicate that $\varphi \in \mathcal{B}_{i,w}^t$. Consider the example of unfair load offer in Section 2. Let the agent machine $A_B^0 = \langle \mathcal{B}_B^0, \mathcal{I}_B^0, \mathcal{N}_B \rangle$ for agent B at time 0 be

- $\mathcal{B}_B^0 = \{\mathsf{Formed}(k_D)\}_B^0$;
- $\mathcal{I}_B^0 = \{\epsilon\}_B^0$ where ϵ indicate the empty intention;
- $\mathcal{N}_B = \{C_{\mathsf{valid}}, C_{\mathsf{D}}, C_{\mathsf{A}}\}$.

Thus, $\mathcal{B}_{B,w}^0 \vDash \mathsf{Valid}(k_D)$. Without further information, the weak belief database cannot infer any result beyond this and therefore determines the contract to be valid at this stage. Accordingly, the rule (C_{valid}) is applicable, whereas neither (C_{D}) nor (C_{A}) applies.

An agentic machine A_i^t is coherent, when $\mathcal{B}_{i,w}^t \nvDash \bot$. As $\mathcal{B}_{B,w}^0 \nvDash \bot$, the agent machine A_B^0 for agent B at time 0 is coherent.

4.3 Revision Mechanisms

These revision mechanisms are employed to maintain consistency when updating beliefs and intentions [16, 6]. The postulates of revision ensure that when new information conflicts with existing information, the latter is removed or revised accordingly. For example, since Valid and Voidable are mutually exclusive, if Valid is present in the database and a revision requires the addition of Voidable, then Valid must be removed during the update to preserve consistency.

An intention revision function $\otimes$ maps an agentic machine at time t and an intention to a new agent machine at time $t+1$ such that

$$\langle \mathcal{B}_i^t, \mathcal{I}_i^t, \mathcal{N}_i \rangle \otimes \alpha = \langle \mathcal{B}_i^{t+1}, \mathcal{I'}_i^{t+1}, \mathcal{N}_i \rangle$$

where the following postulates hold:
- $\mathcal{B}_i^t = \mathcal{B}_i^{t+1}$;
- $\langle \mathcal{B}_i^{t+1}, \mathcal{I'}_i^{t+1}, \mathcal{N}_i \rangle$ is coherent;
- If $\langle \mathcal{B}_i^{t+1}, \{\alpha\}_i^{t+1}, \mathcal{N}_i \rangle$ is coherent, then $\alpha \in \mathcal{I'}_i^{t+1}$;
- If $\langle \mathcal{B}_i^{t+1}, (\mathcal{I}_i^t \cup \{\alpha\})_i^{t+1}, \mathcal{N}_i \rangle$ is coherent, then $(\mathcal{I}_i^t \cup \{\alpha\})_i^{t+1} \subseteq \mathcal{I'}_i^{t+1}$;
- $\mathcal{I'}_i^{t+1} \subseteq (\mathcal{I}_i^t \cup \{\alpha\})_i^{t+1}$;
- For all $\mathcal{I''}_i^{t+1}$ with $\mathcal{I'}_i^{t+1} \subset \mathcal{I''}_i^{t+1} \subseteq (\mathcal{I} \cup \{\alpha\})_i^{t+1}$, the machine $\langle \mathcal{B}_i^{t+1}, \mathcal{I''}_i^{t+1}, \mathcal{N}_i \rangle$ is not coherent.

We can update the prior agentic machine A_B^0 with the intention of sending a note of avoidance $\mathsf{AvoidAttempt}(L, B, k_D)$, leading to $A_B^1 = \langle \mathcal{B}_B^1, \mathcal{I}_B^1, \mathcal{N}_B \rangle$ as

- $\mathcal{B}_B^1 = \{\mathsf{Formed}(k_D)\}_B^1$ and $\mathcal{I}_B^1 = (\mathcal{I}_B^0 \cup \{\mathsf{AvoidAttempt}(L, B, k_D)\})_B^1$.

The new agentic machine A_B^1 is coherent and it still has $\mathcal{B}_{B,w}^1 \models \mathsf{Valid}(k_D)$. However, the new intention does not bring to any institutional effect like $\mathcal{B}_{B,w}^1 \models \mathsf{Voidable}(k_D)$, since neither the necessary premise $\mathsf{GrossDisparity}(k_D)$ nor the premise $\mathsf{NotGratuitous}(k_D)$ is presented in the agentic machine A_B^1.

Belief Revision. A belief revision function $\circ$ maps an agentic machine at time t and a strong belief formula φ to a new agent machine at time $t+1$ such that

$$\langle \mathcal{B}_i^t, \mathcal{I}_i^t, \mathcal{N}_i \rangle \circ \varphi = \langle \mathcal{B'}_i^{t+1}, \mathcal{I'}_i^{t+1}, \mathcal{N}_i \rangle$$

where
- $\mathcal{B'}_i^{t+1}$ is the result of revising $\mathcal{B}_i^t$ with a strong belief φ that satisfies the AGM postulates [1, 5];
- $\mathcal{I'}_i^{t+1}$ is the result of revising the new beliefs with the empty intention ϵ so that coherence is restored, i.e. $\langle \mathcal{B'}_i^{t+1}, \mathcal{I}_i^{t+1}, \mathcal{N}_i \rangle \otimes \epsilon = \langle \mathcal{B'}_i^{t+1}, \mathcal{I'}_i^{t+1}, \mathcal{N}_i \rangle$.

Now we update A_B^1 with the strong belief $\mathsf{GrossDisparity}(k_D)$ and this leads to a new agent machine $A_B^2 = \langle \mathcal{B}_B^2, \mathcal{I}_B^2, \mathcal{N}_B \rangle$ as

- $\mathcal{B}_B^2 = (\mathcal{B}_B^1 \cup \{\mathsf{GrossDisparity}(k_D)\})_B^2$ and $\mathcal{I}_B^2 = \mathcal{I}_B^1$.

So A_B^2 is still coherent and we further know $\mathcal{B}_{B,w}^2 \models \mathsf{GrossDisparity}(k_D)$. We continue to update A_B^2 with strong belief $\mathsf{NotGratuitous}(k_D)$, leading to $A_B^3 = \langle \mathcal{B}_B^3, \mathcal{I}_B^3, \mathcal{N}_B \rangle$ such that $\mathcal{B}_{B,w}^3 \models \mathsf{NotGratuitous}(k_D)$. Given the counts-as norms,

this will lead to two new institutional results: $\mathcal{B}^3_{B,w} \models \mathsf{Voidable}(k_D)$ by applying rule (C_{D}) and $\mathcal{B}^3_{B,w} \models \mathsf{AvoidEffective}(k_D)$ by the rule (C_{A}).

Because $\mathsf{Valid}(k_D)$ and $\mathsf{Voidable}(k_D)$ are mutually exclusive, the postulates in belief revision require that the prior validity claim $\mathsf{Valid}(k)$ be removed from the updated belief base $\mathcal{B}^3_{B,w}$.

5 Related Work

Institutional legal effects are often represented with *constitutive* (counts-as) rules linking observable facts and actions to institutional statuses and normative consequences. Jones and Sergot explain how designated acts performed in context generate institutional effects such as competence and status change, thus separating physical behaviour from the institutional layer [11]. Subsequent work develops general patterns for constitutive norms and counts-as conditionals, clarifying how institutional facts (including classifications) are derived and how they constrain later moves [9]. In normative multi-agent systems, constitutive rules also serve as a design tool: agent moves count as legal acts and trigger institutional outputs [2]. Our meta-level component follows this tradition by treating validity classification and avoidance determination as constitutive outputs that govern repair routing and admissible continuations.

Rights-first approaches take Hohfeldian positions (claim/duty, power/liability, immunity/disability) as primitives for representing legal relations and their dynamics [10]. Formal reconstructions clarify how these positions can be represented as conditional legal relations and how their consequences propagate [13]. Work on legal competences highlights that *powers* are state-changing legal acts, not merely physical abilities, which is central to modelling avoidance [7]. Recent rights-first CTD models treat remedies and revisions as transformations over live bundles of directed positions [8]. We extend this stance to validity control, showing how institutional outcomes about the contract constrain the repair families available after a dispute.

Normative multi-agent systems and agreement technologies emphasise separating the computation of legal effects from agents' choices of which actions to take [3, 15]. Input/output logic supports modular architectures in which rule sets determine permitted and obligatory outputs given contextual inputs [12]. Architecture-oriented work also treats normative reasoning as a component in managing dynamic systems and change [16]. Our contribution instantiates these ideas for contracts by coupling an object-level contract-instance procedure with a meta-level institutional procedure for validity classification, avoidance determination, and repair routing.

6 Summary and Future Work

In this paper, we presented a rights-first hierarchical governance model that separates reasoning *within* a contract from reasoning *about* the contract as an

institutional procedure. The architecture couples two state machines: an object-level contract-instance machine for drafting, acceptance, performance, breach and escalation, and a meta-level institutional machine that classifies a contract as valid, voidable and void, optionally classifies avoidance as an exercised power with retroactive consequences, and dispatches a repair family. Coupling is given by classificatory counts-as rules from legally relevant grounds to institutional statuses and routing outputs. On the agent side, we introduced a database-based practical reasoning model with time-stamped strong beliefs and intentions (including others' actions), and weak beliefs obtained by closure under the institutional rules, enabling agents to anticipate classification and repair while revising plans. We illustrated the approach with two Hungarian-law traces: usury leading to voidness and restoration, and gross disparity leading to voidability, effective avoidance, and rescission-based repair.

One future work direction is to combine the state automaton for rights-first contrary to duty breach and repair from [8] with the state machine for validity control, voidness, voidability, and avoidance developed in this paper. It yields a unified automaton that covers the full range of rights based notions, including primary rights, contrary to duty rights, revision and compensation rights, and institutional construction and deconstruction of enforceability.

Acknowledgement

This paper was supported by the Austrian Science Fund (FWF) and the Luxembourg National Research Fund (FNR) together through Logical Methods for Deontic Explanations (LoDEx; doi: 10.55776/I6372 and INTER/DFG/23/17415164/LoDEx), and by the Luxembourg National Research Fund (FNR) through the project The Epistemology of AI Systems (EAI; C22/SC/17111440), DJ4ME – A DJ for Machine Ethics: the Dialogue Jiminy (O24/18989918/DJ4ME) and the project Symbolic and Explainable Regulatory AI for Finance Innovation (SERAFIN; C24/19003061/SERAFIN). It was also supported by the University of Luxembourg through the Marie Speyer Excellence Grant supporting Formal Analysis of Discretionary Reasoning (MSE-DISCREASON).

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic **50**(2), 510–530 (1985)
2. Boella, G., van Der Torre, L.: Constitutive norms in the design of normative multiagent systems. In: ICLIMA 2005. pp. 303–319 (2005)
3. Boella, G., Pigozzi, G., van der Torre, L.: Five guidelines for normative multiagent systems. In: JURIX 2009, pp. 21–30 (2009)
4. Boella, G., Van Der Torre, L.: Norm negotiation in multiagent systems. International Journal of Cooperative Information Systems **16**(01), 97–122 (2007)
5. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artificial intelligence **89**(1-2), 1–29 (1997)

6. Dong, H., Doder, D., Li, X., Markovich, R., van der Torre, L., van Zee, M.: Rights and practical reasoning in deontic logic. In: DEON 2023. pp. 1–19 (2023)
7. Dong, H., Roy, O.: Dynamic logic of legal competences. Journal of Logic, Language and Information **30**(4), 701–724 (2021)
8. Dong, H., van der Torre, L., Yu, L.: Contrary-to-duty rights: From Hohfeld to agreement revision. In: JURIX 2025, pp. 61–73 (2025)
9. Grossi, D., Jones, A.: Constitutive norms and counts-as conditionals. In: Handbook of deontic logic and normative systems, pp. 407–441. College Publications (2013)
10. Hohfeld, W.N.: Some fundamental legal conceptions as applied in judicial reasoning. Yale Law Journal **23**(1), 16–59 (1913)
11. Jones, A.J., Sergot, M.: A formal characterisation of institutionalised power. Logic Journal of the IGPL **4**(3), 427–443 (1996)
12. Makinson, D., Van Der Torre, L.: Constraints for input/output logics. Journal of philosophical logic **30**(2), 155–185 (2001)
13. Markovich, R.: Understanding hohfeld and formalizing legal rights: the Hohfeldian conceptions and their conditional consequences. Studia Logica **108**(1), 129–158 (2020)
14. North, D.C.: Institutions, institutional change and economic performance. Cambridge university press (1990)
15. Ossowski, S.: Agreement Technologies. Springer (2012)
16. van Zee, M.: Rational Architecture: Reasoning about Enterprise Dynamics. Ph.D. thesis, University of Luxembourg (2017)

Author Index

ISBN 978-4-915905-98-8