



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

in cotutela con University of Luxembourg - Université du Luxembourg

**DOTTORATO DI RICERCA IN
LAW, SCIENCE AND TECHNOLOGY**

Ciclo 37

Settore Concorsuale: 12/H3 - FILOSOFIA DEL DIRITTO

Settore Scientifico Disciplinare: IUS/20 - FILOSOFIA DEL DIRITTO

**LEGAL REASONING THROUGH FACTOR-BASED REASONING AND
ARGUMENTATION IN THE CONTEXT OF EXPLAINABILITY**

Presentata da: Marco Billi

Coordinatore Dottorato

Monica Palmirani

Supervisore

Giovanni Sartor

Supervisore

LEON VAN DER TORRE

Abstract

This thesis explores methods for explaining AI decisions, with a particular focus on ensuring users' rights to appeal such decisions. The central research question guiding this work is: **How can we ensure the right to appeal an AI-generated decision?** This question is critical for safeguarding fairness and transparency in AI-driven systems, especially in the legal domain, where the outcome can have serious implications for individuals.

While the AI Act establishes a 'right to request clear and meaningful explanations' from AI systems involved in decision-making, it stops short of mandating fully transparent, white-box models. This thesis introduces methods to meet transparency requirements by combining symbolic legal models and machine learning techniques to enhance the explainability of AI-driven legal decisions.

A series of experiments provide guidance on representing legally relevant factors and establishing logical connections to outcomes, starting with symbolic expert systems for EU and national law compatibility. The thesis extends to machine learning models for classifying legal judgments, balancing transparency with usability. Expert systems are shown to excel in transparency by offering step-by-step reasoning that enhances user understanding, while machine learning models improve accessibility by streamlining interaction.

In conclusion, ensuring user rights in AI-driven legal contexts requires clear, comprehensible explanations of legal factors, grounded in both statutory and case law. This thesis emphasizes that AI in legal domains must support rational decision-making, aligning with legal standards and user expectations.

Acknowledgements

The work has been supported by the “CompuLaw” project, funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant Agreement No. 833647). The work has been supported by the “CrossJustice” project, funded by the European Union’s Justice Programme (2014-2020), Grant Agreement 847346. The work has been supported by the “Facilex” project, funded by the the European Union’s Justice Programme (2022), GRANT AGREEMENT Project 101089634.

Contents

1	Introduction	1
1.1	Definition	1
1.2	Explanation vs Justification	3
1.3	How to evaluate explainability?	4
2	State of the Art	7
2.1	Logic Based Approaches	7
2.2	Computable Law	13
2.3	ML Based Approaches	18
2.4	Legal Background	22
3	Legal Rule Modelling	27
3.1	Rule Modelling	27
3.1.1	Expert System for European and National Criminal Law . .	27
3.2	Legal Principles	30
3.2.1	Isomorphism	30
3.2.2	Reification	30
3.2.3	Defeasibility	31
3.3	Technical Requirements for a Reasoning System	32
3.3.1	Tractability and Computational Complexity	32
3.3.2	Procedural Order of Inference Rules	32
3.3.3	Semantic Values of Expressions	32
3.3.4	Control Strategies	33
3.3.5	Explanation and Justification	33
3.4	Conclusion	33
3.4.1	Criminal Procedure Law Formalized through Logical Repre- sentation	34
3.4.2	Technical Setup	48
3.5	Argumentation	59
3.5.1	Case 1	60
3.5.2	Case 2	61

CONTENTS

3.5.3	How to Write Conflicts?	61
3.5.4	Incorrect Transposition: diverging implementation	63
3.5.5	Incorrect Transposition: diverging requirements	66
3.5.6	Incorrect Transposition: conformity check	72
3.6	The Role of Interpretation	75
4	Factor Based Approach	79
4.1	Machine Learning for Extraction	79
4.1.1	Automatic Factors and Human Factors	83
4.2	Decision Trees	92
4.2.1	Shapley Values	93
4.2.2	A reflection on XAI	96
4.3	Ontologies through LLMs	97
4.3.1	Outcomes and Discussion	102
4.4	Explanation in Machine Learning Systems	103
5	Combining the two Approaches	105
5.1	Combining Symbolic and Sub-Symbolic Approaches - From Text To Code	105
5.2	Experimental Setup	105
5.2.1	Case 2	107
5.3	Experimental Setup	108
5.4	Article Generation	109
5.4.1	Prompts	110
5.4.2	Output	111
5.4.3	Discussion	114
5.5	Case Generation	115
5.5.1	Prompts	116
5.5.2	Output	117
5.5.3	Discussion	120
5.6	From Code to Text	121
5.6.1	Memory	122
5.6.2	Natural Language Translation	123
5.6.3	Comparison of legal sources	127
5.6.4	Analysis of Case Similarity	131
6	Conclusions	141
	Bibliography	145

List of Figures

3.1	Arguments from the example in Subsection 3.5.4.	65
3.2	Arguments from the example in Subsection 3.5.6.	69
3.3	Arguments from the examples in Subsection 3.5.4 (left) and Sub- section 3.5.6 (right).	71
3.4	Conformity example	74
4.1	SHAP analysis over trained Decision Trees on the two structured datasets.	94
4.2	Small IT Decision Tree	95
4.3	Small BR Decision Tree	97
4.4	Summary of case 44	98
4.5	Prompt ontology	100
4.6	Graph using factual elements	101
4.7	Graph obtained using factors	102
5.1	Pipeline for Article and Case generation.	106
5.2	Natural Language Explanation	123
5.3	Comparison of Legal Sources	128
5.4	Analysis of Case Similarity	132

LIST OF FIGURES

Listings

code/example_bis.txt	63
code/example-2_bis.txt	66
code/directive.txt	67
code/polish.txt	68
code/facts.txt	69
3.1 Conformity between directives and national laws check	72
5.1 Article 4(1) of the 2002/54 FD	111
5.2 Article 4(6) of the 2002/54 FD - Generated by GPT-4o	111
5.3 Ouput Prompt 2 - Generated by GPT-4o	113
5.4 Output Prompt 3 - Generated by GPT-4o	113
5.5 Output Prompt 1 Task 2 - Generated by GPT-4o	118
5.6 Output Prompt 2 Task 2 - Generated by GPT-4o	118
5.7 Natural Language Translation - EU Directive	125
5.8 Natural Language Translation - Dutch	126
5.9 Prompt Task 2	128
5.10 Comparison of Legal Sources	129
5.11 Case Id2	135
5.12 Prompt	136
5.13 Case Analysis	137

Chapter 1

Introduction

1.1 Definition

In recent years, the use of Artificial Intelligence (AI) in decision-making processes has expanded across various domains, including finance, healthcare, and the legal system. While AI offers immense potential for efficiency and accuracy, its deployment raises significant concerns about transparency and accountability, particularly when individuals are affected by automated decisions. One of the most pressing challenges is ensuring that users understand how AI systems arrive at their conclusions and, more critically, how they can challenge or appeal those decisions when necessary.

This thesis explores methods for explaining AI decisions, with a particular focus on ensuring users' rights to appeal such decisions. The central research question guiding this work is: **How can we ensure the right to appeal an AI-generated decision?** This question is critical for safeguarding fairness and transparency in AI-driven systems, especially in the legal domain, where the outcome can have serious implications for individuals.

To address this question, we will focus on user-centered approaches to explainable AI (XAI). Specifically, we will investigate how AI systems can be designed to provide explanations that are comprehensible and actionable for users, allowing them to make informed decisions about whether and how to challenge an outcome. Additionally, we will examine methods for evaluating the effectiveness of these ex-

planations, ensuring that they meet the legal and ethical standards required for fair decision-making.

By the end of this thesis, we aim to provide a framework for integrating user-based explainability into AI systems, along with robust evaluation techniques for assessing whether the explanations provided are adequate for supporting the right to appeal.

The thesis will be structured as follows:

- This chapter will provide an overview of the thesis, and the main topics which will be covered
- The technical background and state of the art will be introduced in Chapter 2, along with a legal analysis of the concept of 'explanation
- in Chapter 3 will focus on Logic Programming approaches for building expert systems, exploring how argumentation can be implemented to develop systems capable of complex reasoning.
- the following chapter, Chapter 4 presents a machine learning-based approach for extracting legally relevant elements from judicial decisions.
- finally, in Chapter 5 we propose a method that combines both approaches, providing a comprehensive discussion of its implications for explainability from both technical and legal perspectives, at the current state of the art from both the technical and legal side.

With regard to existing literature, the groundwork for my thesis has been laid by previous research done by [5], as well as [71]. These works trace the evolution of explanatory methods in legal AI systems, examining current practices, and exploring future directions for enhancing transparency and interpretability in legal decision-making processes.

The authors discuss various approaches to explanation in AI, including rule-based systems, case-based reasoning, and machine learning models. They highlight the challenges associated with each method, such as the trade-off between accuracy and explainability, and the need for models that are both effective and understandable to legal professionals and the public.

Therefore, my thesis is a practical overview that focuses on ensuring the right to appeal AI-generated decisions in the legal domain. My research builds upon these foundations by proposing methods to enhance the explainability of AI decisions, thereby contributing to the broader discourse on responsible AI in legal contexts.

1.2 Explanation vs Justification

To summarize what will arise throughout this thesis, it should be highlighted that the outcomes provided in the following chapters are intended as guidelines in the development of AI systems in the legal domain. There is much debate over what constitutes an explanation, and even legislators do not fully agree over its definition, or whether to recognise it as a right.

The AI Act explicitly provides a "right to request from the deployer clear and meaningful explanations of the role of the AI system in the decision-making procedure and the main elements of the decision taken." However, it does not mandate a clear right to explanation in the sense of offering a fully transparent, white-box system. Instead, the Act seeks to fulfill its policy objectives by emphasizing transparency, including adequate documentation, and ensuring human oversight.

In this thesis, we will introduce methods aimed at addressing these transparency requirements, though the exact nature of what constitutes sufficient documentation remains ambiguous. The experiments outlined here will incorporate symbolic elements that describe legally relevant factors, establishing logical connections between these factors and the decision-making outcomes. For instance, we will explore symbolic expert systems that address legal compatibility between EU and national laws. Later, we extend our approach by incorporating machine learning systems that facilitate the extraction and representation of legal judgments, making them easier to comprehend and classify. These methods provide a pathway toward enhancing transparency in AI-driven legal decisions, while aligning with the goals of both the AI Act and broader legal frameworks.

The conclusion of this thesis will emphasize that, to ensure a user's rights are upheld, it is essential to provide them with the necessary elements for decision-making from a rational perspective. While the explanation may involve technical requirements or methodologies used in system development, all AI systems applied

in the legal domain — particularly those related to outcome prediction and decision support — must offer a clear analysis of the legally relevant factors. These factors should be derived from a factual overview of the situation, highlighting which elements a judge should consider and how they influence the final outcome.

1.3 How to evaluate explainability?

In designing systems for legal decision-making, it is crucial to provide users with control over the amount and type of information they receive. Different users have varying needs and expectations based on their roles, and a one-size-fits-all approach can be inefficient or overwhelming. By allowing users to adjust the level of detail or focus, systems can better align with individual preferences. For instance, some users might only want high-level summaries, while others may prefer detailed explanations or reasoning paths. This flexibility ensures that users can access the information that is most relevant to them without being inundated with unnecessary complexity.

Different users within the legal domain have distinct information requirements. Citizens or parties involved in a case may only be interested in the final outcome of a decision, such as the ruling or judgment. They are generally not concerned with the intricate legal reasoning or arguments that led to the outcome. On the other hand, legal professionals, such as judges, require a thorough understanding of the reasoning behind the decision, including how laws were interpreted and applied. This divergence in information needs highlights the importance of creating adaptable systems that cater to each user type by offering tailored outputs based on their roles and responsibilities.

In the context of explainable AI (XAI), a key consideration is not just whether an AI system can be explained, but how much information can be made available to the user. The goal of XAI is to provide users with transparency and understanding of how a system arrived at a particular outcome. However, this doesn't mean overwhelming users with all the inner workings of the system. Instead, it focuses on offering the right amount of information, depending on what the user seeks. For some, a simple explanation may suffice, while others may require more granular details about the decision-making process. This approach helps ensure that the

system remains accessible and useful to a broad range of users with different levels of expertise.

Finally, the emphasis should be on ensuring that for each outcome generated by the system, there is a probable and logical reasoning that can be followed. This ensures that the decision-making process is transparent, understandable, and justified, which is particularly important in the legal domain. By focusing on delivering clear reasoning for outcomes, systems can help build trust, especially when decisions affect significant aspects of people’s lives. Providing plausible explanations also ensures that decisions are not seen as arbitrary but are grounded in a consistent and logical framework that can be reviewed and understood by all relevant stakeholders.

When evaluating explainable AI (XAI) systems, the evaluation criteria must be specific to the type and domain of the system in question. XAI is highly contextual, meaning that the methods and metrics used to evaluate a system in one field may not be appropriate in another. For instance, an XAI system designed for criminal law have different requirements than civil law systems. This specificity is crucial because different domains have unique demands regarding the transparency and clarity of decisions. Therefore, any evaluation must take into account the particular legal frameworks, practices, and expectations relevant to that domain.

The evaluation should also be based on the users of the system, as their needs and interactions will dictate how useful and understandable the explanations are, similar to the concept of *human-grounded* interpretability as described by [26]. This highlights a limitation in the experiments conducted in this thesis, as access to a diverse pool of real-world users, such as legal professionals and citizens, was not available. Without user feedback, the evaluation is constrained, and the results are less representative of how the system would perform in practice. User-specific evaluations are essential because they reveal whether the explanations provided by the system are clear, relevant, and aligned with the expectations of those who rely on the system to inform decisions.

Another important criterion for evaluation is how closely the system’s reasoning aligns with the law and case law. In legal systems, consistency and adherence to established precedents are critical for trustworthiness. Therefore, any XAI system developed for legal decision-making must be evaluated based on how well its

outputs reflect the underlying legal principles and case law. This alignment ensures that the system not only provides transparent explanations but also maintains legal integrity and can be trusted to make decisions within the correct legal framework.

Additionally, users should be given access to technical information from the development stage of the system. This transparency is necessary to assure the technical reliability of the system, especially for legal experts who may need to scrutinize the system’s methodology and ensure its processes are sound. Understanding the technical framework helps to verify that the system is not only explainable but also built on a reliable and accurate foundation, which is essential for its deployment in sensitive and high-stakes domains like law.

Lastly, the role of the user interface (UI) in XAI evaluation is an open-ended question, but likely crucial due to the social nature of explanations. The UI serves as the main interaction between users and the system, shaping how explanations are delivered and understood. A well-designed UI can enhance the clarity of explanations and make complex reasoning more accessible. Since the explanation process involves communication between the system and the user, the interface plays a significant role in how effectively this communication happens. Therefore, the UI should arguably be considered as part of the overall evaluation process, as it can significantly impact the user’s experience and satisfaction with the system.

Chapter 2

State of the Art

2.1 Logic Based Approaches

The field of law is increasingly embracing digital technologies, with symbolic legal systems playing an established role in automating legal reasoning and juridical tasks across various domains. Such systems achieve their goal by representing expert legal knowledge in a formal and computable way using high-level programming languages, with the approach contributing to the creation of symbolic legal reasoners and computable contracts, both of which have found application in a variety of socio-juridical domains, such as credit, insurance, welfare and pensions [52, 55, 53, 81, 54, 4].

Symbolic AI involves systems that carry out explicit manipulation of symbols, where these symbols are interpreted at a high level, making them understandable to humans. Prior to the widespread use of machine learning, symbolic AI dominated the field. Unlike machine learning, which allows systems to learn behaviors through experience, symbolic AI encodes system behaviors directly through a manually crafted representation of the problem domain.

A key example of symbolic AI is the rule-based system, where an AI operates logically according to predefined rules and axioms that represent the domain. These systems are typically developed in collaboration with experts who provide the necessary domain-specific knowledge. For instance, an expert system can be designed to determine whether a person qualifies for a bank loan. In this case, de-

velopers and domain experts create a set of rules which when combined with the applicant's information (the axioms), generate an outcome. The system's decision can be easily explained by tracing through its rules and axioms, and its behavior is guaranteed to follow the intended logic since it is explicitly programmed by the experts. Any mistakes or undesired outcomes can theoretically be corrected by modifying the knowledge representation, although this is often more difficult in practice [14].

Besides rule-based systems, other symbolic AI knowledge representations exist, such as frames, semantic networks, and ontologies.

When a domain can be clearly defined, symbolic AI not only functions correctly but can also justify its behavior. However, in cases where the domain cannot be explicitly defined, symbolic AI methods tend to underperform compared to modern machine learning techniques. In fields like computer vision or natural language processing (NLP), deep learning vastly outperforms symbolic AI. However, deep learning models, while powerful, function as "black boxes" and are unable to explain their decisions, sometimes learning undesirable patterns.

There are several drawbacks to machine learning and symbolic AI.

Machine learning requires vast amounts of data, significant computational resources, and is vulnerable to adversarial attacks, bias, and lack of robustness. These models do not always reason logically and are often opaque, offering no clear explanation for their decisions. Symbolic AI requires human experts to establish formal rules, which is time-consuming and often impractical for complex domains. These systems struggle with noisy or highly variable data, and their logic-based methods are often computationally complex, limiting the tasks they can handle.

On the first end, a clear and well-known example of symbolic system used for automated reasoning is represented by rule-base systems. These models ground their functioning on the application of an expert knowledge base made of rules, relationships and statements, which are applied through an inference engine to factual data related to specific cases, enabling transparent and justifiable reasoning.

While the symbolic approach offers advantages in terms of justifiability and trustworthiness, with the aforecited systems being prominent examples, it never-

theless holds a well established and documented set of problems, including complications tied to the communicability of their outputs to laypeople. Being often presented in high-level programming languages, the output of such systems can be challenging to understand and appreciate by users, due to its technical and expert nature.

Symbolic systems, due to their predetermined nature and scope, struggle to capture the complexities of legal rules, a limitation known as the knowledge representation bottleneck. The nuances, exceptions, and ambiguities inherent in the law and legal language are difficult to encode in a way that allows for agile and cost-effective updates. This inflexibility hinders the ability of legal expert systems to accurately represent real-world legal scenarios.

Considering these drawbacks, a different approach and branch of legal informatics is developing around the usage of Large Language Models, a type of generative artificial intelligence system that leverages deep learning methodologies to achieve the processing and creation of text in both natural and programming languages. Such models have been the subject of research in the legal domain, with systems being tested on tasks such as retrieval of legal information, contract drafting, artificial legal counsel, and legal writing, including simplification of legal documents and generation of explanations[12, 80, 79, 24].

However, concerns exist regarding the reliability and justifiability of their outputs, particularly in the context of complex legal tasks, given the intrinsically opaque nature of their reasoning, their misinterpretation of specific contextual information and the complications brought by the phenomenon of hallucinations [74]. Legal provisions are often represented as rules, with legal reasoning revolving around the application of these rules to specific facts. As a result, numerous scholars have concentrated their efforts on creating rule-based systems tailored to the legal domain, along with the corresponding logical inference engines. A notable example of such an effort is the modeling of the British Nationality Act [76], which served as a foundational step for the development of various rule-based systems in legal theory [73].

The predominant approach to modeling legal rules involves representing legal knowledge through definite Horn clauses—clauses that consist of a single conclusion (the head) and zero or more premises (the body). These rules are typically

formulated using a subset of predicate logic, enhanced with mechanisms for defeasibility or probabilistic reasoning. A rule-based system typically consists of two key components: (1) a rule base, which contains the legal rules, and (2) an inferential engine, also referred to as a rule interpreter.

Prolog (PROgramming in LOGic) is a programming language designed around a few fundamental mechanisms, such as pattern matching, tree-based data structures, and automatic backtracking. Its structure makes Prolog particularly suitable for modeling problems involving objects and relationships—an essential feature when representing legal norms. Over the years, Prolog has become widely employed in the legal domain to formalize legal texts and assist in legal reasoning processes, including dispute resolution.

In Prolog, programs are constructed using terms, which can be constants, variables, or structures. Constants denote specific objects or relationships and are divided into atoms and numbers. Variables, which resemble atoms but begin with capital letters or underscores, serve as placeholders that can be instantiated with actual values during computation. Structures, also known as compound terms, are complex objects made up of multiple components. During execution, Prolog allows variables to be substituted with concrete objects, a process known as instantiation. Prolog features operators such as Negation ($\backslash +$), Conjunction ($,$), and Disjunction ($;$), and even allows the creation of custom operators. Built on a closed-world assumption, Prolog uses negation by failure, meaning that if Prolog cannot prove a rule's body, it automatically assumes the negation of the rule's head.

Legal reasoning often requires careful consideration of complex relationships between general legal concepts, their exceptions, and the specific contexts of individual cases. This complexity involves addressing arguments and counterarguments, including attacks, defenses, overrides, and undercuts, within a legal framework [69]. To manage this complexity, several frameworks have been developed to analyze the dialectical relationships between arguments in legal contexts, assessing which arguments are justified or overruled. One such framework is ASPIC+ [59], which offers a structured approach to argumentation and has become one of the key models for legal reasoning within the realm of structured argumentation.

In order for such systems to adequately model the law, there are a number of requirements, and issues that need to be addressed [32]. We believe that the

most important such requirement pertains to ensuring that legal experts have full awareness of and control over the relation between the computable model of the law and the legal content that the model is meant to capture. For this purpose, they must be able to map such legal content, as expressed in natural language, into the computable representation, and moreover, they must be able to link automated inferences by the system into reasoning steps that are meaningful and reasonable to a human.

The significance of this requirements can be understood if we take into account on the one hand the nature of the law and on the other hand and on the other hand the nature of computable representations on the other hand.

The law is a multilayered institutional reality: certain social or natural facts trigger (create, modify or terminate) certain institutional facts (legal effects), which, in their turn, may trigger further institutional facts. The triggered institutional facts may consist in normative positions (i.e., permission, obligations or rights), but may also consist in legal qualifications (e.g., the status of being a citizen) or even in the creation of a new legal entity (e.g., a company or a public office).

A legal reasoner must determine whether fact exist that trigger legal effects, thus connecting the linguistic formulations to be found in legal norms and specifications of them to the corresponding social or natural realities, but also to determine whether the linked legal effects do take place, which may required engaging with the interpretation of the corresponding norm, according the underlying rational and shared expectations, and considering possible exceptions and counterarguments. In this activity the legal reasoner is challenged, but also assisted by social knowledge, as provided in particular by precedents cases and doctrinal contributions. To make a reasoned assessment, legal reasoners need awareness of the interests involved in the situation and of the ways in which their decision is going to affect that situation.

In relation to the complexity of legal problem solving, what can be achieved through automated inference may only have a partial significance.

In fact, a computer system, even an AI system, only operates at the syntactical level: it manipulates data structures without taking into account the realities which these data are assumed to represent. In other terms the computable meaning of

the symbolic structures in a knowledge base (but also of the number vectors a non-symbolic machine learning system) is determined only by the way in these structure are to be processes, it is not grounded in the natural or social entities to which the symbols refer. This grounding, on the contrary is a key aspect of human understanding of meaning, and plays a key role in the application of the law[11, 7]. This means that a legal knowledge base acquires meaning only to the extent that its symbols can be understood by humans, who are able to ground the symbols in the extralinguistic reality to which such symbols refer to. Only through this process the terms used by the programmer in building the knowledge base acquire their meaning, and only by being so understood, it is possible to determine whether the program correctly reflects the legal reality. Thus, for instance, the words “father” and “mother” (or whatever symbol used to express them, eg. the abbreviations “ft” and “mt”) which appear within a knowledge base do not mean to the system the social reality that corresponds to such words. Their meaning may be restricted to the fact that both entail the property “parent”.

The need for “grounding” does not only concern the meaning of the basic proposition, such as the fact that a person is the father of another person, it also concerns the further properties that are to inferred according to the law. Thus, the meaning of “citizen” is not reducible to the connection present in a knowledge base: on the contrary, by grounding that symbol in the legal reality (citizenship as a legal status) that it is assumed to express, can we understand its meaning and determine whether a computable formalisation adequately captures the intended aspect of that meaning.

Besides grounding, also awareness of socio-legal contexts is required for the application of the law. That awareness as well is not captured by the computable representation: on the contrary the adequacy of a computable representation has to be judged by considering the extent to which it captures the aspects of the context that are relevant for the use of that computable representation.

These considerations show the need that computable representation of the may has to be be fully understandable, in particular to those who have full awareness of the meanings that are to be associated to the expression in the computable representation, and of the contexts in which these meaning have to be located. We argue that this objective can be achieved by making so that legal content,

even when formalised in a knowledge base, is expressed in natural language, so that it can be understood, contextualised, and critically assessed by competent lawyers.

Humans lawyers should be in the loop, not only over the loop: thus must fully understand the content of the knowledge base, by linking it to the social reality it refers to, and consequently to check whether the decision making process reflects that social reality [56].

To enable human actors to understand the system's functioning and output we propose that computable representations that approximate the written form of the law. The law is in fact encoded in written form in a specialised natural language, and conveys certain messages and instructions, so that it is understandable by humans. Given the complex nature of the law, understanding the normative content of legal provisions is often a difficult task, and may rely on the analysis of contextual information provided by multiple correlated legal documents, on the interpretation of legal terms or concepts, on doctrinal writings, and others. However, an essential precondition is the accessibility of legal content, namely to enable access to the law, so that citizens have knowledge of it, and may engage in reasoning on the law itself.

2.2 Computable Law

The intersection of AI and law is mainly centered around formal argumentation and explainable AI (XAI). In this context, legal reasoning, particularly the process of justifying and explaining legal decisions, has been a significant area of development. Legal argumentation, which deals with interpreting and applying laws to specific cases, benefits from AI tools that structure reasoning processes and clarify decisions.

A core area of focus is modeling legal argumentation within AI frameworks. Techniques from AI-based argumentation have been applied to simulate legal reasoning, supporting processes like probabilistic reasoning, analogy-based arguments, and policy-based decision-making. These AI systems aim to create a transparent, explainable framework to justify legal outcomes. Legal decisions, often constrained by existing norms, require justification by referencing legal rules and facts, empha-

sizing the dialectical nature of law.

XAI has emerged as a key focus in ensuring that AI systems in the legal domain not only make decisions but also provide clear justifications for those decisions. The need for transparent decision-making in legal contexts is critical, especially when decisions affect individuals' rights and freedoms. AI researchers working in law have made significant strides in developing formal models that ensure automated decisions can be thoroughly explained, aligning with normative expectations in the legal community.

One challenging area is connecting legal justifications with stable normative explanations, especially in defeasible logic frameworks. This area emphasizes how AI can be used not only to justify decisions but to ensure that the reasoning process remains robust, even when new facts are introduced. This requires AI models that understand and simulate complex legal reasoning processes, balancing competing rules, exceptions, and normative principles.

In today's society it is inevitable that legal tech applications will be employed by the judicial body in order to support the ever increasing workload of legal professionals, in the decision making process, as well as the legal drafting phase. More and more systems are being developed to help the drafting of legal acts, to find legal sources in the databases, and most importantly, to provide legal professionals and citizens alike with a first response to their demands. In this context, systems based on models of the law or judicial decision-making, may help to assess the likelihood of a given outcome, enabling better-informed approaches to litigation on the part of litigants, and improving the efficiency, consistency, and fairness with which judicial decisions are reached.

Alternative dispute resolution is becoming ever more popular, and ways to achieve a positive result through the employment of artificial intelligence system is starting to see the first real-life use cases, a process that has been kick started with the Covid-19 emergency.

With the recent developments in the field of AI applications in the law, this prerogative may be under scrutiny. Machine learning systems have been widely explored to allow algorithmic application of law, especially in the domain of predictive justice. Just to mention a few examples, this approach has proven effective in assessing the crimes-recidivism risk ([82]), in predicting US Supreme Court de-

cision on lower courts' judgements and in detecting unfair clauses in standardised contractual Terms of Service [49]. Such systems have the advantage of relying on an ever growing amount of data that will increasingly refine and improve the software performance. However, the downside of data-driven models is the inherent opacity of their functioning. It is hard, and prone to misinterpretation, to identify the relational process that, starting from an input, led to a certain output.

In the legal field, the right to obtain an explanation is fundamental. This principle extends across all stakeholders, from citizens to legal professionals and judges. The reasoning behind decision-making is not only a constitutional principle under EU law (Article 41 of the Charter of Fundamental Rights) but also directly impacts the correctness of legal decisions.

The right to an effective remedy, enshrined in Article 13 of the European Convention on Human Rights and Article 47 of the Charter of Fundamental Rights, is particularly at risk when decisions lack transparency. This concern is especially pronounced in criminal law, where unexplained decisions can undermine due process and justice [19].

In legal decision-making, the explanation of a decision can be more critical than the decision itself, particularly when AI systems are involved. A judge may rely on an automated system to identify applicable legal norms but still exercise discretion in forming their judgment. The explanations provided by AI systems influence human reasoning and can shape judicial decision-making [17].

Computable law must remain accessible to all, regardless of technical expertise. Maintaining a close link between computable representations of legal texts and their natural language counterparts ensures that humans remain in control of decision processes. This is crucial for preserving legal reasoning, applying nuanced interpretations, and integrating ethical considerations that AI systems cannot comprehend [56].

If human-comprehensible legal texts were to disappear, citizens would face diminished legal clarity, and legal professionals would lose oversight of AI-driven decisions. The extreme consequence of this scenario would be law applied—or even formulated—entirely by artificial agents, without human intervention or interpretability. As legal knowledge evolves, human interaction with AI systems must increase accordingly.

The goal of human-readable, logic-based computer languages in law is to empower legal professionals and citizens to read, understand, and modify encoded legal rules. This reduces reliance on technical experts and allows legal practitioners to actively engage in legal rule drafting. For example, a judge should be able to trace an automated decision’s reasoning, identify errors, and exercise discretion where necessary. Errors may stem from overly literal rule application, failure to consider legal principles, or omission of relevant precedents [33].

A similar advantage exists in smart contract modeling, where contracts result from complex negotiations and precise wording. The transformation of contracts into code must preserve their intended legal meaning. If only IT specialists without legal training conduct this process, subtle but critical legal nuances may be lost. Therefore, legal professionals must be able to understand contract code to ensure faithful transposition of agreements.

To prevent loss of control over legal frameworks, states should ensure that the natural language form of law remains primary while developing AI-driven legal reasoning systems. These systems should enhance human legal decision-making rather than restrict access and control for both professionals and citizens.

Requirements for Legal Rule Representation One of the key requirements for logic-based legal representation is isomorphism, ensuring a one-to-one correspondence between formal models and natural language legal texts. Karpf outlines five necessary conditions for achieving isomorphism [43]:

- Each legal source is represented separately.
- The structure of each legal source is preserved.
- The mutual relations, references, and connections between legal sources remain intact.
- The representation of legal sources is independent of queries and fact management. Isomorphism offers several benefits, including easier maintenance of legal models. Since laws change over time, maintaining structural alignment between code and legal text simplifies updates.

Another essential requirement is defeasibility, acknowledging that legal rules are

inherently conditional and subject to exceptions. A rule is valid unless contrary evidence, principles, or higher-ranking norms dictate otherwise [33].

Explainability is crucial for ensuring AI-driven legal decisions remain understandable to all stakeholders. Justifications must be based on legal principles, statutory norms, and precedents (particularly in common law). AI systems must account for open-ended legal definitions and discretionary judicial interpretations.

If an automated system can transparently explain its reasoning, trust in its outputs increases [67]. It is important to distinguish between explanation (the system’s provided rationale) and interpretation (the user’s ability to understand it). Different users require different types of explainability, such as:

- Case-based explainability: Suitable for precedent-driven systems, offering insights by referencing previous court decisions.
- Rule-based explainability: Ensures clarity and reduces misinterpretations, though it may lack the contextual nuance provided by case law [58].

For legal professionals, AI-generated rules must be easily accessible and editable. A judge should be able to modify a rule to reflect evolving legal interpretations and immediately assess its impact on a legal corpus. Ideally, the rule itself should serve as the explanation, ensuring judicial discretion remains intact.

This principle also supports legal drafting. When legislative amendments are proposed, their effects can be tested by applying them to known cases, improving the predictability and robustness of legal reforms.

The approaches discussed in this thesis aim to identify legal concepts and rules, translate them into computable languages, and maintain their connection to natural language sources. This ensures legal professionals can navigate and oversee AI-driven decision systems while preserving the integrity of the legal domain.

By linking computable code to legal texts, AI-generated explanations become inherently aligned with legal principles, eliminating unnecessary translation layers. An explanation is truly interpretable when it conveys insights that are readily understandable to legal practitioners and citizens alike [48].

2.3 ML Based Approaches

This section presents a review of recent research exploring the interaction between expert systems and generative artificial intelligence, specifically emphasizing Large Language Models. It highlights the application of these models in the development and enhancement of expert systems.

In particular, state-of-the-art of prompt engineering is developing input approaches capable of overcoming the notorious limitations of Large Language Models by giving such models reasoning steps, context and examples. Research has developed the Chain of Thought (CoT) approach[46] around this idea, and has continued to create tangent methodologies for prompt engineering based on logical inputs and contextual references, such as those seen in augmented legal information retrieval and generation[40], as well as for the successful completion of legal reasoning tasks[45]. All of the above employ the concept of chain as an approach which creates causal steps and context guiding the output of the model towards a more reliable, accurate and sound response.

One of the key studies, [42], explores the promising potential of large language models (LLMs) in automating the extraction of structured legal representations. The work emphasizes how LLMs can support the development of legal expert systems, ultimately contributing to improved access to justice. The study demonstrates a positive correlation between pathways generated by LLMs and manually crafted ones for legal decision support systems. This finding underlines how LLMs can streamline the creation of symbolic legal systems, making them more efficient and scalable.

Similarly, [10] highlights the significant role of LLMs in enhancing the accessibility of rule-based legal systems. The research introduces the "Chain of Prompts" methodology, which focuses on translating the explanations produced by rule-based systems from programming languages into natural language. This approach empowers non-experts to independently perform complex legal tasks, including comparing different inferences drawn from rule-based systems. By bridging the gap between legal rules encoded in formal languages and the understanding of these rules by non-professionals, this methodology can help democratize access to legal reasoning and decision-making.

In the context of addressing the knowledge representation challenge, [62] and [28] examine how Natural Language Processing (NLP) and LLMs can be used to transform legal language into machine-readable rules. These studies focus on the complexities involved in converting legal documents into formats that can be processed by machines, and they highlight both the challenges and opportunities that come with automating the formalization of legal knowledge. By enabling the automated translation of legal text into structured rules, LLMs could significantly contribute to the development of intelligent systems capable of supporting legal analysis and decision-making.

The use of LLMs for information extraction tasks has been explored from various perspectives in the literature. These works explore how LLMs can be applied to a range of data extraction challenges, laying the foundation for future research aimed at refining and adapting these models for specific legal contexts. This exploration reflects the broad potential of LLMs in transforming legal processes, especially in terms of automating data extraction and knowledge formalization, which are central to the development of AI-powered legal systems.

Bellan et al [6] investigated the feasibility of using LLMs to extract process knowledge from business process descriptions and transform them into KGs. The aim of the study is to determine how effectively LLMs, in particular GPT-3, can support this task, especially in resource-constrained environments. The authors use a prompt-based, in-context learning strategy in a multi-turn dialogue to extract conceptual information such as activities, actors and their temporal relationships from natural language documents. The evaluation includes different prompt settings, KG construction strategies and textual output evaluations. The results show that minimal and high quality prompts improve process knowledge extraction, while incremental KG construction shows moderate effectiveness. While injecting domain context has a limited impact, the study highlights the potential of LLMs for extracting structured knowledge from unstructured text, paving the way for future research to refine these techniques.

Dong and Wu [25] addressed the challenge of improving Named Entity Recognition (NER) in the context of safety hazard analysis, particularly in industrial environments. The aim was to improve the accuracy of NER by integrating Graph-Based Dependency (GBD) models with LLMs. The method involved applying

GBD to establish relationships between entities and using the contextual understanding of LLMs to refine detection accuracy. The results showed a significant improvement in precision and recall metrics compared to traditional NER approaches, highlighting the synergy between GBD and LLMs. The discussion highlighted that this hybrid approach improves entity recognition, which is crucial for the analysis of security-related textual data. The conclusions suggest that this method has strong potential for broader applications in security-critical environments and could be extended to other domains requiring accurate entity identification.

Zhao, Chen and You [86] proposed a novel method called LlmRe for zero-shot entity relation extraction using large language models. The aim is to overcome the limitations of traditional entity relation extraction methods, which require a large amount of labelled data and customised model structures. The method exploits the generalisation ability and parameter knowledge of large language models, such as GPT-3 and ChatGPT, to perform entity relation extraction without fine-tuning. In-context learning is introduced to guide the output of the model in a specified format, reducing the randomness of language model generation. A three-stage extraction framework is proposed to decompose the task into head entity extraction, tail entity extraction and relation inference, reducing complexity. The method is evaluated on three self-generated datasets in different domains, and experimental results demonstrate its effectiveness and domain adaptability. The proposed method outperforms comparative models on several metrics, and achieves impressive performance in zero-shot entity relationship extraction. The paper concludes that LlmRe effectively meets complex extraction requirements, reduces dependency on labelled data, and has good domain adaptability, making it suitable for practical application scenarios such as knowledge graph construction and intelligent question answering.

De Bellis [22] explored the potential of LLMs to model external semantic knowledge about concepts from unstructured text, focusing on their latent representations. The aim is to assess the ability of LLMs to encode and extract factual and ontological knowledge, and to determine whether this knowledge matches KG ground truth. The method involves analysing LLM latent spaces for patterns of semantic grounding in structured open-domain knowledge bases, using techniques such as linear probing and clustering. The results show that LLMs can capture

large amounts of factual knowledge, and that their latent spaces reveal patterns of semantic grounding in knowledge bases. The paper discusses the potential of exploiting these patterns for complex downstream tasks such as KG completion and semantic tagging, and highlights the importance of understanding where and how this knowledge is stored in LLMs. The paper concludes by emphasising the need for further research to fully interpret LLMs and their potential in various Semantic Web applications.

In the field of legal research, the use of LLMs has been applied to various tasks such as summarising, question and answering, etc. [84, 50, 61, 20, 34] However, there is a few research focusing on information extraction from legal documents using LLMs.

Shi et al [77] presented Legal-LM, a knowledge graph-enhanced LLM tailored for the Chinese legal domain. They aimed to address key challenges in legal AI, including domain-specific adaptation, data veracity, and providing legal advice to non-professional users. To achieve this, the authors pre-trained the model using extensive legal corpora, applied Direct Preference Optimization (DPO) to generate varied and accurate legal advice, and integrated an external legal knowledge base with soft prompts to retrieve relevant information. They conducted experiments using both objective and subjective legal datasets to evaluate Legal-LM’s performance in answering legal questions, analysing cases and making recommendations. The results showed that the model outperformed baseline models in terms of accuracy, completeness, clarity and language quality. The authors concluded that Legal-LM has the potential to significantly improve legal advice and education in the Chinese legal domain.

The integration of large language models into expert systems has also gained significant attention in recent research, with a focus on improving legal reasoning, decision-making processes, and knowledge extraction. These studies explore how LLMs can be harnessed to enhance the functionality of expert systems, addressing challenges such as the automatic extraction of legal information and facilitating better operational decision support.

One of the pioneering experimental studies is by [63], which explores the use of LLM interfaces for expert systems. The study focuses on the automatic extraction and identification of legally relevant symbolic elements from natural language,

demonstrating how LLMs can assist in processing legal information. Following this, [31] investigates the role of GPT-3 in question-answering tasks related to operational decisions. The findings suggest that while GPT-3 excels at identifying appropriate reasoning mechanisms, the generation of explanations is more effectively achieved by integrating GPT-3 with DMN (Decision Model and Notation) chatbots. This integration significantly enhances the quality of support for operational decision-making. Other studies have primarily addressed the knowledge representation bottleneck, which remains a critical challenge in fully leveraging the potential of expert systems.

The powerful language capabilities of LLMs have also opened up new avenues for developing chatbots across various fields and user needs, with a growing focus on their application in the legal domain. These chatbots are increasingly used to improve access to justice, offering individuals easier ways to obtain legal information and assistance.

For example, [70] introduces two chatbot systems designed to provide users with legal information. One chatbot addresses immigration-related questions using publicly available data, while the other assists financial institution employees by answering job-specific legal FAQs. [3] further suggests the integration of AI-powered chatbots with retrieval-augmented generation (RAG) technology to offer more personalized and context-specific legal information.

Additionally [83] presents the JusticeBot methodology, which combines rule-based and case-based reasoning to guide users through legal questions.

2.4 Legal Background

The **right to explanation** as proposed by the EU AI Act (AIA) is intended to provide individuals with a clear understanding of decisions made by high-risk AI systems that significantly impact them. This right builds upon principles found in existing regulations like Article 22 of the General Data Protection Regulation (GDPR), which grants individuals protection against solely automated decisions with legal or similarly significant effects. However, the lack of a uniform definition between these two laws introduces complexities. Article 22 of the GDPR implies a more general right for individuals to contest fully automated decisions, while the

AIA specifically requires deployers of high-risk AI systems to explain the role of AI in such decisions, the key parameters involved, and the input data used. This divergence highlights a gap between GDPR’s broader protection and the AIA’s more targeted focus on AI deployment.

The interplay between Article 22 GDPR[30] and Article 68 of the AI Act reflects the ongoing challenge of ensuring transparency in automated decision-making. Under the GDPR, individuals have a general right to avoid being subject to automated decisions without human involvement. Meanwhile, the AIA introduces a more specialized duty for providers of AI systems to give explanations about high-risk decisions that affect people’s rights, health, or well-being. This creates the potential for inconsistency in how organizations respond to requests for explanations[60]. A system considered low-risk under the AIA might still trigger GDPR requirements if it makes a solely automated decision, leaving entities to navigate these overlapping obligations without a unified standard for what constitutes a sufficient explanation.

Given this lack of uniformity, organizations face the challenge of extracting meaningful information from AI systems to meet regulatory requirements. Merely providing technical details or data inputs may not suffice if the explanation does not help individuals understand why a decision was made and how it impacts them. The goal is to make AI decision-making processes comprehensible, bridging the information gap between AI systems and individuals. This calls for developing frameworks or standards that translate complex AI functions into user-friendly explanations, enabling affected individuals to assess the fairness and legality of the decisions made about them.

In recent years, the field of explainability and explainable artificial intelligence (XAI) has gained significant attention due to the growing use of AI systems in various aspects of life, particularly with the rise of opaque ”black box” models. Researchers have developed popular XAI methods like LIME and SHAP to address the need for transparency, yet the concept of explainability itself remains highly contested. Different scholars have proposed varied interpretations of what constitutes an explanation, leading to numerous taxonomies. This ambiguity reflects the multifaceted nature of explainability, especially as it intersects with legal concerns, including those in automated decision-making (ADM) in EU administration. Legal

scholars have begun exploring the implications of explainability within the context of AI’s impact on governance, pioneering analysis of these intersections. However, more research is needed to fully map out how specific XAI techniques align with or diverge from legal requirements.

The challenge lies in bridging the gap between the legal and technical dimensions of explainability, particularly in relation to regulations like the General Data Protection Regulation (GDPR) and the forthcoming AI Act. These legal frameworks impose requirements such as transparency, accountability, fairness, and privacy, which XAI methods aim to support. Yet, current XAI methods do not always fulfill these expectations seamlessly. For example, while legal requirements might focus on providing clear justifications for decisions to ensure fairness and non-discrimination, technical methods like LIME or SHAP focus primarily on making AI outputs interpretable to technical audiences. This dissonance highlights the complexity of ensuring that explanations provided by XAI align with legal standards and societal expectations, where understandability varies depending on the audience, such as AI experts versus lay users or legal entities.

A key tension arises because neither legal nor technical approaches fully encapsulate the intent of regulations like the GDPR or the AI Act. These frameworks require AI systems to provide meaningful explanations that ensure individuals’ rights to understand how decisions that affect them are made, yet this “meaningfulness” can be subjective. Technical explainability might focus on providing information about an AI model’s inner workings, while the legal domain demands that this information be presented in a way that non-expert stakeholders can understand. This multidimensional nature of explainability requires a holistic approach, where legal mandates on fairness and transparency are carefully integrated with the capabilities of XAI methods to meet the requirements of different stakeholders effectively.

Furthermore, the AI Act in Recital 38, notes that a lack of transparency, explainability, and documentation could hinder the exercise of procedural fundamental rights such as the right to a fair trial and effective remedy.

Transparency, frequently mentioned in the AI Act, is distinct from explainability. For example, Article 14(1) mandates that high-risk AI systems should be designed to allow for effective human oversight through appropriate human-

machine interfaces. This human oversight is essential to prevent or minimize risks to safety and fundamental rights, even when AI systems are used as intended or in cases of foreseeable misuse. Such oversight requires users to have a comprehensive understanding of the AI system’s outputs and functioning, emphasizing the need for clear interfaces to facilitate user interaction and control.

The Act also outlines several transparency-related obligations for AI providers, such as maintaining technical documentation, logging occurrences, and implementing risk management systems. Although these measures don’t guarantee full transparency or explainability, they support empirical verification of AI operations and help in identifying malfunctions. By providing interfaces that enable users to interact with and oversee AI systems effectively, these provisions contribute to a certain level of transparency and allow users to better understand AI behavior, fostering trust and accountability in AI decision-making.

Chapter 3

Legal Rule Modelling

3.1 Rule Modelling

In our increasingly interconnected world, the enforcement and application of laws require consideration of multiple overlapping legal systems. A structured legal representation of rules can assist professionals in navigating the growing complexity of legal frameworks.

The computable representation of legal norms has been an area of study since the 1970s and 1980s, with logic programming marking the first major breakthrough. Modern research focuses on structured rule-based systems, alternative reasoning models, and argumentation frameworks.

Regarding explainability, we will explore the legal information that can be conveyed to users and how it can be effectively presented.

3.1.1 Expert System for European and National Criminal Law

This chapter introduces an expert system designed to represent European and national legal sources in criminal law. The system aims to address legal questions such as applicable rights, the validity of legal sources, and the degree of alignment between European and national legislation.

The CrossJustice project focuses on EU and national legal frameworks, partic-

ularly in criminal law. Given the multilevel nature of these systems, identifying applicable legal provisions is complex due to overlapping international conventions, bilateral agreements, and national instruments. A careful analysis of legal subordination, as well as the temporal and territorial scope of laws, is essential.

Additionally, rulings from the EU Court of Justice and national courts contribute to the legal landscape, though at varying speeds depending on national contexts. The global nature of today's digital society introduces further legal challenges, requiring a nuanced approach to legal interpretation.

Legal cases often involve multiple normative systems, which can lead to conflicts or differing outcomes. This complexity affects both national and transnational legal systems, making it difficult for legal professionals to determine which sources to consult. A comprehensive approach is necessary to ensure proper legal reasoning.

CrossJustice examines defendants' rights across various EU Member States and provides a decision-support system for both professionals and citizens. The system assesses individual cases and evaluates the alignment of national laws with EU legislation, determining the extent of harmonization with the EU acquis.

The goal is to support legal practitioners by offering a platform that enhances interoperability and communication among national legal frameworks. It clarifies how different laws interact and assesses compliance with EU directives.

The project focuses on building a rule-based expert system based on computable representations of key European directives related to the rights of suspects and accused persons in criminal proceedings. These directives include:

- Directive (EU) 2016/343 of the European Parliament and of the Council of 9 March 2016 on the strengthening of certain aspects of the presumption of innocence and of the right to be present at the trial in criminal proceedings.
- Directive 2010/64/EU of the European Parliament and of the Council of 20 October 2010 on the right to interpretation and translation in criminal proceedings.
- Directive (EU) 2016/800 of the European Parliament and of the Council of 11 May 2016 on procedural safeguards for children who are suspects or accused persons in criminal proceedings.

- Directive (EU) 2016/1919 of the European Parliament and of the Council of 26 October 2016 on legal aid for suspects and accused persons in criminal proceedings and for requested persons in European arrest warrant proceedings.
- Directive 2012/13/EU of the European Parliament and of the Council of 22 May 2012 on the right to information in criminal proceedings.
- Directive 2013/48/EU of the European Parliament and of the Council of 22 October 2013 on the right of access to a lawyer in criminal proceedings and on the right to have a third party informed upon deprivation of liberty and to communicate with third persons and with consular authorities while deprived of liberty.

All relevant provisions from these directives and their national transpositions have been encoded in Prolog. Legal experts have analyzed complex rules and dependencies between norms, ensuring that the system remains interpretable by both legal professionals and developers while enabling traceability of reasoning processes.

Unlike direct legislative transposition, EU directives require adaptation to fit the legal structures of individual Member States. Their purpose is not to impose uniform solutions but to ensure compliance while allowing national legislators discretion in implementation. This flexibility can lead to discrepancies, resulting in legal inconsistencies.

The system serves two primary purposes: first, to determine the existence and applicability of rights within national legal frameworks; second, to analyze the relationship between EU directives and their national implementations. It identifies deviations from the directives and highlights potential violations.

Effective legal reasoning requires defining key requirements from both legal and technical perspectives, ensuring accurate interpretation and application of laws.

3.2 Legal Principles

When modeling various legal sources, several key legal requirements have been observed to ensure accuracy, consistency, and effective representation of legal norms. These requirements are outlined as follows:

3.2.1 Isomorphism

Isomorphism refers to the one-to-one correspondence between legal norms in their formal model and their natural language representation. Karpf[43], as cited in Bench-Capon and Coenen, outlines five fundamental rules:

1. Each legal source is represented separately.
2. The structure of each legal source is preserved in the representation.
3. Mutual relationships, references, and connections between legal sources are maintained.
4. The representation of legal sources and their relationships is distinct from other model components, such as query processing and fact management.
5. If procedural law is included, both material and procedural rules are represented, ensuring that the system functions in accordance with procedural order.

Adhering to these principles supports the verification, validation, and maintainability of well-engineered knowledge systems.

3.2.2 Reification

Legal norms must be treated as objects with properties to handle various legal aspects effectively, including:

- **Jurisdiction:** Defining the boundaries within which rules are authoritative and binding.

- **Authority:** Identifying the source of a rule and its hierarchical status within legal frameworks (e.g., constitutional law, statutory law, regulations).
- **Temporal Properties:** Legal rules often include temporal aspects:
 - *External time:* The period during which a norm is part of the legal system, covering validity, annulment (retroactive effect), and abrogation (prospective effect).
 - *Internal time:* The timeframe during which a norm applies, determining when conditions must be met for its effects to be realized.

3.2.3 Defeasibility

Legal rules can be presumptive but subject to exceptions. Key aspects of defeasibility include:

- **Conflict Resolution:** Managing rule conflicts using principles like:
 - *Lex specialis:* Preference for the more specific rule.
 - *Lex superior:* Preference for rules from higher authorities.
 - *Lex posterior:* Preference for newer rules over older ones.
- **Exclusionary Rules:** Explicitly overriding or invalidating other rules.
- **Handling Vagueness:** Some legal terms lack precise definitions, requiring interpretative flexibility. For example, Directive 2010/64 (Art. 3) mentions a "reasonable period of time" and "sufficient quality," both of which are context-dependent.
- **Normative Effects:** The persistence of obligations, permissions, and rights over time.
- **Qualificatory Rules:** Assigning legal status (e.g., "X is an EU citizen if X is an Italian citizen").
- **Definitional/Constitutive Rules:** Establishing legal concepts.

- **Deontic Rules:** Imposing obligations or granting permissions.
- **Potestative Rules:** Assigning legal power.
- **Evidentiary Rules:** Determining conclusions from evidence, including legal presumptions.

3.3 Technical Requirements for a Reasoning System

3.3.1 Tractability and Computational Complexity

Ensuring that the system can handle legal reasoning efficiently is crucial. Problems should be divided into manageable subproblems to avoid intractability. The computational complexity also depends on the language used for the reasoning engine, which will be analyzed in further detail.

3.3.2 Procedural Order of Inference Rules

Legal reasoning systems involve two main types of rules:

- **Knowledge Rules (Declarative):** Define facts and relationships (e.g., "If a person does not understand a language and is on trial in that language, they have a right to interpretation").
- **Inference Rules (Procedural):** Guide decision-making (e.g., "If there is an exception to a rule, the rule is invalid"). These rules help filter invalid rules and optimize reasoning processes.

3.3.3 Semantic Values of Expressions

Legal reasoning involves different truth values beyond "true" and "false," such as:

- **Decision-making under certainty:** Assumes complete knowledge.
- **Decision-making under uncertainty:** Deals with unknown or uncertain values, requiring abductive reasoning.

- **Engine Customization:** Some reasoning engines offer built-in methods to manage uncertainty.

3.3.4 Control Strategies

Implementing a rule-based system requires effective rule execution strategies, including:

- **Rule Chaining:** Determining the order in which rules are applied:
 - *Forward chaining:* Starts from facts and derives conclusions.
 - *Backward chaining:* Starts from a goal and determines supporting facts.
 - *Mixed strategy:* Uses both approaches as needed.
- **Conflict Resolution:** Handling competing rules and interactions through meta-rules.

3.3.5 Explanation and Justification

A legal reasoning system must be able to explain its decisions. Two primary methods exist:

- **Static Explanation:** Predefined textual responses linked to questions. While simple, this method can become inconsistent over time.
- **Dynamic Explanation:** Reconstructs justifications based on rule execution patterns, providing more accurate and context-aware explanations.

Additionally, a well-designed decision report should outline the rationale behind conclusions in an understandable manner, ensuring transparency and interpretability.

3.4 Conclusion

The legal requirements outlined above help shape the technical foundations of reasoning systems for legal applications. By ensuring isomorphism, reification, defeasibility, computational efficiency, procedural structuring, semantic clarity, rule

control, and justification mechanisms, these systems can provide reliable and interpretable legal reasoning.

3.4.1 Criminal Procedure Law Formalized through Logical Representation

As previously outlined, the CrossJustice project focuses on analyzing the rights of defendants in criminal proceedings across multiple EU Member States. It provides a decision-support system designed for both legal professionals and citizens, offering case-specific evaluations and assessing the extent to which national laws harmonize with EU law. Specifically, the project evaluates the alignment of national legal frameworks and regulatory measures with the EU *acquis*, as well as the relevant legislative acts of the European Union.

The primary aim of CrossJustice is to assist legal practitioners by providing a platform that fosters interoperability and communication among various national legislative frameworks. By demonstrating how different legal provisions interact and examining their compliance with EU law, the system aids professionals in navigating the complexities of legal interpretation, ensuring that cases are handled consistently across jurisdictions.

A central objective of the project is the development of a rule-based expert system that incorporates a computable representation of six key EU directives related to the rights of suspects and accused persons in criminal procedures. These directives serve as foundational legal instruments that guide Member States in providing fair treatment during criminal proceedings.

The system ensures that users are fully informed of all applicable rights under both European and national legal systems. It also includes representations of the national transpositions of these directives within four EU countries: Italy, the Netherlands, Poland, and Spain. This feature enables a comparative analysis of legal sources, facilitating the identification of potential conflicts or inconsistencies in how national laws have implemented these directives.

The relationship between EU directives and national laws is intricate and multifaceted. Unlike EU regulations, which are directly applicable in all Member States, directives require Member States to establish legal frameworks that align with the

principles set out by the EU, while allowing for national adaptations based on local legal systems and traditions. Although Member States are constrained in their discretion when transposing these provisions, variations in how these directives are implemented can result in discrepancies that occasionally lead to legal challenges, especially in cases involving multiple jurisdictions.

Such differences often arise in the application of laws and may stem from:

- Variations in terminology, where different legal concepts may be referenced, and
- The inclusion or omission of certain legal requirements in national legislation.

Standardization is crucial when modeling legal norms. To ensure consistency, we focused on foundational legal concepts and structured computable rules around them. We carefully selected lexical terms to maintain a close correspondence between legal concepts, their attributes, and relationships, as reflected in the source texts. This approach builds on an ontology developed in a previous project related to private international law.

In constructing the logical framework, each lexical term was formalized as distinct object types within the legal rules—such as predicates with specific properties, functions, and constants. Following this methodology, we identified three primary legal concepts: Person, Claim, and Contract. For each concept, we determined relevant attributes and assigned a unique identifier to each object. This enables differentiation between instances of the same entity (e.g., two different persons) and facilitates connections between entities (e.g., a specific individual being a defendant in a particular claim).

In the following section, we will examine the concept of ‘Person’ and its associated attributes.

The Person Concept

Nature (Legal/Natural)

Role (Plaintiff/ Defendant/Third Party)

Type (Consumer/Business/Employer/Employee/Insurer/Trust)

Work (Country)

ActivityIn (Country)

Domicile

3.4. CONCLUSION

Habitual Residence (Country)
(Country)
Establishment (Country)
Document (Document)

The *Person* concept the most reoccurring concept throughout the all european regulations, as there will always be at least a party in any given case. All predicates linked to the ‘Person’ concept will be written as the main concept followed by the attributes, properties and relations that are linked to it. Each predicate will also contain an ID needed to identify the person who is the subject of the case, followed by the arguments related to the values attributed to that property. Let us take a look at an example that shows the process of building a predicate.

```
%% brusselsRegulationApplies(ClaimId, _brusselsRegulation)
%
% Article 1.1
%
% This Regulation shall apply in civil and commercial
% matters whatever the nature of the court or tribunal. It
% shall
% not extend, in particular, to revenue, customs or
% administrative
% matters or to the liability of the State for acts and
% omissions in
% the exercise of State authority (acta iure imperii).
% It shall not apply, in particular, to revenue, customs
% or administrative matters or to the liability of the State
% for acts
% and omissions in the exercise of State authority (acta
% iure imperii).

brusselsRegulationApplies(ClaimId, brusselsRegulation):-
    \+ exception(brusselsRegulationApplies(ClaimId,
        brusselsRegulation), _),
    claimMatter(ClaimId, civilCommercial).
```



```
claimMatter(ClaimId, civilCommercial):-
    claimMatter(ClaimId, civil).

exception(brusselsRegulationApplies(ClaimId,
    brusselsRegulation), _):-
    claimMatter(ClaimId, revenue)
;    claimMatter(ClaimId, customs)
;    claimMatter(ClaimId, administrative).

exception(brusselsRegulationApplies(ClaimId,
    brusselsRegulation), actsOrOmissions):-
    actsOrOmissions.
```

Since a full analysis of the rule is redundant for our purpose right now, we will only focus on use of the concept person. `PersonId` is written with a capital letter, as it is a variable which stands in the place where we will input the subject's name when needed, while words like 'insurer' and 'defendant' are constants which assert that a person has that unique role in the proceeding. In this rule there are two predicates that are part of the person concept. `PersonRole` is used to specify the role of the subject, while `personType` is used to identify the qualification of that subject depending on the section that must be applied. While the role of a person in the proceedings can only be the defendant, claimant or plaintiff, or a third party, we use `personType` to identify a consumer, a professional, all people involved in trusts (e.g. founder, trustee, beneficiary), employer and employees, tenants and landlords, etc. Building a rule through the repetitive use of a concept such as 'Person' helps both the user and the programmer have a clearer understanding of the computable rule, while keeping as much of the original wording, thus having a higher degree of isomorphism. Let us now consider the Claim concept, its attributes, properties and relations.

```
The Claim Concept
Matter    (Civil/Commercial)
Grade     (First/ Appeal/)
Type      (Original/Counter/Incidental/)
```

```
Object (Contract/Tort/Ownership/Rights in rem/Liability/
      Trust/)
Seised (Country/Court)
```

We've already seen in previous examples how `claim` has been used in order to identify the matter of the proceedings (i.e. `brusselsRegulationApplies`). The `Claim` concept finds his most use when listing exceptions and special jurisdiction rules (Section 2). It is a fast way to identify the relevant section of the Brussels Regulation and immediately apply the correct rule.

For example, Article 7(2) states that a person domiciled in one Member State may be sued in another Member State for matters related to tort, delict, or quasi-delict, in the courts located where the harmful event occurred or is likely to occur.

The Prolog representation is the following:

```
hasJurisdiction7_2(article7_2, Country, Court, ClaimId,
    brusselsRegulation):-
    claimObject(ClaimId, tort),
    eventOccurredOrMay(ClaimId, Country, Court).
```

Just as with the `Person` concept, Predicate names result from the combination of the main concept, i.e. `claim`, and its attributes, properties and relations. Each predicate contains an ID representing the unique object, i.e. the specific claim, the values related to that property and the ID of a different object whenever it is necessary to identify a particular relation (e.g. `claimObject` may contain both the `ClaimId` and the `ContractId` relevant in the specific claim.).

For example, according to EU Directive 2013/48, every accused individual has the right to consult a lawyer. However, the Dutch transposition of this directive stipulates that this right is guaranteed only for individuals who are deprived of their liberty, under the assumption that those who are not subject to such restrictions are already free to consult a lawyer without needing additional protection.

```
%% has_right(_art3_1, PersonId, _right_to_translation,
    _document_translation)
%has_right(art3_1, PersonId, right_to_translation,
    essentialDocument):-
%    proceeding_language(PersonId, Language),
%    essential_document(_, PersonId, documents),
```

3.4. CONCLUSION

```
%    \+ person_understands(PersonId, Language).

%% essential_document(_art, PersonId, _documents)
%
% Article 3.2
%
% Essential documents shall include any decision depriving a
% person of his liberty, any charge or indictment, and any
% judgment.
%essential_document(art3_2, PersonId, documents):-
%    person_document(PersonId, document_deprives_liberty).

[...]

%% essential_document(_art3_3, PersonId, _documents):
%
% Article 3.3
%
% The competent authorities shall, in any given case, decide
% whether any other document is essential. Suspected or
%   accused
% persons or their legal counsel may submit a reasoned
%   request to
% that effect.
%essential_document(art3_3, PersonId, documents):-
%    authority_decision(PersonId, essential_document),
%    person_request_submitted(PersonId, essential_document).
```

This is not the only challenge in modeling such laws. Another common difficulty in the computable representation of legal norms is how to handle open-ended clauses without unduly narrowing their scope.

For instance, Article 3, paragraph 1, of Directive 2010/64/EU states that suspected or accused persons must receive a written translation of all documents essential to ensuring their right of defense and maintaining the fairness of proceedings. However, the legislator does not define what qualifies as an "essential

document,” instead leaving this determination to the judge, who must assess it on a case-by-case basis depending on the circumstances, outside of certain cases.

This flexibility is further illustrated in the second and third paragraphs of the same article. The second paragraph provides examples of what may constitute an essential document, while the third explicitly states that the final decision rests with the judge.

To transpose this article into a computable framework, we have chosen to encode both the examples provided by the directive and a general rule allowing the judge to designate any document as essential to the proceedings. In our codebase, we use a single predicate, *essential_document*, to indicate that a document has been deemed essential. This predicate can be instantiated either by applying the examples listed in the second paragraph or by relying on the judicial discretion outlined in the third.

The second paragraph is represented by a predefined list of documents that meet the directive’s criteria, while the third paragraph is modeled through a predicate, *authority_decision*, which signifies that the judiciary has already made a ruling affirming a document’s essential nature. In this way, our system determines whether a document falls within the scope of the directive either by referencing established examples or by acknowledging a judicial decision.

Additionally, the system records whether the determination was made under the second or third paragraph and includes this information in the final output as a reference to the relevant article. This enhances explainability and transparency, helping users understand why a particular document has been classified as essential.

Importantly, this approach does not constrain judicial discretion. The system either applies one of the directive’s predefined examples or relies on an explicit decision from a legal authority. If no applicable article were available to support a rule’s application, the final determination would remain with the judge, and our system would require judicial input before confirming a positive outcome.

```
%% has_right(_art37_1, PersonId, _right_to_be_visited,
    _relatives)
%
% Article 37(1) Decree no 230 of the President of the
```

3.4. CONCLUSION

```
Republic of 30 June 2000
%
% 1. Visits of convicted persons, inmates and defendants
% after the verdict of the first stage
% are authorized by the Head of the institute. Visits with
% persons different from relatives and
% cohabitees are authorized when sensible reasons occur.
%has_right(art37_1, PersonId, right_to_be_visited, relatives
% ) :-
%     (person_status(PersonId, convicted);
%     person_status(PersonId, inmate)),
%     authority_decision(PersonId, authorized).
```

As an example, consider Article 37, paragraph 1, of Decree No. 230 of the President of the Republic in the Italian legal system. This provision states that any visit to a convicted person requires authorization from the Head of the Institute. However, since the legislator does not specify the conditions under which such authorization should be granted and instead delegates the decision to the relevant authority, we have chosen to represent this rule using the predicate *authority_decision* to indicate that authorization has been granted.

Furthermore, the system does not impose any specific interpretation regarding the scope of relevant documents. The final part of Article 3, paragraph 1, of Directive 2010/64/EU states that such documents must be necessary to ensure the accused can exercise their right of defense and to uphold the fairness of proceedings. This introduces an element of vagueness, as the legislator does not explicitly define what constitutes fairness in this context.

This issue of vagueness extends to broader legal concepts that are not clearly defined within the legal framework, relying instead on general consensus. Terms such as "good character" or "reasonable amount of time" represent subjective conditions that the system cannot interpret directly. To address this, we have opted to disregard these vague criteria, effectively treating them as inherently true.

An alternative approach could have been to introduce an additional condition requiring judicial confirmation whenever a rule necessitates ensuring the fairness of

3.4. CONCLUSION

proceedings. However, we determined that such a requirement would lack sufficient informational value to be meaningfully incorporated into the system.

```
%% has_right(_art4, PersonId, _right_to_legal_aid, _free)
%
% Article 4
% Legal aid in criminal proceedings
%
% 1. Member States shall ensure that suspects and accused
    persons who lack sufficient resources to pay for the
% assistance of a lawyer have the right to legal aid when
    the interests of justice so require.
%% 2. Member States may apply a means test, a merits test,
    or both to determine whether legal aid is to be granted
    in accordance with paragraph 1.
has_right(art4, PersonId, right_to_legal_aid, free) :-
    (    person_status(PersonId, suspect)
      ;   person_status(PersonId, accused)
    ),
    authority_decision(PersonId, lack_resources).

authority_decision(PersonId, lack_resources) :-
    proceeding_matter(PersonId, means_test);
    proceeding_matter(PersonId, merits_test).
```

Another challenge we encountered in our transposition, closely related to the previous issue, concerns cases where the European legislator has not established a minimum standard for applying a norm. Since EU guidelines are not directly applicable and require implementation at the national level, any instance where the European legislator provides no concrete requirements necessitates an alternative approach. In such cases, we have opted to use the *authority_decision* predicate, signifying that the final determination—both in form and substance—is left to the judge or another legal authority.

For example, consider the transposition of Article 4 of Directive (EU) 2016/1919, which establishes the right to legal aid. Our transposition explicitly includes para-

3.4. CONCLUSION

graphs 1 and 2, which state that a suspect or accused person is entitled to legal aid if the legal authority has conducted a means or merits test, confirming that the person lacks the financial resources to pay for legal representation. However, paragraph 3 does not specify concrete requirements and instead introduces broad concepts that require further national implementation. As a result, our transposition does not define what constitutes a means or merits test, as that determination is left to national legislation.

The situation varies significantly across Member States, as national laws provide more detailed conditions for accessing legal aid, requiring their translation into a computable format. For instance, under Article 78(1) of the Polish Code of Criminal Procedure, any person who cannot afford legal costs may request the appointment of a defense counsel. In this case, we represented the article with two distinct requirements: (1) the individual must be unable to bear the costs, and (2) they must formally request the appointment of a lawyer.

In contrast, both the Dutch (Article 28 of the Legal Aid Act) and Italian (Article 76 of Presidential Decree No. 115/2002) legislations impose more specific and practical conditions for accessing legal aid. These laws stipulate that eligibility is contingent on the individual's income falling below a certain threshold, which is set by the respective national provisions. As such, we have defined the predicate *person_income(PersonId, X)*, where X represents the income limit established by the relevant legislation.

In this scenario, there is no need to introduce an additional predicate requiring legal authority confirmation, as this step is already implicitly required in all proceedings. Furthermore, based on a literal interpretation of this article, the judge's discretion is limited to assessing the validity of the individual's financial status and approving the request for legal aid.

```
%% has_right(_art78_1, PersonId, _right_to_legal_aid,
    _appoint_ex_officio)
%
% Article 78(1-1a) code of criminal procedure
%
% 1. An accused, who does not have a defence counsel of his
    own choice, may request the appointment of % a defence
```

3.4. CONCLUSION

```
counsel ex officio ,
% if he can duly prove that he is unable to bear the costs
  of defence without prejudice to the necessary      %
  maintenance of himself, or his family.
has_right(art78_1, PersonId, right_to_legal_aid,
  appoint_ex_officio) :-
  person_status(PersonId, cannot_bear_costs),
  person_request_submitted(PersonId, appoint_lawyer).

%% art34Applies(PersonId)
%
% Article 34 legal aid act
%
% [as of 1 January 2020: 39400 Euros].
art34Applies(PersonId) :-
  person_event(PersonId, single_householder),
  person_income(PersonId, X),
  X < 27900.

art34Applies(PersonId) :-
  person_event(PersonId, share_householder),
  person_income(PersonId, X),
  X < 39400.

%% art76Applies(PersonId)
%
% Article 76 Presidential Decree no 115/2002
%
art76Applies(PersonId) :-
  person_income(PersonId, X),
  X < 11746.68.
```

The absence of a uniform minimum standard for providing legal aid across Member States results in varying scenarios, particularly when comparing national legislations. Efforts have been made to maintain the original structure as isomor-

3.4. CONCLUSION

phic as possible, while allowing the European transposition to remain an open-ended clause, subject to the discretion of the legal authority. As a result, this tool is valuable for both citizens seeking information about the law in a specific Member State and legal professionals interested in verifying the conformity of legal sources and their relationship to the original articles referenced in the Directive.

```
%% has_right(_art3_2, PersonId, _right_to_access_lawyer, _)

has_right(art3_2_a, PersonId, right_to_access_lawyer,
questioning) :-
    (    person_status(PersonId, suspect)
      ;   person_status(PersonId, accused)
    ),
    proceeding_matter(PersonId, questioning).

has_right(art3_2_b, PersonId, right_to_access_lawyer,
evidence_gathering_act) :-
    (    person_status(PersonId, suspect)
      ;   person_status(PersonId, accused)
    ),
    proceeding_matter(PersonId, evidence_gathering_act).

has_right(art3_2_c, PersonId, right_to_access_lawyer,
deprived_of_liberty) :-
    person_status(PersonId, deprived_of_liberty),
    \+ exception(has_right(art3_2_c, PersonId,
        right_to_access_lawyer, deprived_of_liberty), _).

has_right(art3_2_d, PersonId, right_to_access_lawyer,
summoned_court) :-
    proceeding_matter(PersonId, summoned_court).

%% has_right(_art5_1, PersonId, _right_to_information,
_inform_person)
```

```

has_right(art5_1, PersonId, right_to_inform, PersonId2) :-
    (    person_status(PersonId, suspect)
      ;    person_status(PersonId, accused)
    ),
    person_status(PersonId, deprived_of_liberty),
    person_nominate(PersonId, PersonId2),
    \+ exception(has_right(art5_1, PersonId, right_to_inform
        , PersonId2), _).

```

A critical feature of any legal reasoning system is the ability to represent the temporal sequence of events. As illustrated in the representation of Article 3(2) of Directive 2013/48, which states that suspects or accused individuals must be granted access to a lawyer without unnecessary delay. They are also entitled to consult a lawyer at the earliest opportunity, which includes before being questioned by the police or any other law enforcement or judicial authority, when investigative or competent authorities conduct an evidence-gathering act, immediately following deprivation of liberty, or if they are summoned to appear before a criminal court, ensuring they have access to a lawyer well before their court appearance.

We should highlight that it is stated that a specific right should be granted at the earliest possible moment in time. However, this poses a challenge in the computable translation of such provisions, as legal events do not always follow a strictly chronological order. Instead, they may occur intermittently throughout the proceedings, and their presence or absence at different stages complicates the ability to establish a direct temporal link between them.

In the context of this challenge, the solution adopted in the system involves associating each legal article with the particular phase of the proceedings in which the right is granted. When a user queries the system, and more than one event has occurred, the same article may be applicable multiple times. The user then has the responsibility of determining which of these occurrences represents the earliest point in time and ensuring the correctness of the procedure accordingly. This approach allows for a more comprehensive understanding of the legislative intent by illustrating the points at which rights are granted, though it places the burden on the user to interpret the temporal sequence.

3.4. CONCLUSION

One advantage of this methodology is its ability to provide a clearer and more consistent overview of the solutions adopted by the legislator. Specifically, it allows for an easier comparison of national implementations of EU directives. For example, when multiple legal events occur and the user seeks to find a similar right in national sources of law, understanding when the right is granted in the Directive versus national law can reveal discrepancies in the transposition of the Directive. If the Directive yields two results while the national system returns only one, it could signal a failure to fully or correctly transpose the Directive.

Moreover, another critical issue arises from the absence of the requirement *without undue delay* as indicated in Article 3, letter c, and Article 5, paragraph 1, of the Directive. The article states that Member States must ensure that suspects or accused individuals who are deprived of their liberty have the right to have at least one person, such as a relative or employer, whom they have nominated, informed of their deprivation of liberty without delay, should they wish to exercise this right.

This element of *delay* is not captured in the current computable representation, as the system focuses on identifying which rights individuals are entitled to or have been denied. The notion that rights should be granted immediately or within an appropriate time frame is considered beyond the scope of the system’s reasoning. The system is built around the technicalities of the legal texts themselves, rather than incorporating broader procedural dynamics or judicial discretion.

While in practice a judge may exercise a considerable degree of control over the timing of proceedings, our system’s transposition is presently limited to the textual dictates of the written law. Thus, references to the “manner provided for by the law” are interpreted strictly within the confines of the article’s wording, excluding interpretations based on judicial practices or jurisprudence.

Another limitation of our current system lies in the non-inclusion of legal consequences for the violation of certain rights. For instance, Article 16(1) of the Polish Code of Criminal Procedure specifies that there are no adverse consequences for breaching a defendant’s right to be informed of their rights. This type of legal nuance is not addressed in our system, as its primary function is to notify users of their rights rather than the consequences that follow from their violation. The one notable exception to this principle is the right to seek remedy for a legal decision

when a right has been violated, but only where this is explicitly stipulated by law.

Looking toward the future, future chapters of this thesis will integrate accepted judicial opinions to address interpretative issues within the predicates in our code. The integration of legal practices, customs, and judicial traditions could provide a richer, more nuanced reasoning framework that accounts for the diverse interpretations of laws. Ultimately, this would lead to a system capable of processing multiple interpretations of a legal rule, using either logical transpositions or judicial opinions to guide the analysis.

We propose that such a system should remain continually updated with the relevant legal corpus. Efforts should be made to expand the knowledge base and adapt to evolving legal standards. By doing so, the system can continue to provide meaningful support to both citizens and legal professionals, ensuring that it stays relevant and effective in the face of changing legal landscapes.

3.4.2 Technical Setup

As previously mentioned, the CrossJustice rulebase is developed using SWI-Prolog, a logic programming language with which the development team is well-acquainted. SWI-Prolog offers numerous advantages: it is highly documented, readily available across various systems, and deeply extensible through both pre-built and custom tools, as we will demonstrate. The language is partially standard-compliant, allowing the integration of other Prolog implementations to enhance the system's functionality.

The decision was made to organize legal texts in a folder structure similar to the one used on the CrossJustice portal. The structure places the Directive in a single file and the corresponding national articles in separate files for each state. This structure may evolve in the future to also separate different national legal texts.

Key goals of the system can be described as entry points for various legal sources, and these are often common across different national implementations.

For example, Directive 2010/64/EU can be represented as follows:

```
directive_2010_64(Right, dir, Article, PersonId, Matter)
```

The elements represent the following:

- **Right:** This is a variable (denoted by `Right`) that would typically represent a particular legal right under the directive, such as the right to translation, interpretation, or other related legal rights provided by the directive.
- **dir:** This is identifier that represents a category within the directive, such as "directive" itself.
- **Article:** This variable refers to a specific article or section within Directive 2010/64/EU that is relevant to the given right.
- **PersonId:** This variable likely refers to the unique identifier for the person involved in a legal case, such as a defendant or an accused individual. It helps to identify which person the legal provision applies to in a particular scenario.
- **Matter:** This variable likely refers to the legal matter or case to which the directive and the rights under it are applied. It could represent the type of case or a specific aspect of the legal issue that is being addressed by the directive.

This straightforward change allows for clear distinctions between the various national documents, both in the code and in the logical structure. Each representation begins with the top-level goal, followed by exceptions and primary rules (such as assessing the applicability of the law).

The following example illustrates the initial predicates associated with Directive 2010/64/EU, highlighting its division into conditions: the necessary facts for applying the Directive (primarily implemented in Article 1) and a predicate concerning any right under the Directive (e.g., the right to interpretation regarding the lawyer, trial, and specific documents). The system determines that a person has a right when the conditions for applying the Directive are met and at least one right is granted.

The Structure of the Directives

Directives tend to be broad in scope and impose minimal requirements for specific rights. For instance, many rules regarding the right to translation are based simply

on whether an individual is a suspect or accused. As an example, Article 2 (1) of Directive 2010/64 states:

Article 2 (1):

Member States shall ensure that suspected or accused persons who do not speak or understand the language of the criminal proceedings concerned are provided, without delay, with interpretation during criminal proceedings before investigative and judicial authorities, including during police questioning, all court hearings and any necessary interim hearings.

This can be modelled as follows:

```
has_right(art2_1, PersonId, right_to_interpretation, trial)
:-
    proceeding_language(PersonId, Language),
    \+ person_language(PersonId, Language).
```

A similar approach is used to model national implementations, such as the following for the Italian and Polish adaptations:

- Italian Implementation Art 143 (1) (1) of the Italian Criminal Procedure Code states that:

The accused who does not know the Italian language is entitled to be assisted by an interpreter – free of charge and regardless of the outcome of proceedings – to understand the accusations against him and follow the actions and hearings in which he participates.

The formal representation in Prolog is:

```
has_right(art143_1_1, PersonId, right_to_interpretation,
    trial):-
    proceeding_language(PersonId, italian),
    \+ person_language(PersonId, italian).
```

Polish Implementation

3.4. CONCLUSION

Art 72(1) of the Polish Code of Criminal Procedure states that:

An accused, who does not have a sufficient command of Polish, is entitled to the gratuitous help of an interpreter (translator).

```
has_right(article72_1, PersonId, right_to_interpretation,
trial):-
    proceeding_language(PersonId, polish),
    \+ person_language(PersonId, polish).
```

To express specific aspects of rights granted by the Directives, we have introduced two additional predicates that reflect high-level categories identified by legal experts during analysis:

- `auxiliary_right(Article, PersonId, Right, Matter)`
- `right_property(Article, PersonId, Property, Value)`

The term *auxiliary* is used to define the relationship between a primary right and the additional rights that expand or facilitate the implementation of the primary right. We refer to these as "auxiliary rights" to emphasize their connection to the primary right. Primary rights form the foundation of the defendant's defense. They are generally applicable, spanning multiple stages of the proceedings (e.g., the right to an interpreter during the trial phase). The EU further enhances these rights by adding specific provisions and characteristics that only come into play once the primary right has been recognized and granted to the defendant (e.g., the right to appropriate assistance for vulnerable persons).

Auxiliary rights do not directly regulate the defendant's legal status but are dependent on, or closely related to, a primary right. This connection can be temporal, meaning that a right only becomes applicable after the primary right is applied (e.g., the right to appeal a decision). Alternatively, the connection may be subjective, implying that the defendant has specific needs, must submit a request, or take action to have the right recognized (e.g., the right for the state to cover the costs of an interpreter).

This distinction is also crucial for the clear transposition of national laws. Since the transposition of EU directives is the responsibility of Member States, it often results in multiple national legal sources implementing a single EU right, each applying to a different stage of the proceedings. Consequently, a single EU right may be transposed into several national laws. Our system is designed to identify both the general rights and the corresponding secondary rights that apply to the individual based on the specific circumstances.

The second rule serves to clarify certain characteristics or details of a right. The key difference between auxiliary rights and properties is that properties exist independently of the defendant's presence and are generally determined by the relevant authority. In our framework, a property of a right refers to a characteristic that provides a marginal benefit to the individual (e.g., the requirement that lawyers be appointed from an official list or that investigative procedures be recorded). Alternatively, it is the responsibility of the national authority, such as the police or the judiciary, to ensure that the proper procedures are followed (e.g., ensuring the quality of lawyers' training to uphold the principle of fair proceedings or monitoring restrictions on the presentation of arrested individuals, such as limits on the application of restrictive measures).

In modeling the various legal sources, these specific predicates have been invaluable for understanding and visualizing the relationship between general and procedural rights.

```
1.  has_right(interpretation, art2_1, PersonId, trial) :-
2.    proceeding_language(PersonId, Language),
3.    \+ person_language(PersonId, Language).

4.    auxiliary_right_scope(art2_3, [art2_1, art2_2, art2_7
5.      ]).

6.    auxiliary_right(art2_3, PersonId, assistance,
7.      vulnerable):-
8.      person_condition(PersonId, hearingSpeechImpediments)
9.      .
```



```
7.   right_property_scope(art2_6, [art2_1, art2_2, art2_7]).

8.       right_property(art2_6, PersonId, means, technology) :-
9.           \+ physical_presence_required(PersonId, interpreter,
                                           safeguard_fairness).
```

In this part listing, we can easily distinguish three articles, located at lines 1, 5, and 8.

Article 2(1) serves as the fundamental provision, with the other articles deriving from it. These connections are represented through *auxiliary-right_scope* and *right_property_scope*.

Article 2(3) is classified as an auxiliary right since it grants an additional entitlement that takes effect only after the general right (Art 2(1)) has been established.

Conversely, Article 2(6) is designated as a right property, as it specifies a particular attribute of Article 2(1), outlining the manner in which the right should be exercised.

In this instance, the relationships between the articles are clearly defined, making their interpretation straightforward. However, in more complex cases, distinguishing between general and auxiliary rights can be challenging.

This approach has also been adopted in national legislation, facilitating the identification of the various rights conferred by lawmakers.

Justification and Explanation

To effectively contest a decision, the affected individual must have full access to all relevant information, including a clear explanation of the reasoning behind it. Anyone seeking to challenge an automated decision should be able to identify the legal foundations on which it is based, as well as understand the inference process that led the system to its conclusion.

Judges and other relevant actors must grasp the underlying logic of the system, including the conditions that influenced its outcome. Additionally, the implementation of explainable systems can offer valuable insights into legal frameworks, enabling policymakers to detect and analyze potential conflicts or shortcomings in legislative drafting and enforcement.

Legal professionals, beyond determining whether a specific right is recognized within a national jurisdiction, are also concerned with the conditions under which that right is granted and the mechanisms available for its enforcement and protection.

The primary non-ISO structure within the rule-based system developed for the CrossJustice Project is the meta-interpreter. A Prolog meta-interpreter is a program that takes a Prolog goal and another Prolog program, then attempts to verify the goal against the second program by applying the rules defined in the first.

At its most basic level, a meta-interpreter could be expressed as *prove(Goal):-call(Goal)*. However, this would provide minimal additional information, as it merely instructs the Prolog interpreter to verify a goal by invoking it directly.

Below, we present a refined version of the prototype meta-interpreter, which serves as the foundation for the explanation method.

```
solve((A,B), Result) :- !,
    solve(A, ARes),
    solve(B, BRes),
    append(ARes, BRes, Result).

solve((A;_), Result) :-
    solve(A, Result).

solve((_;B), Result) :- !,
    solve(B, Result).

solve(member(A, B), [system_predicate]) :- !,
    call(member(A, B)).

solve(\+(A), [not(A)]) :- !,
    call(\+(A)).

solve((A)\=(B), [doNotUnify(A, B)]) :- !,
    call((A)\=(B)).
```

```
solve(A, [system_predicate]) :-  
    predicate_property(A, built_in), !,  
    call(A).  
  
solve(A, [A|[Res]]) :-  
    clause(A,B),  
    solve(B, Res).
```

This flexible yet expandable meta-interpreter is designed to construct a proof tree, mapping the execution flow of the program by logging the sequence of evaluated predicates. In its basic form, it enriches certain predicates with metadata—for instance, tagging user-input data (asserted predicates) as "facts" during evaluation. It then presents users with a clear connection between the input data and the corresponding rules within the rulebase.

This feature is particularly beneficial for legal experts and other users who require detailed and transparent explanations. Additionally, the system supports further integration with external platforms by providing a stable API, facilitating seamless communication with other applications.

Within the CrossJustice project, this integration has been achieved through a web-based graphical interface that enhances usability. The platform allows non-expert users to interact with the system through a simplified form, ensuring accessibility. As a result, the output is enriched with additional metadata and can be formatted according to specific requirements. For example, the system can generate documents automatically, which may prove valuable in legal proceedings.

Advanced users, such as legal professionals or policymakers, can also engage with the system at a deeper level by modifying rules in real time and analyzing the impact of these changes. This capability is particularly useful for European or national legislators when considering amendments to existing regulations or the introduction of new legal provisions.

To illustrate this, let us examine the transposition of Article 5, Paragraph 1, of Directive 2013/48 into the Italian legal framework. The article states that Member States must guarantee that suspects or accused individuals who are deprived of their liberty have the right to have at least one person, such as a family member or employer, whom they have designated, notified of their deprivation of liberty

without unnecessary delay, if they wish to do so.

This directive is reflected in multiple national provisions, including Article 387 of the Italian Code of Criminal Procedure, Article 29 of Law No. 354/1975 Implementing Provisions, and Article 62, Paragraph 1, of Decree No. 230/2000.

The European legislator establishes that any individual deprived of liberty has the right to inform a designated third party of their status. The computable representation of this provision is as follows:

```
%% has_right(_art5_1, PersonId, _right_to_information,
    _inform_person)
has_right(art5_1, PersonId, right_to_inform, PersonId2) :-
    (    person_status(PersonId, suspect)
      ;   person_status(PersonId, accused)
    ),
    person_status(PersonId, deprived_of_liberty),
    person_nominate(PersonId, PersonId2),
    \+ exception(has_right(art5_1, PersonId, right_to_inform
        , PersonId2), _).
```

The reasoning step will inform the user that they have the right to notify a designated third party, as established in Article 5, Paragraph 1. Additionally, the meta-interpreter's explanation will outline the specific conditions required for the application of this rule:

- the individual is an accused person,
- the individual is deprived of liberty,
- the individual has nominated a person to be informed, and
- no exceptions to the article apply.

Unlike the Directive, Italian legal provisions do not include a single, broadly applicable article but rather distribute the rule across several narrower legal norms. Article 387 of the Italian Code of Criminal Procedure mandates that family members be notified in cases of arrest or temporary detention. Similarly, Article 29 of Law No. 354/1975 Implementing Provisions stipulates that relatives must be

informed when an individual enters a detention facility—an obligation reiterated in Article 62, Paragraph 1, of Decree No. 230/2000 concerning penitentiary institutions. However, the conditions for applying these provisions differ from those set forth in the Directive: under Italian law, the person must either be formally arrested or in a state of detention for the right to notification to take effect.

At first glance, the Italian implementation may appear to align with the Directive, as both legal frameworks ultimately grant the same right to the same category of individuals. However, a deeper analysis—enabled by the data provided through the meta-interpreter—reveals key discrepancies. Specifically: (1) under Italian law, the individual does not have the right to personally nominate a third party to be informed, and (2) the Italian provisions apply only to cases of arrest or detention, whereas the Directive employs the broader concept of deprivation of liberty, with arrest and detention cited merely as examples.

Explanation:

```
has_right(right_to_inform, dir, art5_1, mario, luigi)
  has_right(art5_1, mario, right_to_inform, luigi)
    person_status(mario, accused) [FACT]
    person_status(mario, deprived_of_liberty) [FACT]
    person_nominate(mario, luigi) [FACT]
    not(exception(has_right(art5_1, mario, right_to_inform,
      luigi), _))
```

In the example above, the different levels of indentation illustrate how a person named Mario has the right to inform another individual, Luigi, about his deprivation of liberty. This right is established based on the following conditions: (1) Mario is accused of a crime, (2) he is deprived of his liberty during the proceedings, (3) he designates Luigi—whether as a relative or employer—to be notified, and (4) there are no applicable exceptions preventing this notification.

The explanation module of the platform enables a direct comparison between the rights granted under the Directive and their national implementation. Even when both systems recognize the same right, differences in the requirements for its application can lead to discrepancies in transposition or enforcement. This comparative approach allows for the identification of inconsistencies or potential

misalignments between legal frameworks.

To enhance this comparative analysis, argumentation techniques have been integrated into the system. These techniques establish rule hierarchies and identify conflicts between legal norms and predicates. The system also implements a structured preference mechanism that prioritizes legal sources, ensuring that EU Directives take precedence over national legislation. By juxtaposing results from both levels of law, the platform highlights significant differences in the applicability of legal provisions, providing a clear visualization of where national laws diverge from European regulations.

It is widely accepted that legal reasoning is inherently defeasible, meaning that conclusions can be overridden by new evidence or exceptions. In EU legal texts, defeasibility is expressed through phrases such as “*unless proved otherwise*”, “*except in those cases*”, “*subject to*”, “*notwithstanding*”, and similar formulations.

In Prolog, defeasibility is typically handled using negation by failure, making logic programming a form of non-monotonic reasoning. Each of the aforementioned expressions must be explicitly accounted for within the system through this technique. While several approaches to managing defeasibility were considered during rule modeling, existing solutions did not fully meet the platform’s requirements. However, future iterations of the system may explore more sophisticated methods for handling legal exceptions.

Legal reasoning encompasses multiple forms of inference, including deductive, inductive, and abductive reasoning. According to Douglas Walton, deductive and inductive reasoning begin with a predefined set of conditions and derive logical conclusions from them. Abductive reasoning, by contrast, seeks to infer what might be true, relying on presumptive arguments that lead to plausible but uncertain conclusions.

Although abductive reasoning is not traditionally applied in legal contexts, it may have practical benefits in this system. Users—who may not be legal experts—might not always have access to all the necessary information to answer every question definitively. This challenge mirrors real-world scenarios where individuals, even acting in good faith, may lack critical details.

Consider a lawyer preparing a defense for a client located in another country. The lawyer might be uncertain whether the client is fluent in the language

required for the trial. Instead of assuming an answer, the system can leave this information as unknown and generate two alternative legal outcomes—one assuming the client speaks the language, and another assuming they do not. This approach enables dynamic reasoning, particularly in complex legal environments where multiple rulebases interact.

Another potential improvement is the development of a rule representation format that closely resembles natural language (e.g., English). This would make the system more accessible to legal professionals unfamiliar with programming languages like Prolog, allowing them to read and modify rules with greater ease.

Additionally, a natural language interface could transform rule-based statements into user-directed questions. For example, if the system encounters an unverified condition such as “a right is granted if the person speaks the language of the trial”, it could automatically prompt the user with “Does the person speak the language of the trial?” This interactive approach would facilitate more accurate legal assessments while reducing the burden on users to manually input all relevant details.

3.5 Argumentation

In this section, we present an experiment conducted using Arg-tuProlog[15], a logic-based argumentation tool built on top of Prolog that supports defeasible reasoning, argumentation, and rule prioritization. Prolog theories developed in Crossjustice are invoked and executed by Arg-tuProlog as modules.

Two focus cases will be analyzed in the following section. The first demonstrates how Arg-tuProlog—integrating logic programming and argumentation—can reveal explicit contrasts between EU and national laws, particularly highlighting a definition by the Polish legislator that differs from those provided by EU law and other national legal systems. The second case will offer the user a visual representation of the conformity between national legislation and the European Directive.

3.5.1 Case 1

First, we introduce the premises of the focal case. For the sake of simplicity and clarity, we will consider only the relevant legal rules, disregarding any other legal issues that might affect the applicability of the following norms. This chapter aims to demonstrate how argumentation can provide an explainable response representing the legal issue rather than offering a definitive solution.

One major point of contention between Polish national law and European Directives is the differing definitions of the term 'child.' Article 3, paragraph 1, of Directive 2016/800 defines a child as a person below the age of 18. Consequently, the directive applies to all individuals who committed an offense before reaching that age, with certain exceptions.

In this context, Polish criminal law predominantly employs the term "minor" (*nieletni*). This is legally defined in a separate Act on Proceedings in Juvenile Cases as a person who committed a punishable act between the ages of 13 and 17 at the time of the offense, until they reach the age of 18. This definition does not encompass individuals who engaged in criminal activity after turning 17 but were still under 18 during the criminal proceedings. However, certain provisions do afford specific rights to individuals under 18 who do not fall within the category of minors.

For this example, we shall examine the applicability of Article 15 of the directive, which guarantees children the right to be accompanied by the holder of parental responsibility. Additionally, we will consider its Polish counterpart, Article 32(f) of the Act on Proceedings in Juvenile Cases (hereinafter "the Act"), which ensures the same right for minors.

We assume that an individual engaged in a legally punishable act in Poland at the age of 17 (but before their 18th birthday). Following an investigation, they were formally charged with a crime while still in Poland and subsequently summoned to court for a hearing. At this stage, the defendant decides to consult the decision-support system to determine their rights in this scenario.

3.5.2 Case 2

Maintaining the same assumptions as in the first focal case, this example illustrates the effectiveness of argumentation in identifying the requirements that determine the concrete application of rights.

As previously noted, the transposition of directives introduces a higher degree of complexity compared to the direct transposition of EU regulations into national law. This complexity is also reflected in the development of computable legal representations. Unlike regulations, which are generally self-contained, requiring minimal reference to other legal sources and being directly applicable in each Member State, directives establish only general principles and guidelines. These must be interpreted and implemented by national legislators, leading to significant variations in application. Moreover, legal culture further influences the definition of legal terms, adding another layer of complexity.

In this example, we examine the applicability of Article 16 of the directive, which guarantees children the right to be present at their trial. The Polish legislator has not directly transposed this article but instead asserts that the general right of accused persons (regardless of age) to participate in the main trial sufficiently guarantees the applicability of this right, as stipulated in Article 374 of the Code of Criminal Procedure. Specific rights exist for juveniles deprived of liberty, but these are subject to more stringent requirements, such as ensuring the proper exercise of their right to defense and submitting a formal request, as outlined in Article 62, paragraph 1, of the Act.

For this scenario, we assume that an individual engaged in a legally punishable act in Poland while under the age of 17. Following an investigation, they were formally charged with a crime while still in Poland and placed in a temporary detention center for juveniles. The defendant then requests to be summoned to court to explain their actions and consults the decision-support system to determine their rights in this instance.

3.5.3 How to Write Conflicts?

How does argumentation help in dealing with legal issues?

Briefly, we use argumentation to introduce conflictual relationship in the knowl-

edge base. From a legal standpoint, there are several level to consider, first the european rules that should be applied; second, the national laws that deal with the same topic; and finally, how these rules merge and complement each other.

Argumentation is used to highlight the final part. Argumentation can be used to either provide certainty over which legal source is valid, or to compare the meaning of terms between two legal sources and highlight differences.

Furthermore, while the outcome might be same according to both rules, argumentation is useful to provide the user additional information regarding the conditions for the application of each rule.

In writing such conflicts, there are certain key elements which must be considered first, and which will be expanded upon later in this chapter.

First, certain conflict are independent of the knowledge engineer, such as the conformity evaluation. This means that when deciding whether the elements shall be considered equal between two rules, what matters is the language used by the legislator itself. The only error which can be attributed on the bias of the Knowledge Engineer can be the translation of terms from one language to another, but this project has used official translations from several EU sources and legal experts, so this problem is not relevant.

Second, all other conflicts depend on the current legal knowledge of the Knowledge Engineer. During this project the help of various legal experts from each national state has been invaluable to deepen the understanding of this topics. Therefore, what can be added to this representation is only possible due to an in depth study of the subject matter.

While the representation of rules requires less understanding of the intricacies of the matter, in order to introduce conflicts over single terms, more study is necessary. This is directly connected to the risk of error in this stage, as it is harder to achieve an objective truth in this part of the study.

Finally, it should be noted that all added conflicts do not technically provide new information to the user, but only highlight what can be inferred from the available information. This added conflict is dependent on the knowledge of the user as well, as an expert user will be able to understand more from the same information as a non-expert may be able to.

3.5.4 Incorrect Transposition: diverging implementation

```

1  % Directive
2
3  % rule_8
4  has_right(Article, PersonId, Right, Matter, directive_2016_800) :-
5      directive_applies(PersonId),
6      has_right(Article, PersonId, Right, Matter).
7
8  % rule_7
9  directive_applies(PersonId) :-
10     person_status(PersonId, child).
11
12 % rule_6
13 person_status(PersonId, child) :-
14     user_fact(person_age(PersonId, X)),
15     X < 18.
16
17 person_status(PersonId, adult) :-
18     user_fact(person_age(PersonId, X)),
19     X >= 18.
20
21 % rule_9
22 has_right(article15_1, PersonId, right_to_be_accompanied, HolderId) :-
23     user_fact(person_status(HolderId, holder_of_parental_responsibility)),
24     user_fact(proceeding_status(PersonId, started)),
25     user_fact(proceeding_matter(PersonId, court_hearing)).
26
27 % Polish implementation
28
29 law1982_applies(article1_2, PersonId) :=
30     user_fact(proceeding_type(PersonId, criminal)),
31     user_fact(person_age(PersonId, X)),
32     X > 13,
33     X < 17.
34
35 % rule_5
36 person_status(PersonId, adult) :=
37     user_fact(person_age(PersonId, X)),
38     X >= 17.
39
40 person_status(PersonId, minor) :=
41     law1982_applies(article1_1, PersonId).
42
43 has_right(article32f, PersonId, right_to_be_accompanied, HolderId) :=
44     person_status(PersonId, minor),
45     user_fact(person_status(HolderId, holder_of_parental_responsibility)).
46
47 % Facts
48
49 user_fact(person_age(nino, 17)). % rule_1
50 user_fact(person_status(alf, holder_of_parental_responsibility)). % rule_2
51 user_fact(proceeding_matter(nino, court_hearing)). % rule_3
52 user_fact(proceeding_status(nino, started)). % rule_4
53
54 % Conflict
55
56 conflict([person_status(PersonId, child)], [person_status(PersonId, adult)]).
57 conflict([person_status(PersonId, adult)], [person_status(PersonId, child)]).

```

The figure above illustrates the applicable rules in the first focus case. The first five rules, categorized under the Directive label, represent the transposition of articles from Directive 2016/800. In contrast, the rules under Polish Imple-

mentation reflect the corresponding provisions from the Polish Code of Criminal Procedure and the Act. The Facts section contains user-inputted elements, which vary depending on the factual circumstances and the specific query being analyzed. Finally, the Conflict section highlights the core argumentative issue, illustrating the contradictory definitions of child and adult within the legal frameworks.

Each rule plays a crucial role in determining the outcome of the focus case.

The first rule seeks to establish whether the right to be accompanied exists. This determination relies on two premises:

The Directive (2016/800) applies to a given individual. If the Directive applies to a person, they are entitled to the rights granted under its provisions, subject to specific conditions (e.g., the stage of the trial or the authority responsible for ensuring the right). Next, we see that the Directive applies to individuals classified as children. For simplicity, this example considers only one possible criterion: a person is deemed a child if they are under 18 years old. From this, we can infer an additional rule—though not explicitly stated in the Directive—that a person is classified as an adult when they are 18 or older.

Following this logic, the transposition of Article 15(1) specifies that an individual has the right to be accompanied by their legal guardian (identified by the variable `HolderId`) during proceedings. This right applies under three conditions:

- The guardian has been identified,
- The proceedings have commenced, and
- The relevant stage is a court hearing.

Turning to the Polish Implementation, the first rule states that the Act applies only in criminal proceedings involving individuals between 13 and 17 years of age. Consequently, anyone aged 17 or older is treated as an adult under Polish law.

The final rule in this example transposes Article 32f of the Act. In the knowledge base, this has been represented as follows: If an individual is a minor and their legal guardian has been identified, they have the right to be accompanied by that guardian.

A key distinction between the two legal frameworks lies in the nature of their rules. In this representation, all rules derived from the Polish implementation are

```

A0 : rule_1  $\Rightarrow$  user_fact(person_age(nino, 17))
A1 : rule_2  $\Rightarrow$  user_fact(person_status(alf, holder_of_parental_responsibility))
A2 : rule_3  $\Rightarrow$  user_fact(proceeding_matter(nino, court_hearing))
A3 : rule_4  $\Rightarrow$  user_fact(proceeding_status(nino, started))
A4 : A0,rule_5  $\Rightarrow$  person_status(nino, adult)
A5 : A0,rule_6  $\Rightarrow$  person_status(nino, child)
A6 : A5,rule_7  $\Rightarrow$  directive_applies(nino)
A7 : A6,A8,rule_8  $\Rightarrow$  has_right(article15_1, nino, right_to_be_accompanied, alf, directive_2016_800)
A8 : A1,A3,A2,rule_9  $\Rightarrow$  has_right(article15_1, nino, right_to_be_accompanied, alf)

```

Figure 3.1: Arguments from the example in Subsection 3.5.4.

defeasible (i.e., subject to exceptions), whereas the rules based on the Directive are strict (i.e., absolute).

By prioritizing European legal sources over national laws, the system effectively highlights instances where domestic legislation may contradict EU directives. However, this preference structure is adaptable and can be modified to suit different analytical needs.

In this scenario, the input data specifies the following:

- A person, referred to as nino, is 17 years old.
- Another individual, alf, is nino’s legal guardian.
- The proceedings involving nino have begun.
- The relevant stage is a court hearing.

The final two rules encode the fundamental conflict in this case:

- If a person is classified as a child, the system infers that they cannot simultaneously be an adult.
- Conversely, if a person is classified as an adult, they cannot simultaneously be a child.

This logical contradiction forms the core of the legal challenge in determining whether nino qualifies for the right to be accompanied, depending on whether the Directive or the Polish legal framework takes precedence.

Figure 3.1 shows the generated arguments and in Figure 3.3 (left) we see the results of the framework evaluation according to grounded semantic. The system determines the applicable right based on the input facts. In this focus case, the

defendant is entitled to be accompanied by a holder of parental responsibility under Article 15 of Directive 2016/800 (argument A8). However, under Polish law, the same right is not recognized.

In this specific example, manually tracing each rule to identify discrepancies is feasible due to the limited number of rules involved. However, in real-world applications, such an approach would require navigating an extensive rule set, making it impractical to inspect the entire codebase.

By incorporating argumentation into our expert system, we can visually represent the attack-defeat relationship between conflicting legal definitions. In this case, the individual is simultaneously classified as a child under the Directive and an adult under Polish law, creating a direct contradiction. This conflict is represented by the arrows from argument A5 to A4, indicating that the European definition overrides the Polish one due to the defeasible nature of the latter.

If both the Polish law and the Directive were transposed as defeasible norms, neither argument would prevail, leaving the conflict unresolved. Conversely, if both were transposed as strict rules, no conflict would arise at all.

The system does more than just provide a final answer—it enables users to trace the origin of legal conflicts by pinpointing the exact clauses responsible for discrepancies between different legal sources. In this case, the explanation for the missing transposed article in the Polish implementation can be traced back to the differing definitions of child under each legal framework.

While this example highlights argumentation’s effectiveness in revealing explicit contradictions within the rule base, it is important to note that all potential conflicts must be preemptively encoded by the knowledge engineer. As a result, the system’s ability to explain its conclusions is limited to conflicts that have been explicitly incorporated into the rule set.

3.5.5 Incorrect Transposition: diverging requirements

```

1 modulesPath('home/crossjustice-explainability').
2
3 generate :
4     module(Module),
5     prolog(call_module([Module, 'facts'], with_facts_and_length(has_right(X, Y, Z, U, S), F, L)))
6         => right(Module, X, Y, Z, U, F, L).
7
8 directiveconformity :
```

```

9      right(directive, X, PersonId, Right, U, F, L),
10     right(polish, XX, PersonId, Right, UU, FF, LL)
11     => conformity(polish, PersonId, Right).
12
13 directiveNotConformity :
14     right(directive, X, PersonId, Right, U, F, L),
15     ~(right(polish, XX, PersonId, Right, UU, FF, LL))
16     => -conformity(polish, PersonId, Right).
17
18 module1 :-> module('directive').
19 module2 :-> module('polish').
20
21 conflict([right(directive, XX, A, Z, U, F, L)], [right(polish, X, A, Z, U, FF, LL)]) :-
22     \+ conflictFunction(F, FF).
23 conflict([right(polish, XX, A, Z, U, F, L)], [right(directive, X, A, Z, U, FF, LL)]) :-
24     \+ conflictFunction(FF, F).
25
26 conflictFunction(F, FF) :-
27     sameFacts(FF, F),
28     sameFacts(F, FF).
29
30 sameFacts([], _).
31 sameFacts([H|T], Facts) :-
32     member(H, Facts),
33     sameFacts(T, Facts).

```

For the second focus case, we will demonstrate the application of argumentation to verify the conformity between EU directives and national laws.

In this case, we introduce two new arguments that contribute to the goal the system aims to achieve. The variables **F** and **L** represent facts and length, respectively, and are instantiated by the creation of the set of facts necessary for the goal to be satisfied. The conflict in this scenario arises from comparing the set of facts generated by the rules of the EU directive and the set produced by the Polish national transposition. The system will then assess whether the facts in both sets are identical.

Additionally, two new rules have been added: **directiveconformity** and the dual **directivenotconformity**. These rules verify whether both the directive and the Polish national laws return at least one result with the same **Right**, applicable to the same **Person**. This approach allows the system to determine if the national transposition aligns with the original directive or if discrepancies exist, highlighting potential areas of non-conformity.

```

1  % r1
2  has_right(article4_a_iii, PersonId, right_to_information, privacy) :-
3      person_status(PersonId, accused),
4      user_fact(person_made_aware(PersonId, person_status)).
5
6  % r2
7  has_right(article6_3_a, PersonId, right_to_access_lawyer, questioning) :-
8      person_status(PersonId, accused),
9      user_fact(proceeding_matter(PersonId, questioning)).
10

```

```

11 % r3
12 person_status(PersonId, accused) :-
13     user_fact(person_made_aware(PersonId, charge)).

```

Above, we present a list of articles derived from the Directive, following the same ontology structure as in the previous example. For this focus case, we have included the transposition of Article 6(1) and Article 16(1).

For the right to access a lawyer to be granted under the Directive, three conditions must be met: the individual must be either a suspect or an accused, the proceedings must pertain to criminal matters, and the individual must be classified as a child. These three elements collectively define the conditions under which the right is applicable.

Conversely, the right to be present at trial is recognized under the Directive based solely on the individual's status as a child. This single criterion forms the basis for the second right.

```

1 % r4
2 has_right(art301, PersonId, right_to_access_lawyer, interrogation) :-
3     person_status(PersonId, suspect),
4     user_fact(proceeding_matter(PersonId, interrogation)),
5     user_fact(person_request_submitted(PersonId, defence_counsel)).
6
7 % r5
8 person_status(PersonId, suspect) :-
9     user_fact(person_made_aware(PersonId, charge)),
10    \+ user_fact(proceeding_type(PersonId, trial_charge)).

```

The above listing includes articles from Polish national laws, maintaining the same ontology structure as the previous example. For this focus case, we have incorporated the transposition of Article 79(1), Article 62(1), and Article 374(1).

Under the Polish Code of Criminal Procedure, the right to access a lawyer is granted when three criteria are met: the individual must be a suspect or an accused, the proceedings must involve criminal matters, and the individual must be under 18 years of age. These conditions correspond to the set of requirements for the first right.

Regarding the right to be present at trial, the Polish legislator has implemented two articles to align with the Directive's provisions. Article 374 of the Code of Criminal Procedure establishes that the only requirement for this right is that the individual must be an accused. This single criterion defines the applicability of the second right.

Additionally, Article 62 introduces an additional condition: for minors under

3.5. ARGUMENTATION

```
A0 : module1 ==> module(directive)
A1 : module2 ==> module(polish)
A2 : A0,generate ==> right(directive, article16_1, nino, right_to_be_present, trial, [
    person_age(nino, 16),
    person_age(nino, 16)], 2)
A3 : A0,generate ==> right(directive, article6_1, nino, right_to_access_lawyer, trial, [
    proceeding_type(nino, criminal),
    person_status(nino, accused),
    person_age(nino, 16)], 3)
A4 : A1,generate ==> right(polish, article374_1, nino, right_to_be_present, trial, [
    person_status(nino, accused)], 1)
A5 : A1,generate ==> right(polish, article62_1, nino, right_to_be_present, trial, [
    person_request_submitted(nino, present_trial),
    person_status(nino, temporarily_detained),
    person_age(nino, 16),
    proceeding_type(nino, criminal)], 4)
A6 : A1,generate ==> right(polish, article79_1_1, nino, right_to_access_lawyer, trial, [
    person_age(nino, 16),
    proceeding_type(nino, criminal),
    person_status(nino, accused)], 3)
A7 : A3,directivenotconformity ==> -conformity(polish, nino, right_to_access_lawyer)
A8 : A2,directivenotconformity ==> -conformity(polish, nino, right_to_be_present)
A9 : A3,A6,directiveconformity ==> conformity(polish, nino, right_to_access_lawyer)
A10 : A2,A4,directiveconformity ==> conformity(polish, nino, right_to_be_present)
```

Figure 3.2: Arguments from the example in Subsection 3.5.6.

temporary detention (i.e., individuals under 17 years of age), the right to attend the trial is contingent upon the submission of a formal request by the defendant.

```
1 user_fact(proceeding_matter(nino, interrogation)).           % f1
2 user_fact(proceeding_matter(nino, questioning)).           % f2
3 user_fact(person_request_submitted(nino, defence_counsel)). % f3
4 user_fact(person_made_aware(nino, charge)).                 % f4
5 user_fact(person_made_aware(nino, person_status)).          % f5
```

The facts entered into the system specify that an individual, identified as **nino**, is 16 years old. Furthermore, it is established that **nino** is involved in criminal proceedings and has been placed in temporary detention. His status as an accused was assigned following a formal criminal charge. Additionally, the input states that **nino** has submitted a request to be present at his trial.

Figure 3.2 shows the generated arguments and in Figure 3.3 (right) we see the results of the framework evaluation according to grounded semantic. The system returns the resulting applicable right from the facts we have added as input. It is important to note that in this instance, the evaluation does not rely on the standard Prolog-like syntax. Instead, the Prolog code within the modules is assessed through a deductive reasoning process, facilitated by the **generate** rule. The evaluation of the Prolog goal **has_right** is achieved using the specialized **prolog/1** predicate, which allows the integration of pure Prolog code into the ASPIC-like syntax. The results derived from this deductive reasoning are subsequently utilized to construct the **right** arguments. Here, argumentation functions as an additional layer on top

of the existing logical system, providing enhanced explainability and clarity in the process.

In the case of the right to access legal aid, we observe that both the EU directive (argument A3) and Polish law (argument A6) assert the same right under identical conditions. Through the argumentation framework, we can conclude that Polish law is a perfect transposition of the European directive. This is evidenced by the argument for conformity with the directive (A9), which successfully attacks and defeats its counter-argument (A7), primarily due to the presence of the Polish right (A6). If Polish law had not included an article implementing this right, the outcome would have been different. In that case, the argument **notConformity** would be the only one present due to the absence of the Polish right.

However, the situation differs when considering the right to be present at trial. According to the directive, this right is present once (A2), whereas Polish law reflects it twice (A4 and A5). Notably, neither of the Polish implementations mirrors the same arguments as the European source. While the system identifies that both the European and Polish modules assert the same right, it also reveals that neither of the national implementations includes the same requirements for guaranteeing the right. As a result, the system is unable to definitively determine whether Polish law has successfully transposed the directive.

The argumentation framework reveals a conflict: the European right (A2) is in opposition to each of the Polish rights (A4 and A5). Furthermore, all three arguments attack the conformity argument (A10), as neither of the Polish rights fully transposes the European directive. Interestingly, both Polish rights also challenge the **notConformity** argument, as they do return the same right as the directive. Despite these findings, the system cannot make a clear determination, and as a result, the arguments remain inconclusive.

In this scenario, the user is informed that while the right is present in Polish law, the system cannot definitively confirm that the right has been fully implemented according to the European directive. This raises potential concerns regarding the applicability of the right within the national legal system. The ability to highlight such discrepancies offers a level of insight into legal concerns that a traditional expert system might overlook, providing a deeper understanding of potential legal gaps or inconsistencies that may otherwise go unnoticed.

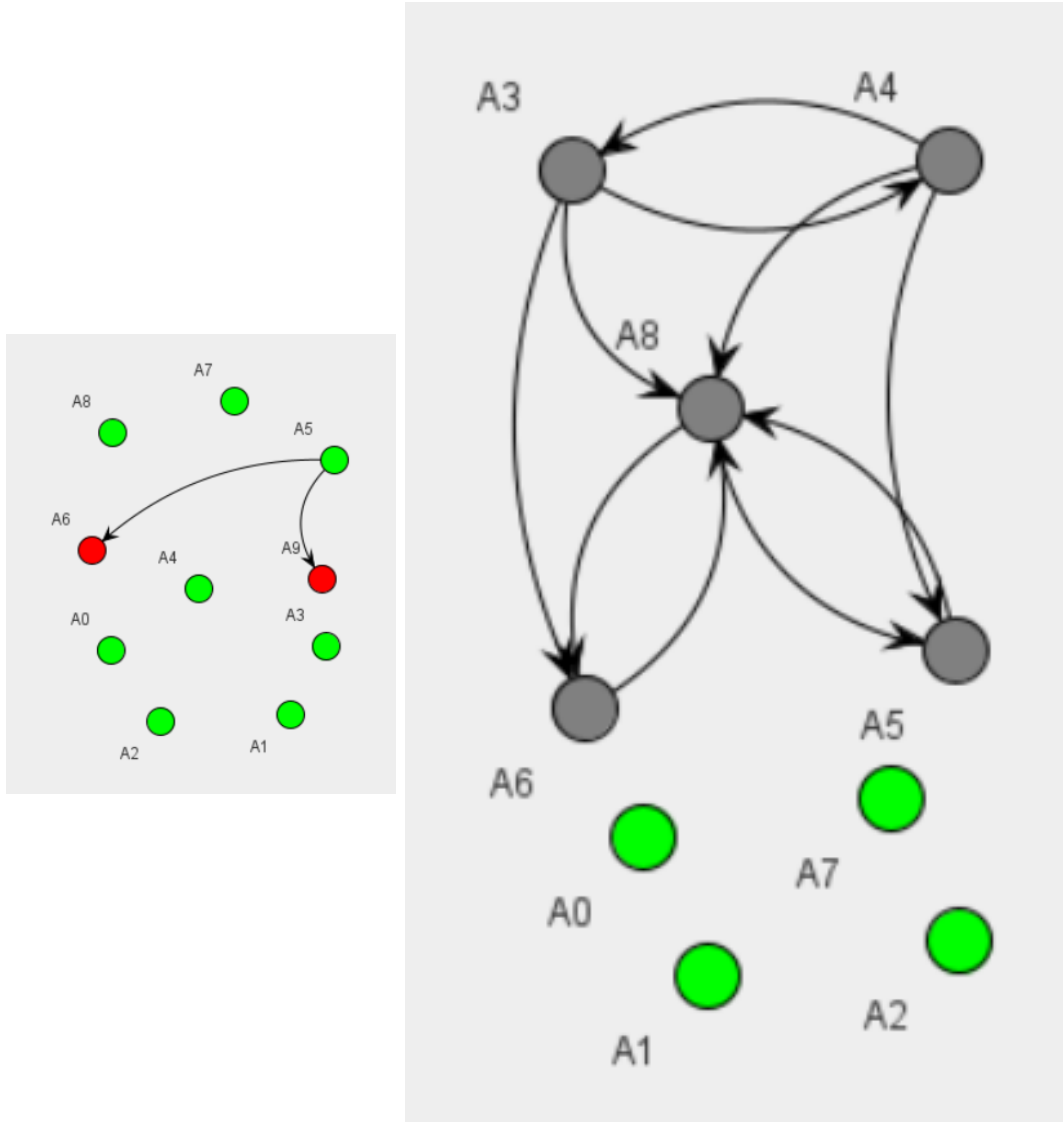


Figure 3.3: Arguments from the examples in Subsection 3.5.4 (left) and Subsection 3.5.6 (right).

To conclude the analysis of this case, we must highlight the next steps for developing our system. First, any conflict within the framework maintains its attack/defeat relationship throughout the subsequent inferences derived from an argument. To demonstrate the efficiency of argumentation, we have intentionally limited the number of applicable rules in our focus cases. However, if we were to extend the inference paths and introduce multiple rules acting as both premises and conclusions, we could show that if a premise is found to be defeated, all inferences built upon that argument would also be defeated.

In theory, both types of conflicts we have discussed—those related to the outcome and the requirements—could apply simultaneously. In such cases, the system would assist the user in pinpointing the exact factor in direct opposition to the rule. The system would not only identify the source of the conflict but also provide a list of facts necessary to generate the conflict. By analyzing the set of factors involved and the system’s outcome, users could efficiently narrow down the relevant rules and identify the premise(s) in direct opposition to another rule.

3.5.6 Incorrect Transposition: conformity check

Listing 3.1: Conformity between directives and national laws check

```
modulesPath('home/crossjustice-explainability').

generate :
    module(Module),
    prolog(call_module([Module, 'facts'],
        with_facts_and_length(has_right(X, Y, Z, U), F, L))
    )
    => right(Module, X, Y, Z, U, F, L).

c0 : right(directive, X, PersonId, Right, U, F, L),
    ~(right(polish, XX, PersonId, Right, UU, FF, LL))
    => -conformity(polish, PersonId, Right).

c1 : right(polish, XX, PersonId, Right, UU, FF, LL),
    ~(right(directive, X, PersonId, Right, U, F, L))
```

```

=> -conformity(polish, PersonId, Right).

c2 : right(directive, X, PersonId, Right, U, F, L),
    right(polish, XX, PersonId, Right, UU, FF, LL)
    => conformity(polish, PersonId, Right).

module1 :-> module('directive').
module2 :-> module('polish').

conflict([right(directive, XX, A, Z, U, F, L)], [right(
    polish, X, A, Z, UU, FF, LL)]) :-
    \+ conflictFunction(F, FF).
conflict([right(polish, XX, A, Z, U, F, L)], [right(
    directive, X, A, Z, UU, FF, LL)]) :-
    \+ conflictFunction(FF, F).

conflictFunction(F, FF) :-
    sameFacts(FF, F),
    sameFacts(F, FF).

sameFacts([], _).
sameFacts([H|T], Facts) :-
    member(H, Facts),
    sameFacts(T, Facts).

```

We will now demonstrate how argumentation is used to verify the conformity between directives and national laws (Listing 3.1).

The conflict here arises from the set of facts used by both the directive and the Polish national law to establish a right. The system verifies whether the facts in both sets are identical. The variables *F* and *L* represent facts and length, respectively, and are populated with the facts used by the system to achieve the goal.

Additionally, we have introduced three new rules: *c0* and *c1*, which verify whether the national law conforms with the Directive, and *c2*, which checks for the absence of conformity. In this example, these rules will determine if Polish

national law is in alignment with the Directive.

It is important to note that, in this case, Prolog-like syntax evaluation has not been utilized. Instead, the Prolog code within the modules is evaluated via the rule `generate`. The evaluation of the Prolog goal `has_right` is accomplished using the special Arg-tuProlog predicate `prolog/1`, which allows the execution of pure Prolog code within the ASPIC-like syntax. The results from the deductive reasoning are then used to build the arguments for the predicates `right-[name of the right]`. This process enhances the system's explainability.

The rules applied in this case, and the required facts, are the same as those in Figure 3.4, with the key difference being that no source of law takes precedence over the other. Therefore, all rules are written using a strict syntax.

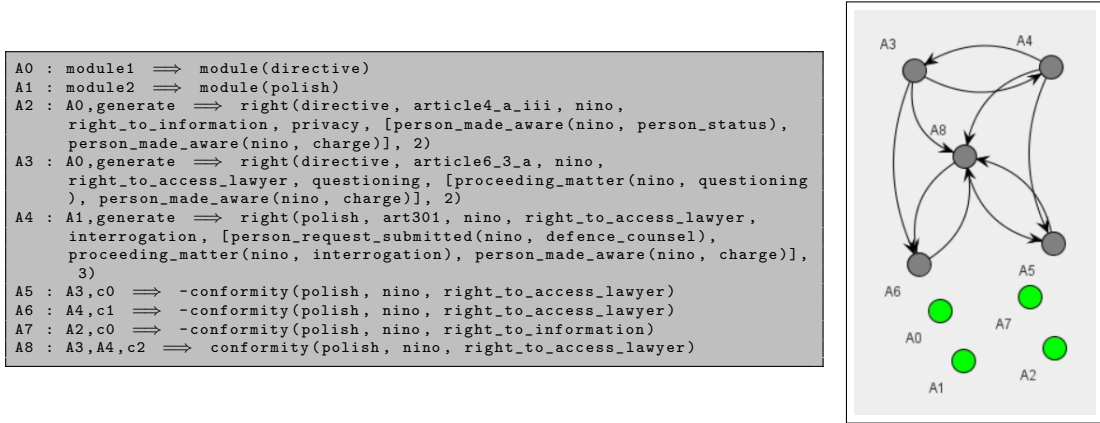


Figure 3.4: Conformity example

Figure 3.4 is used to demonstrate, first and foremost, the most basic result the system can provide when evaluating the conformity of a national source of law with the European Directive, specifically regarding the right to be informed of the right to privacy. According to argument *A2*, the right exists as per Article 4(a_iii), and the only inference derived from that argument is found in *A7*. This illustrates the lack of conformity of Polish law with the Directive, as no corresponding right is returned by the Polish law. Both arguments are marked green, indicating that there are no attacks on any of the inferences.

Additionally, Figure 3.4 demonstrates the system's evaluation of conformity regarding the right to be assisted by a lawyer. Both the Directive (*A3*) and Polish

law (A4) recognize the same right, although different conditions are required to establish that right.

The system only considers the facts manually provided by the user. In this case, the conformity evaluation does not account for the different definitions of "suspect" and "accused" between the EU and Polish legislators, but instead relies on the `user.facts`. The system's results show that the conditions required by the Directive include that the proceeding involve questioning the defendant, while Polish law refers to any interrogation, adding the condition that the suspect must explicitly request the presence of a defense counsel.

The fact that the individual has been made aware of the charges against him, which in Figure 3.4 was the source of conflict between the two legal systems, is the only common condition. The differing definitions of the defendant's status are irrelevant to the conformity evaluation.

To summarize, while the system recognizes that both the European and Polish modules return the same right, neither implementation contains the same requirements for ensuring the right is guaranteed. As a result, the system cannot determine whether Polish law has successfully transposed the Directive.

The graph shows that the European right (A3) conflicts with the Polish right (A4), and both arguments attack the argument for conformity (A8), as it does not fully transcribe the European Directive. Both Polish rights also challenge the two arguments for `-conformity`, as they indeed return the same right as the Directive. Consequently, the system cannot reach a definitive conclusion, and the arguments remain unresolved, indicated by greyed-out results.

Therefore, the user is provided with the information that while the right exists within the Polish legal system, it cannot be definitively stated whether the right has been fully implemented. This creates an issue of applicability. Highlighting these contrasts offers deeper insight into underlying legal concerns that a traditional expert system might not easily uncover.

3.6 The Role of Interpretation

The final discussion of this chapter should focus on the impact of the Knowledge Engineer on the transposition of legal rules.

To start with, it should be clear that the simple act of formalizing a rule in logic form is already an act of interpretation.

By making clear which conditions should be evaluated, even if the language and meaning is similar to that of the legal rule, it does not necessarily mean it is either the right one, or the one applied by judges.

Furthermore, the meaning of legal terms can change and evolve, and different judges and legislators may attribute different meanings to the same term.

We stated before that the role of the Knowledge Engineer can have an impact on the validity of rules and conflicts between rules. Furthermore, even if we assume that the interpretation given reflects the correct one at the current moment in time, it does not necessarily mean that it shall not change in the future.

That is why explanations in the domain is fundamental, as it gives rise to possible points which can be appealed by users and citizens alike.

The purpose of this final part is to describe the relationship between the Knowledge Engineer, the formal model, and the user, and how the correct visualization of the steps taken to achieve that answer is fundamental in the legal domain.

We shall focus on:

- the conditions behind the application of a single rule
- the way the formal model has been written
- the link between legal meaning and formal rules

First, as explained before, explanation in this project is given by having a proof tree of the conditions applied for each rule. This means that each legal conditions is explicitly transposed and reflected in the outcome. A first point of appeal may arise when the user reflect that either a condition should not be valid in the current case, or a condition not seen should be valid in this case.

An additional level to this approach, which has not been implemented in the final version of this project, lies in abducible reasoning. Briefly, it is possible to ask, from a logic based point of view, why such a rule was not applied. It would then give as an answer the missing logical conditions which were found not to be true in the specific case.

Basically, instead of returning the output of the present conditions, it would return as output the missing conditions. This was not implemented as it would go beyond the scope of the project, but in the context of providing an explanation it is interesting to note the various possibilities in this case.

This list of conditions has power in explanation only insofar as it captures the meaning of the natural language legal rule. The connection between the norm and the formalisation is necessary beyond the mind of the KE, as it should be communicated to the user of the model. This issue, although solvable if one were to look at it from a purely logical side, is not immediately clear in its communicability to the non-expert user, which does not understand the logic language.

How to give access to the language? There are several projects in this topic, as seen in Chapter 2, and we will provide an overview of our implementation in the final chapter of this thesis.

As a final point, let us discuss the way argumentation can be used to augment interpretation. By having multiple sides from which one can argue towards an outcome, it is possible to evaluate and compare these sides, and figure out which has the stronger claim.

Solving these issues can be based on legal principles, such as validity of the source, or personal principles, such as which jurisdiction is preferred to bring a controversy to.

By putting these conflict solving approaches on the same level, argumentation can be used to display such conflicts, and solve them accordingly.

In our project, we implemented an evaluation method which displays the conformity between EU and national sources, giving priority to the EU one. This is due to the nature of the project, but if multiple national sources were to be taken into consideration, one could develop a priority over these.

Chapter 4

Factor Based Approach

4.1 Machine Learning for Extraction

We aim to develop a system for classifying legal cases and providing detailed explanations using factor-based methods. Factor-based reasoning involves breaking down legal cases into key factors or elements that influence the decision. Instead of relying on the manual extraction of legally relevant elements, which can be time-consuming and prone to human error, we seek to incorporate automatic methods. These automatic approaches would be faster, more efficient, and scalable, enabling the system to handle large volumes of cases with minimal human intervention.

The goal is to enhance the system’s capacity for explainable AI (XAI), ensuring that each decision is accompanied by a clear and comprehensive justification. For every outcome, the system will provide a set of elements—legal principles, facts, and precedents—that are joined together to form the basis of the decision. This not only improves transparency but also allows users to trace how the system arrived at its conclusion, making the process more trustworthy and understandable.

Additionally, the system will reference previous decisions that support the current outcome, further reinforcing the reasoning behind it. By linking the present decision with past cases, we aim to create a system that mirrors the way legal professionals reason by drawing on relevant precedents. This approach ensures that the system’s decisions are not only automated but also aligned with established legal practices, providing both transparency and consistency in legal decision-

making. Pretrial detention in criminal proceedings is a contentious issue, as it constitutes an exception to the fundamental principle of the presumption of innocence. This measure deprives defendants of their liberty at the earliest stages of legal proceedings, prior to any determination of guilt.

The legitimacy of pretrial detention depends on specific conditions, including reasonable suspicion that the individual has committed the offense, the necessity of preventing flight or further criminal activity, and the risk of interference with judicial processes. These factors require case-by-case evaluation based on the discretionary judgment of the court.

Furthermore, the duration of pretrial detention must not exceed what is strictly necessary to fulfill the objectives established by law [29]. Despite extensive studies on the legal framework regulating pretrial detention, there remains a lack of research on how such decisions are actually made in practice.

Italy and Brazil serve as particularly relevant case studies for investigating this issue¹.

According to the latest data from the World Prison Brief², approximately 30

This research aims to identify the key factors that lead the Supreme Courts of Italy and Brazil to uphold pretrial detention rulings rather than modifying or overturning decisions made by lower courts. Specifically, we examine the relationships between these factors and judicial determinations. To achieve this, we constructed two distinct corpora of Italian and Brazilian judicial decisions and applied unsupervised learning techniques—primarily association and clustering methods—to analyze and extract the most predictive legal features from the case texts.

This study aligns with recent efforts to enhance the interpretability of judicial decision-making through factor-based reasoning, which explains rulings based on legally relevant case features [39, 38]. To identify such factors, described by [8] as predictors of case decisions, we followed methodologies similar to those used in [21].

As no pre-existing datasets were available to support this research, we compiled

¹For more information, see Brazil has the world’s 3rd largest prison population.” <https://www.conectas.org/en/noticias/brazil-worlds-3rd-largest-prison-population/> (2017), online; accessed 30 May 2022; and A measure of last resort? The practice of pretrial detention decision-making in the EU.” <https://www.fairtrials.org/articles/publications/a-measure-of-last-resort-the-practice-of-pre-trial-detention-decision-making-in-the-eu/> (2016), online; accessed 30 May 2022.

²World Prison Brief. <https://www.prisonstudies.org/>, online; accessed 09 Jun 2022.

two separate datasets of Brazilian and Italian judicial decisions. The Brazilian corpus consists of 2,018 rulings retrieved from the official website of the Brazilian Supreme Court (stf.jus.br). These documents are structured into the following sections: (a) heading (lawsuit metadata), (b) summary of the judgment, (c) case report (including grounds of appeal), (d) reasoning and decision of the judge-rapporteur, (e) votes of other judges (if they differ from the judge-rapporteur), and (f) final decision.

The Italian corpus comprises 718 decisions from the Italian Supreme Court, sourced from the DeJure database. These documents follow a similar structure: (a) heading (lawsuit metadata), (b) summary of the judgment, (c) case report (including grounds of appeal), (d) reasoning, and (e) final decision. The primary distinction between the two corpora is that Italian judgments do not include dissenting opinions.

This section provides an overview of the methodology and unsupervised learning techniques employed in our study. We approach the research problem through two primary objectives: (i) identifying the relevant factors that influence judicial decisions, and (ii) analyzing the relationships between these factors and court rulings—specifically, whether the Italian and Brazilian Supreme Courts tend to uphold or modify lower court decisions on pretrial detention.

For both the Brazilian and Italian corpora, we implemented a four-step analytical process. First, we manually extracted a set of *objective* factors—explicitly stated elements in the judgments. Second, we conducted an association analysis to identify potential relationships between these *objective* factors and decision outcomes.

Third, we refined our feature extraction process by dividing each dataset into two subsets based on the final decision. Finally, we applied clustering methods to each subset to identify *subjective* factors—those that are less directly observable in the text. Notably, we did not apply association analysis to *subjective* factors, as the datasets had already been partitioned according to decision outcomes.

To conduct our experiments, we utilized existing implementations and standard analytical tools, including the open-source software Orange 3 [23] and Carrot2 [66], as elaborated in section Section 4.2.

To identify relationships between factors and outcomes, we extracted associa-

tion rules having the forms $x \rightarrow y$, where x is a set of factors and y is one of the two outcomes. For each rule, we determined its support and confidence, namely (a) the proportion of the cases in which both the antecedent x and outcome y are satisfied (the likelihood of finding x and y cases), as a fraction of all cases in the dataset, (b) the proportion of cases in which outcome y is satisfied, as a fraction of all cases satisfying factors x (the likelihood of x cases have outcome y).

$$s(x \rightarrow y) = \frac{\text{Frequency}(x,y)}{N} \quad ; \quad c(x \rightarrow y) = \frac{\text{Frequency}(x,y)}{\text{Frequency}(x)} \quad (4.1)$$

Here, N is the total number of cases in the dataset.

To discover these association rules, we applied the FP-Growth algorithm, which scans the entire dataset to identify rules that meet the specified support thresholds. The resulting rules were represented as a conditional tree, which serves to optimize the mining process by minimizing the need for repeated dataset scans [37].

Clustering is an unsupervised learning technique used to detect underlying patterns within unlabeled data [47]. Since documents may share common factors, we adopted a soft clustering approach, where documents can belong to multiple clusters simultaneously. Specifically, we used Hierarchical Clustering, which constructs tree-like structures by merging documents or existing clusters based on their similarities [1]. To measure similarity, we utilized the cosine similarity metric [18].

After forming clusters, we applied the Latent Semantic Indexing (LSI) algorithm, which uncovers the semantic relationships between words in textual documents, thereby identifying the key topics present within the corpus [68]. Additionally, we used the Lingo algorithm to extract frequent word phrases from documents. These phrases are assumed to provide valuable, human-readable summaries of the underlying topics. Lingo leverages LSI to detect latent structures and thematic clusters, matching the descriptions of these clusters with the topics extracted. The best label for each cluster is then determined by a score measure based on cosine similarity [65].

4.1.1 Automatic Factors and Human Factors

In the first step, we manually extracted five objective factors: the prisoner’s status, the name of the judge rapporteur, the crime category, the crime location, and the judgment date. Below, we describe each factor and the possible values it can take based on the data.

1. Prisoner Status refers to the situation of the accused following the appeal ruling. This factor can take two possible values: released or not released. Cases where the Court replaced pretrial detention with house arrest are categorized as released.
2. Judge Rapporteur denotes the judge who prepares a report on the case. The Italian dataset exhibits greater variance in this factor compared to the Brazilian dataset, owing to the differing number of seats in the two Supreme Criminal Courts. The Italian Supreme Court has at least 35 members, who are regularly replaced.³, versus 11 seats in the Brazilian one, where judges have a permanent position.⁴
3. *Crime*, i.e., the general category to which the committed crime belongs to, under the Brazilian and Italian criminal laws. In particular, we identified four main categories: (i) “crimes against the person”, (ii) “crimes against property”, (iii) “drug-related crimes”, and (iv) “criminal organization”.
4. *Location*, i.e., the place where the crime took place. While in Brazil it corresponds to a state, in Italy it is represented by a regional capital.
5. *Date*, i.e., when the judgment was issued. It corresponds to the ruling year.

In the second step, we conducted experiments using the FP-Growth association algorithm (see section Section 4.2). The configuration of parameters used in our experiments is outlined in Table 4.1. To generate a reliable set of rules with Released as the outcome, we had to adjust the support and confidence thresholds. This adjustment was necessary due to the relatively small number of cases where the accused were released in each dataset.

³Corte di Cassazione (Area Penale): <https://www.cortedicassazione.it/corte-di-cassazione/it/area-penale.page/>, online; accessed 30 May 2022.

⁴Supremo Tribunal Federal: <https://portal.stf.jus.br/ostf/>, online; accessed 30 May 2022.

Table 4.1: Association Setup parameters.

Technique	Tool	Consequent Itemset	Parameters
FP-Growth	Orange 3	BR <i>Not released</i>	Min. Supp. 4%, Min. Conf. 70%
		IT <i>Not released</i>	Min. Supp. 4%, Min. Conf. 70%
		BR <i>Released</i>	Min. Supp. 1%, Min. Conf. 40%
		IT <i>Released</i>	Min. Supp. 1%, Min. Conf. 40%

Tables 4.2 and 4.3 show some selected results. In particular, we report the rules presenting a certain degree of similarity within the two corpora.

Table 4.2: Association rules in Italian dataset.

No.	Antecedent	→	Consequent	Supp.	Conf.
1	criminal organization, Reggio Calabria	→	not released	6,6%	93,8%
2	drug law crime	→	not released	23,8%	84,0%
3	Napoli	→	not released	14,5%	82,0%
4	2019	→	not released	4,1%	96,8%
5	crime against property, criminal organization	→	not released	7,2%	82,5%
6	2013, drug law crime, Napoli	→	released	1,1%	88,9%

Table 4.3: Association rules in Brazilian dataset.

No.	Antecedent	→	Consequent	Supp.	Conf.
1	judge rapporteur MA	→	not released	39,8%	82,2%
2	drug law crime	→	not released	30,5%	73,6%
3	São Paulo	→	not released	31,9%	73,9%
4	2019	→	not released	22,7%	94,4%
5	crime against property, criminal organization	→	not released	4,1%	81,1%
6	2013, drug law crime, São Paulo	→	released	1,0%	47,4%

As observed in rules 2 and 5 across both the Italian and Brazilian datasets, certain factors such as drug-related offenses, or the combination of organized crime and property-related crimes, are typically associated with the not released outcome. Similarly, rules 3 and 4 highlight the correlation between the not released outcome and specific dates (2019), as well as locations like São Paulo and Naples.

In contrast, rule 6 in both datasets demonstrates a relationship between the released outcome and a combination of factors, including the year 2013, drug-related crimes, and locations, specifically Naples and São Paulo. However, it is important to note that the confidence level for this association rule is significantly lower in the Brazilian dataset compared to the Italian one.

Overall, the results indicate a high degree of reliability for association rules tied to the not released outcome in both datasets. However, we did not uncover any high-confidence association rules related to the released outcome, even after lowering the confidence threshold.

Moving to the third step, we divided each dataset into two subsets: one for

judgments favoring the defendant (Released) and the other for those favoring the prosecution (Not released). In the Italian corpus, the first subset contains 614 judgments, while the second includes 104 judgments. In the Brazilian corpus, there are 1,503 judgments in the Released subset, and 515 in the Not released subset. Prior to clustering, we applied several pre-processing techniques, including normalization, tokenization with regular expressions, stemming, stop-word filtering, and bigram extraction [47].

For sentence encoding, we aimed for a general approach, opting for well-established methods. For the Lingo algorithm, we utilized the Bag of Words (BOW) model [75, 41], where each word in the vocabulary corresponds to a feature. The value for each feature is computed as the $TF - IDF$ score, reflecting the word's importance. For the Hierarchical algorithm, we employed Word Embeddings, a popular technique in language models and deep learning [13, 57]. The clustering parameters we used are outlined in Table 4.4, taking into account the outcomes and the number of documents in each subset.

Table 4.4: Clustering Setup parameters.

Technique	Tool	Subset	Parameters
Lingo	Carrot2	IT <i>Not released</i> and <i>Released</i>	Cluster Count Base* 15%
		BR <i>Not released</i> and <i>Released</i>	Cluster Count Base 10%
Hierarchical Clustering	Orange 3	BR and IT <i>Not released</i>	Height Ratio* 30%
		BR <i>Released</i>	Height Ratio 30%
		IT <i>Released</i>	Height Ratio 60%
LSI	Orange 3	All	3 Topics

*Measures used to calculate the number of clusters based on the number of documents on input.

Following the last step, for clustering, we rely on the Lingo algorithm, Hierarchical clustering and LSI. Tables 4.5, 4.6, 4.7 and 4.8 report some results obtained by using Lingo, sorted by highest score.

We classified the obtained labels as follows: (a) *grounds* of appeal (i.e. elements alleged by the defendant); (b) the *reasons* of the decision (elements indicated by the judges); (c) the type of committed *crime*; (c) the *location* of the lower court; (d) the *date* of the Supreme Court judgment; (e) and the name of the *judge rapporteur*. During the analysis of the results, we encountered some challenges, particularly due to overlapping labels with similar meanings, and instances where certain documents appeared in multiple clusters.

Table 4.5: Lingo clusters and labels in Italian *Not released* subset.

No.	Label and cluster	DN	Score	Type	Outcome
1	Maggio 2013 (C26)	61	36,15	Date	Not released
2	Nullità dell'interrogatorio dell'indagato (C10)	63	35,53	Grounds	Not released
3	Termini di fase previsti dall'art 303 (C4)	79	35,47	Grounds	Not released
4	Gravità indiziaria delle esigenze cautelari (C23)	61	33,05	Reason	Not released
5	Ipotesi di cui all'art 304 (C24)	61	32,22	Grounds	Not released
6	Napoli Emessa in data (C26)	61	31,43	Location	Not released
7	Principio della presunzione (C12)	63	30,27	Grounds	Not released
8	Reato Associativo Reati Fine (C5)	78	24,65	Crime	Not released

Table 4.6: Lingo clusters and labels in Brazilian *Not released* subset.

No.	Label and cluster	DN	Score	Type	Outcome
1	Vítima compareceu (C27)	150	25,87	Reason	Not released
2	Excesso prazo custódia perdurar 5 meses (C13)	152	24,65	Grounds	Not released
3	Senhora Ministra C. L. Presidente Exatamente (C3)	151	24,23	Judge	Not released
4	Prática crimes tráfico drogas porte (C25)	150	22,34	Crime	Not released
5	Nulidade absoluta processo (C23)	150	20,75	Grounds	Not released
6	Prevista art 44 Lei n 11343 (C24)	150	17,22	Reason	Not released
7	Dezembro 2014 (C12)	152	16,98	Date	Not released
8	Natureza droga apreendida cocaína (C28)	149	10,06	Reason	Not released

In the Not released subset of the Italian corpus, we identified several grounds of appeal, such as the nullification of the defendant's interrogation (label 2), the expiration of the pretrial detention term (label 3), and the violation of the presumption of innocence (label 7). Lingo also extracted labels linked to manually identified objective factors, including location (Naples, label 6), date (May 2013, label 1), and crime type (criminal organization, label 8). The seriousness of risks (label 4), a factor relevant to pretrial detention, is also associated with maintaining prison order.

In the Not released subset of the Brazilian corpus, we found similar grounds of appeal, such as the expiration of the pretrial detention term (label 2) and procedural nullity (label 5). As for the reasons behind the judgment, labels included the victim's appearance in court (label 1), the inability to convert prison sentences into alternative measures in cases involving drug-related crimes (label 6), and the specific nature of the drug seized (cocaine, label 8). Additionally, we identified manually extracted labels, such as the date (December 2014, label 7), the crime (drug-related offense), and the judge rapporteur (C. L., label 3).

In the Released outcome, the Italian subset reveals several notable reasons for the decision. These include procedural nullity related to the defendant's hearing (label 1), as well as the suspension of the prison term and its expiration (labels 2

Table 4.7: Lingo clusters and labels in Italian *Released* subset.

No.	Label and cluster	DN	Score	Type	Outcome
1	L'interrogatorio di garanzia ex art 294 (C5)	12	42,16	Reason	Released
2	Periodi di sospensione di cui all'art 304 (C2)	14	34,52	Reason	Released
3	Sostituzione degli arresti domiciliari (C3)	14	34,52	Grounds	Released
4	Difensore alle ore (C11)	9	29,59	Reason	Released
5	Febbraio 2009 (C6)	11	26,45	Date	Released
6	Doppio dei termini previsti dall'art 303 (C9)	10	26,03	Reason	Released
7	Caso di regressione (C8)	10	24,13	Reason	Released
8	Tribunale di Catanzaro (C12)	8	18,53	Location	Released

Table 4.8: Lingo clusters and labels in Brazilian *Released* subset.

No.	Label and cluster	DN	Score	Type	Outcome
1	Rio de Janeiro RJ (C2)	57	36.85	Location	Released
2	Constrangimento ilegal decorrente excesso prazo (C5)	52	36.69	Reason	Released
3	Regime inicial aberto requer (C10)	52	33.96	Grounds	Released
4	Imppte Defensoria Pública (C3)	57	29.59	Reason	Released
5	Empresas investigadas (C17)	42	22.30	Reason	Released
6	Junho 2017 (C14)	50	20.06	Date	Released
7	Furto insignificante (C21)	9	18.83	Crime	Released
8	G. M. Segunda Turma Habeas Corpus 112 (C12)	51	15.25	Judge	Released

and 6). These factors can also be framed as grounds for appeal, as they were raised by the defendant. Other key reasons include issues related to the defense (label 4), cases being sent back to the lower courts for reconsideration (label 7), and the replacement of imprisonment with less restrictive measures, such as house arrest (label 3). Furthermore, we observed that factors related to the date (February 2009, label 6) and location (Catanzaro Court, label 8) were also relevant.

Similar trends were observed in the Brazilian Released subset, where the grounds of appeal and reasons for judgment included the expiration of the prison term and unlawful detention (label 2), the substitution of prison with alternative, less restrictive measures (label 3), and appeals lodged by the public defender (label 4). Additionally, cases involving investigations of companies were classified as reasons for judgment (label 5). Other relevant factors included minor burglary charges (label 7) and the judge-rapporteur handling the case (label 8).

Tables 4.9, 4.10, 4.11 and 4.12 show some selected results from Hierarchical and LSI.

LSI assigns positive and negative weights to words, which correspond to green and red colors, respectively. A positive weight signifies that a word is strongly associated with a topic, while a negative weight indicates that the word is poorly

Table 4.9: Hierarchical clusters and LSI topics in Italian *Not released* subset.

Topics and cluster	DN	Type	Outcome
(C16) 1: p, art, 2020, comma, n, sospension, termini, d, p p, 2 2: art 304, 304, termini, p, sospension, comma, 304 p, p comma, p p, è 3: tribunal, 3, riesam, 304, art 304, periodo, art 309, 309, 309 p, sospen- sion	11	Grounds/ Date	Not released
(C19) 1: p, n, art, sez, rv, p p, 3, 1, cautelari, comma 2: r, co, cautelari, sentenza, cautelari, esigenz cautelari, esi- genz, associazion, stupefacenti, dott 3: presunzion, art 275, 3, 275, r, interrogatorio, 275 p, co, comma, comma 3	27	Reason/ Crime	Not released

Table 4.10: Hierarchical clusters and LSI topics in Brazilian *Not released* subset.

Topics and cluster	DN	Type	Outcome
(C12) 1: hc, habeas, art, corpu, habeas corpu, ministro, min, tribun, prisão, voto 2: lei, art, pena, liberdade, tráfico, provisória, liberdade provisória, turma, crime, droga 3: pena, provisória, liberdade, prisão, liberdade provisória, 33, art 33, regim, 4º, senhor	235	Crime	Not released
(C21) 1: crime, nº, lei, ministro, voto, tribun, habeas, turma, marco, corpu 2: crime, código, criminosa, organização criminosa, lei, orga- nização, s, art, sob código, código senha 3: habeas, habeas corpu, corpu, crime, lavagem, nº, acórdão, relat, delito, dinheiro	28	Crime	Not released

related to that topic [23]. We experimented with adjusting the number of topics, but found that it did not significantly affect the clarity or comprehensibility of the results. A limitation of combining Hierarchical Clustering with LSI is that we ended up interpreting individual words instead of word combinations or phrases.

In the Not released subset of the Italian corpus, we identified factors previously detected by Lingo, such as the suspension and expiration of the prison term as grounds (C16 topics). Additionally, we observed the seriousness of precautionary measures and the connection between criminal organizations and drug-related crimes, which were key reasons for applying pretrial detention (C19 topics). This pattern was also evident in the Brazilian Not released subset (C12 and C21 topics). In the Released subset of the Italian corpus, we observed results similar to those obtained with Lingo. Specifically, we identified terms related to the defendant's hearing, the general conditions for applying precautionary measures (C7 topics), and the expiration of the prison term (C4 topics). In the Brazilian corpus, we found words referring to the prison term limit (C16 topics) and house arrest as an

Table 4.11: Hierarchical clusters and LSI topics in Italian *Released* subset.

Topics and cluster	DN	Type	Outcome
(C7) 1: p, art, p p, n, comma, cautelar, misura, 1, 2, ordinanza 2: sentenza, appello, fase, interrogatorio, cort, misura, grado, p, pena, p p 3: misura, 2, interrogatorio, bi, pena, comma, art 275, 275, comma 2, carcer	54	Reason	Released
(C4) 1: p, art, comma, cautelar, custodia, n, 1, custodia cautelar, p p, termini 2: art, termin, 1, comma, termini, p, fase, art 1, sentenza, durata 3: misura, custodia, termin, p, sospens, 1, termini, giudic, custodia cautelar, sentenza	16	Reason	Released

Table 4.12: Hierarchical clusters and LSI topics in Brazilian *Released* subset.

Topics and cluster	DN	Type	Outcome
(C16) 1: prisão, min, cautelar, hc, penal, liberdade, c., m., c. m., rel 2: direito, art, prazo, prisão, cautelar, rs, excesso, preventiva, prisão preventiva, duração 3: pena, liberdade, lei, prazo, n ^o , privativa, sp, pena privativa, penal, privativa liberdade	20	Reason/ Judge	Released
(C5) 1: hc, min, prisão, turma, art, sp, habeas, corpus, habeas corpus, ministro 2: liberdade, turma, lei, art, m., c., c. m., dje, liberdade provisória, provisória 3: primeira, primeira turma, g., m., g. m., prisão, domiciliar, min g., turma, prisão domiciliar	33	Reason/ Judge	Released

alternative measure (C5 topics). Additionally, the algorithm extracted the names of two judges associated with the release outcome (C16 and C5 topics).

We constructed four distinct datasets: two consisting of unstructured judicial decisions from the Brazilian and Italian Supreme Courts (982 and 718 documents, respectively), and two others based on structured data. These latter datasets were prepared using clustering techniques to extract relevant legal factors (F) [72].

To summarize, after applying clustering methods to the documents, legal experts reviewed the resulting clusters and identified the key legal factors present. From this analysis, they extracted the relevant factors from each document, categorizing them according to the following variables (features).

(A) Decision reasons (binary variables). Whether the decision addresses:

- $F1$: excess of time in prison.
- $F2$: suspension of time in prison or suspension of the proceedings.
- $F3$: nullity of the interrogation or hearing of the accused.

- *F4*: connection between different crimes or proceedings.
- *F5*: sending back the case to a previous stage or to a judge of another instance (remittance of proceedings)
- *F6*: presence of wiretaps.
- *F7*: the complexity of the proceedings or existing risks such as of the prisoner's flight.
- *F8*: facts inferred and not proven (i.e., the judge understands that a person is aggressive from the evidence of the case).
- *F9*: the victim's statement.
- *F10*: a prisoner caught in *flagrante delicto*.
- *F11*: defence restriction (i.e, the defendant did not have access to prosecution documents).

(B) Crime categories (*C*) (binary variables). Whether the committed crime was:

- *C1*: against a person (including sexual crime).
- *C2*: against a property.
- *C3*: public safety (including crime provided in the firearms law).
- *C4*: against the government, justice administration or public economy.
- *C5*: provided in special laws.
- *C6*: related to criminal organizations.
- *C7*: provided in the drug law.

(C) Others (categorical variables):

- Location: State or regional capital where the crime occurred.
- Judge rapporteur: Judge's name who was responsible for reporting the case.
- Date: Year in which the decision was issued.

(D) Prisoner status (binary variable): whether the decision stated that the accused must be released. This is our target variable, according to which we split each corpus into two subsets, one containing the decisions in favour of the defendant (i.e., ordering release), the other containing the judgements in favour of prosecution (i.e., ordering detention). In the Italian corpus, the first subset contains 104 judgments and the second 614; in the Brazilian corpus 282 and 700.

We approached the research with two main objectives: (i) to compare the learning performance of predictions generated using structured versus unstructured data, applying traditional classification techniques, and (ii) to assess whether the extracted variables from structured data contribute to explaining and interpreting the predictions. For the second goal, we utilized XAI techniques and decision trees. To conduct the experiments, we leveraged existing implementations and standard methods, including Orange 3 [23] and other open-source libraries.

Supervised classification involves learning a function from labeled data (the training set) and using this function to classify new, unlabeled data. For classification to be effective, a sufficient number of labeled records is required to build a reliable model [47].

We employed several approaches for the classification task, as outlined in the following references: [2, 1, 36, 78, 85]⁵:

- Statistical Learning models: Support Vector Machines (SVM) with Radial Basis Function (RBF) kernel, Logistic Regression, and Naïve Bayes.
- Tree-based/Ensemble models: Adaboost, Decision Tree, Gradient Boosting, and Random Forest.
- Instance-based model: k-Nearest Neighbors (kNN).
- Neural-based model: Multi-Layer Perceptron (MLP).

For the structured datasets, no preprocessing was necessary, as the variables used for the experiments (such as crime types and legal factors) were binary.

⁵For all models, we used the default hyperparameters in Orange Data Mining

However, preprocessing was required for the unstructured datasets: conversion to lowercase, lemmatization, stopwords removal, and feature extraction using Term-Frequency Inverse-Document-Frequency (TF-IDF) [1].

We trained the models using k-fold Cross-validation with five stratified folds. The final performance metrics were the average values from the five-fold validation process.

We used Accuracy and F1 Score metrics to assess the quality of the predictions [1].

Following established practices in the literature [44], we defined the majority classifier (which always predicts the majority class) as our baseline.

4.2 Decision Trees

We applied several Explainable AI (XAI) techniques to clarify the predictions of our model.

Specifically, we used Shapley Additive Explanations (SHAP) to provide detailed explanations for individual predictions [51], and Decision Trees to visualize decision rules and offer insights into how the model makes classifications [27].

In this section, we first evaluate the performance of our model's predictions and then analyze the explanations provided for those predictions.

To assess the importance of the extracted features (factors and crime categories), we applied classification techniques to four distinct datasets. These datasets were used to predict whether the court would uphold or alter a prisoner's sentence. For the unstructured judgment data, the input consisted of the judges' opinions, with the exception of the final decision (verdict). The results are summarized in Tables 1 and 2.

When comparing the performance of different methods, we observed that predictions based on structured data performed notably better than those derived from textual input for all techniques in the Brazilian dataset. In contrast, the performance for the Italian dataset showed a more mixed result, with some methods performing slightly better using structured data and others slightly worse. Overall, the Italian dataset yielded slightly higher Accuracy and F1 Scores compared to the Brazilian dataset.

In both datasets, Decision Trees delivered the best F1 scores, with the Italian dataset also showing a tie with AdaBoost.

It's important to note that our datasets are highly imbalanced, with a disproportionate number of "Not Released" outcomes for both the Brazilian and Italian datasets. If we were to use a baseline model that always predicts "Not Released," the Macro F1-scores would be 0.416 for the Brazilian dataset and 0.46 for the Italian dataset, with an accuracy of 71% for Brazil and 85% for Italy.

This highlights the inherent challenge in predicting outcomes in this domain. Both countries' Supreme Courts are generally hesitant to overturn previous decisions, especially regarding pretrial detention cases. Italy, in particular, presents a more difficult scenario, as the number of "Not Released" decisions is significantly higher than those of Brazil.

4.2.1 Shapley Values

To assess the influence of each factor (categorical variable) on the model's predictions, we utilized classical XAI methods, particularly SHAP, as demonstrated in Figures 4.1a and 4.1b. The interpretation of these results is as follows: a positive impact indicates that a factor promotes a "Not Released" decision, while a negative impact suggests it favors a "Released" outcome.

In the Brazilian dataset, the factors with the strongest influence in favor of a "Not Released" decision are:

Factor 5 (remand, referring to sending the case back), Factor 6 (presence of wiretaps), Factor 9 (victim's statement). In terms of crime categories, Crime 2 (crime against property) significantly favors the "Not Released" outcome, while Crime 6 (criminal organization) shows a notable effect on both outcomes, albeit slightly favoring "Released."

The only feature that strongly influences the "Released" outcome is Factor 2 (suspension of prison time or proceedings), which has a more pronounced positive effect on this outcome.

The results for the Italian dataset present some notable contrasts. A key finding is that Factor 8 (inferred facts) clearly favors a "Not Released" decision. Overall, most factors in the Italian dataset lean toward "Not Released," with stronger

4.2. DECISION TREES

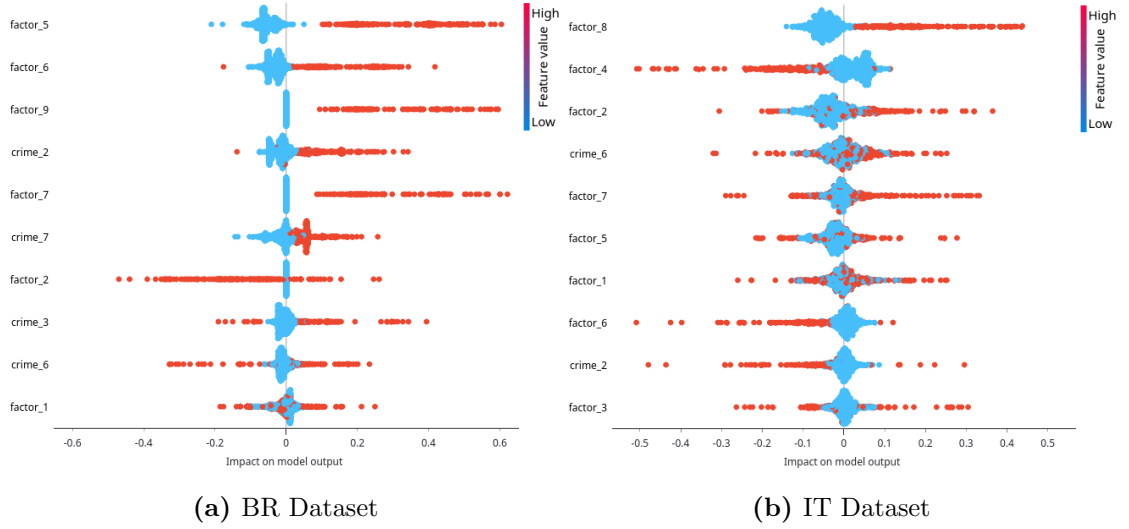


Figure 4.1: SHAP analysis over trained Decision Trees on the two structured datasets.

impacts compared to the Brazilian dataset. However, Factor 4 (connection between proceedings) and Factor 6 (presence of wiretaps), along with Crime 2 (crime against property), are the only features that influence the "Released" outcome.

These results differ from the Brazilian dataset, where Factor 6 and Crime 2 have opposite effects. Additionally, several factors in the Brazilian dataset do not show a strong preference for either outcome. Specifically, Factors 1 (excessive time in prison), 2 (suspension of time in prison), 5 (sending back the case), 7 (complexity of the proceedings), and Crime 6 (criminal organization) all exhibit a weaker effect, although their tendency is slightly towards "Not Released." Interestingly, Factor 1 plays a relatively minor role in the Brazilian dataset as well.

For the final experiment, we constructed small decision trees based on the structured datasets, as shown in Figures 4.2 and 4.3. This method allows us to better interpret the model's outcomes by analyzing the correlation between categorical variables at each node.

In the Italian dataset, the shortest path to a "Not Released" decision is influenced by Crime 6 (criminal organizations) and Factor 8 (inferred facts), which together account for 70 samples. Similarly, the absence of Factor 8 and the presence of Factor 11 (defense restriction) are also common paths leading to a "Not Released" outcome, representing 37 instances.

For "Released" decisions, the majority (253 instances) are associated with the absence of Factor 8 (inferred facts), Factor 11 (defense restriction), Factor 2 (suspension of time in prison or proceedings), Factor 4 (connection between crimes), and Factor 9 (victim's statement). Following the path from node #2, the absence of Factor 8, Factor 11, and Factor 2, along with the presence of Factor 7 (risk of the prisoner) and Crime 7 (drug law), leads to 13 more "Released" decisions.

This suggests that when the crime is drug-related and there are no inferred facts (e.g., no signs of aggression), the likelihood of release increases due to a lower perceived risk to public order. Furthermore, individuals suspected of being part of a criminal organization may be released if they argue that they are not a threat to the proceedings and their detention time has been unjustly suspended — reflecting a common pattern in our analysis and expected outcome.

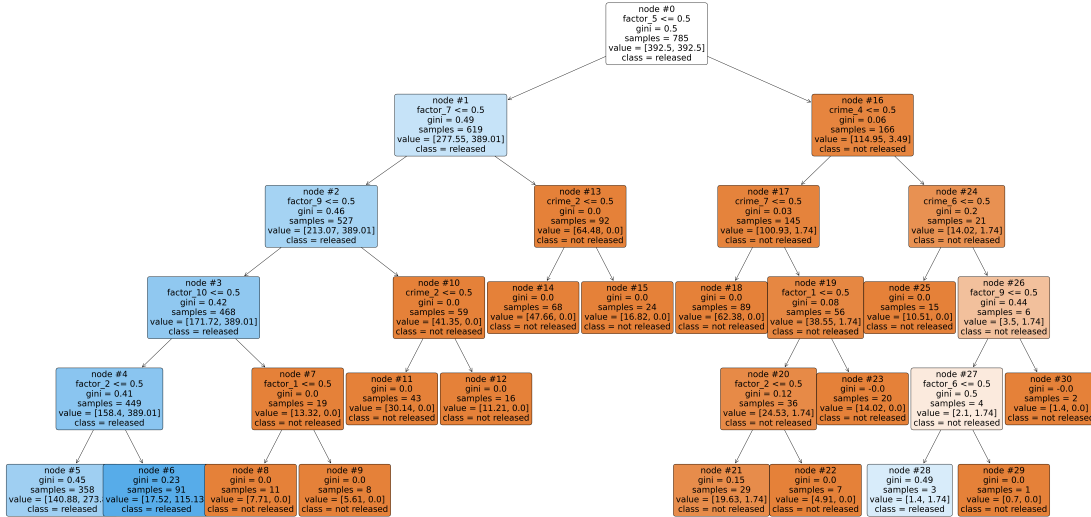


Figure 4.2: Small IT Decision Tree

In the Brazilian small decision tree (4.3), the absence of Factor 5 (remittance of proceedings) combined with the presence of Crime 4 (against the government), Crime 7 (drug law), or Crime 6 (criminal organizations) points to a "Not Released" decision, with 125 and 21 instances, respectively, in the dataset. Interestingly, Crime 4 (white collar crime) tends to favor imprisonment in most cases, which contradicts our initial expectations.

The majority of "Released" decisions stem from the absence of Factors 5 (re-

mittance of proceedings), 7 (risk of the prisoner), 9 (victim’s statement), 10 (*in flagrante delicto*), and 2 (suspension of time in prison or proceedings). Only three instances deviate from this primary path, clearly indicating which factors should be absent for release to occur.

When comparing the two legal systems, we observe that participation in criminal organizations tends to favor detention in Brazil, as seen in both datasets, while the correlation is less straightforward in Italy. Additionally, factors such as the presence of a victim’s statement, when omitted from a case, can lead to the release of a prisoner in both countries.

When comparing the decision tree results with the SHAP analysis, we see that they complement each other in the Italian dataset, although some inconsistencies arise. Specifically, 8 out of 10 factors identified by SHAP are also present in the decision tree, confirming their importance. However, the impact of these factors, particularly which outcome they favor, is subject to interpretation. For example, Factor 6 is predicted by SHAP to favor a "Released" outcome, while the decision tree (node #19) primarily associates it with a "Not Released" outcome. A similar discrepancy appears in the Brazilian dataset, where 9 out of 10 factors are present in both the decision tree and SHAP analysis, sometimes more than once. However, Factors 7 and 9 (nodes #1 and #2) are linked to "Released" decisions, while SHAP suggests they favor "Not Released."

4.2.2 A reflection on XAI

These experiments highlight an important distinction in understanding the results of an experiment, particularly in the context of artificial intelligence (AI) models. The description refers to how certain experiments or models provide a reasoning path that explains how inputs relate to outcomes, but they do so by identifying correlations rather than establishing causality. In essence, while the reasoning can trace a pathway from inputs to outputs, it doesn’t prove that the inputs cause the specific outcomes—only that they are correlated. This is a crucial point in the interpretation of AI systems, particularly when comparing methods for extracting ways of explaining the model of the ML system, and extracting possible reasons for reaching a conclusion.

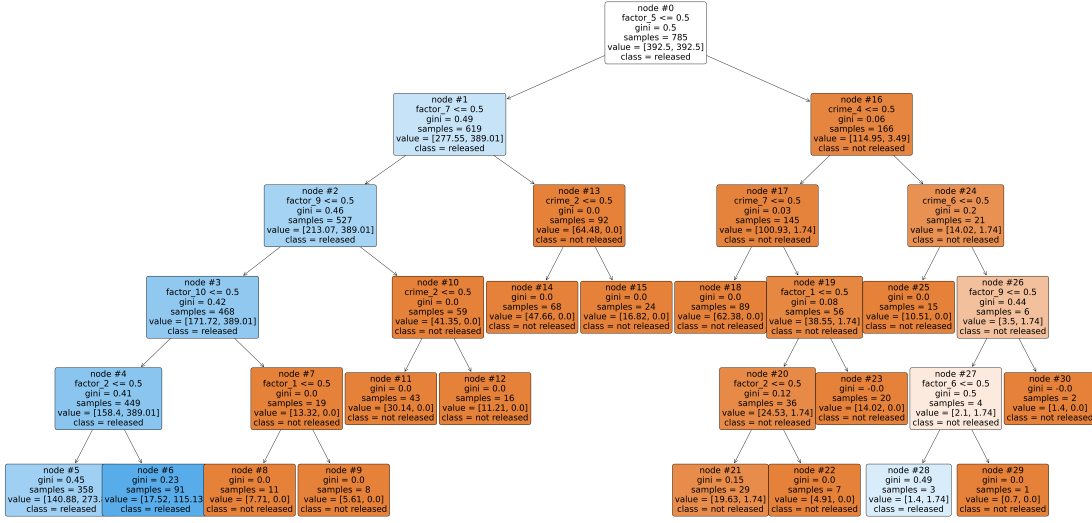


Figure 4.3: Small BR Decision Tree

4.3 Ontologies through LLMs

The following methodology was developed by leveraging a Large Language Model (LLM) approach to extract legal information.

First, we assessed whether the information extracted both automatically and manually using this method aligns with what can be achieved using the latest LLM technology. Second, we evaluated whether this extracted information effectively enables the LLM to concentrate on the pertinent legal issues.

To achieve this, we devised the following strategy: (1) Extract a summary of facts and a summary of law from the case law; (2) Derive the ontology from these summaries; (3) Repeat this process as necessary.

The initial task requires the LLM to adhere to a few basic instructions. Specifically, instructing the model to generate a summary of the text with a word limit, such as 400-500 words, is sufficient to produce a concise text that captures the most critical legal factors.

Subsequently, we verified that the legal elements extracted by the LLM are generally consistent with those identified by other methods. Although there is not a 100% compatibility rate, as the LLM sometimes extracts additional elements — such as the relevance of telecommunications interception — these elements were

4.3. ONTOLOGIES THROUGH LLMS

The text reviews a judicial appeal that was declared inadmissible due to the generality of the complaints, which were not sufficiently linked to the reasoning of the contested order. The grounds for appeal were vaguely formulated and did not concretely support their arguments.

Inadmissibility of the Appeal: The appeal was considered inadmissible because the complaints were generic and not connected to the reasoning of the challenged order.

Phone Interceptions:

Admissibility of Interceptions: The appeal raised the issue of the admissibility of phone interceptions, but in a general manner, without explaining the impact of excluding this evidence on the overall case.

Specificity of the Complaint: The appellant did not specify which interceptions were linked to the allegedly invalid decrees, nor did they provide the supposedly unusable acts.

Content of Interceptions: The court noted that the interceptions used were numerous and well-supported, and the appellant failed to address the actual reasoning behind them.

Backdating of Preventive Detention:

Start Date of Detention: The issue of backdating the start of preventive detention was presented generically, and the court found the argument insufficiently grounded in legal reasoning.

Specific Legal Requirements: The court emphasized that the requirements of Article 297 of the Criminal Procedure Code were not met, particularly the need for the events of the second detention order to predate the first order.

Independent Evaluation of Evidence:

Nullity Claim: The appeal argued that the detention order was null due to the lack of independent evaluation of the evidence by the investigating judge. However, the court found this claim vague, as it merely invited the judge to compare the order with the detention request.

Lack of Specific Complaints: The appellant did not identify the specific findings allegedly adopted uncritically by the order, nor their significance in assessing the seriousness of the evidence.

Evaluation of Evidence Seriousness:

New Investigative Elements: The court highlighted that the criminal association charges involved events after 2014, with new investigative elements revealing the mafia's ongoing activity and the appellant's role.

Statements and Conversations: The statements of a new collaborator, M.E., and intercepted conversations between him and the appellant were considered significant.

Figure 4.4: Summary of case 44

typically overlooked in our previous work due to their dependence on the factual specifics of the case. However, this information proves to be valuable for our purposes.

For CASE 44 (LINK) this is the summary obtained, rephrased and summarised in english:

We have confirmed that utilizing the original language of the judgment, rather than relying on English translations, generally yields more accurate outcomes.

Summarizing judgments and legal decisions during the processing stage is essential to avoid redundancy and reduce the risk of errors when working with large volumes of information. First, we need to divide the judgement into two parts: the first containing the description of the factual elements of the case, and the claims of the parties, and the second half which describes the legal considerations of the judge. Legal texts often contain factual and legal descriptions of the case, some of which may be irrelevant to the core issues.

When a case reaches the highest court, it is typically after having undergone review by multiple lower courts and procedural stages. This extensive progression

can obscure the main factors influencing the final judgment. We ensure that the key factors are always highlighted by simplifying and focusing on the most critical elements. This process is supported by the use of an established table of factors, which allows for a systematic approach to summarization. By doing so, we enhance the clarity and reliability of the legal analysis, ensuring that only the most important information is present in the result.

In our experiments, we also explored the option of leaving the LLM unrestricted, wherein it was tasked with generating a summary without being provided with additional contextual information, such as relevant factors. Our findings indicated that such summaries generally tended to emphasize a smaller set of factors, leading to outcomes that overlooked key elements we had identified in our analysis.

```
Summarize the following text from a judgment:  
[JUDGMENT TEXT].
```

```
Focus on the following legal elements:
```

```
C6: crime related do criminal organizations
```

```
F2: suspension of time in prison or suspension of the  
proceedings
```

```
F7: the complexity of the proceedings or existing risks  
such as of
```

```
the prisoner's flight
```

```
F8: acts inferred and not proven
```

```
The summary should be between 400 and 500 words.
```

```
Use the following format:
```

```
- Case Name:
```

```
- Summary:
```

```
- Key Factors:
```

We also experimented with the language involved, and we found that the capabilities of the LLM to identify key elements in the original language of the judgement was slightly higher than an English summary of the same judgement.

It is evident that the factors previously extracted impose an additional constraint on the language model (LLM), resulting in the extraction of only those

This prompt instructs an expert in knowledge extraction to create a detailed knowledge graph based on a provided text fragment. The task focuses on identifying and extracting relationships between relevant entities in legal texts, particularly regarding magistrate decisions related to the release of detainees under preventive detention measures. The entities include people, organizations, laws, documents, actions, and relevant factors, while relationships between them are described using verbs and other relational terms. The output must be presented in JSON format, with each entry representing two entities (E1 and E2) and their relation (R), described in a maximum of four words. The goal is to ensure clarity and conciseness in identifying the interactions within the legal context.

Figure 4.5: Prompt ontology

selected factors.

In the summary, the related information is further elaborated upon in the ontology section, which becomes the primary focus of the LLM. This approach generates an ontology schema that is centered on the relevant legal elements. However, this constraint may lead to the omission of factors that were not initially considered.

The prompt for ontology extraction is based on the instructions contained in this prompt, rephrased for easier access 4.5:

The objective is to generate an output comprising multiple connections between the pertinent legal factors, previously identified in earlier phases, and the factual circumstances associated with the case.

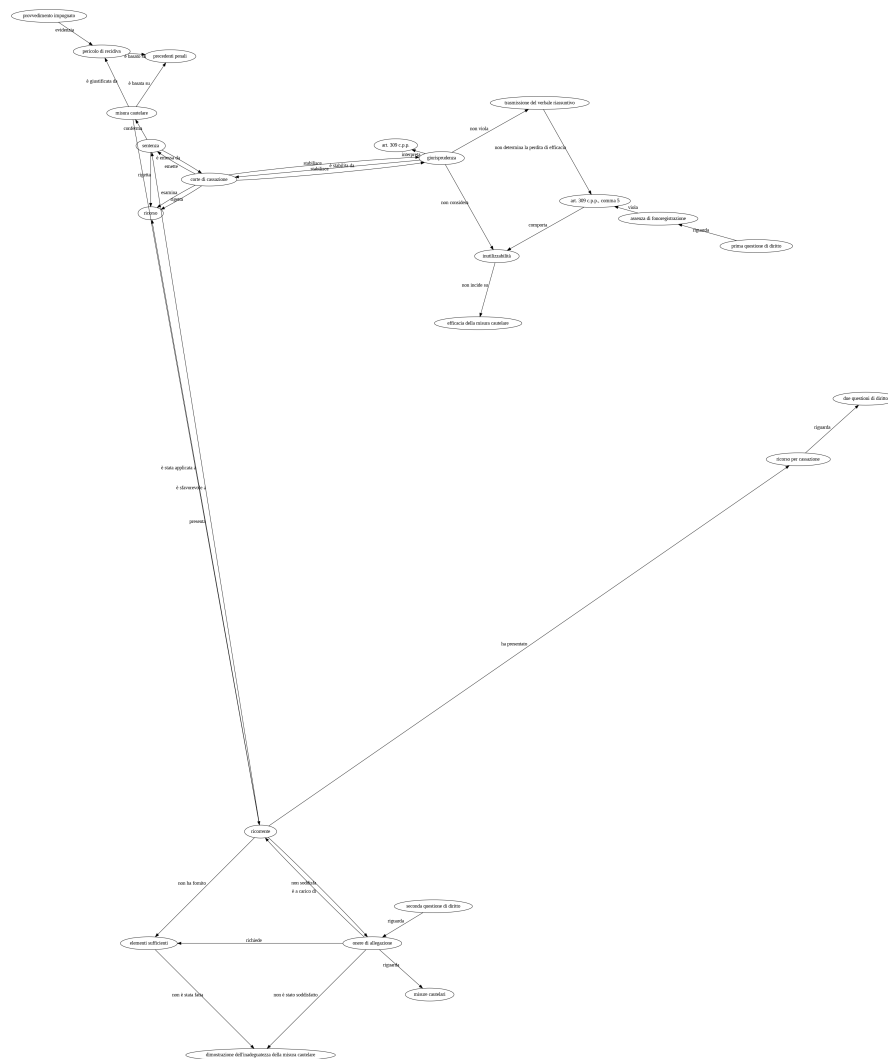
To ensure that the output conforms to the desired format, we include brief examples. Entities and their interrelations are identified, and a corresponding list of these examples is provided.

It is important to highlight that the established connections remain linked, allowing for the existence of multiple relationships for each entity.

During the Knowledge Graph creation we focus on the relationships between key factors and their evaluation by judges. We take as input the summary created, and require the model to provide a graph, extracting relevant entities (factors, parties, crimes) and their relationships. Our goal is the same for this task, extracting relevant information, which deepens our understanding of the relevant key elements the judge has taken into consideration in the decision.

Figure 4.6 is an example of a graph obtained using the factual elements.

One of the issues encountered in this task is to lead the model to create relationships which focus on the key legal elements instead of creating a KG which only analyses the nature of the judgement, such as what pre-trial detention is, or the legal validity of the appeal.



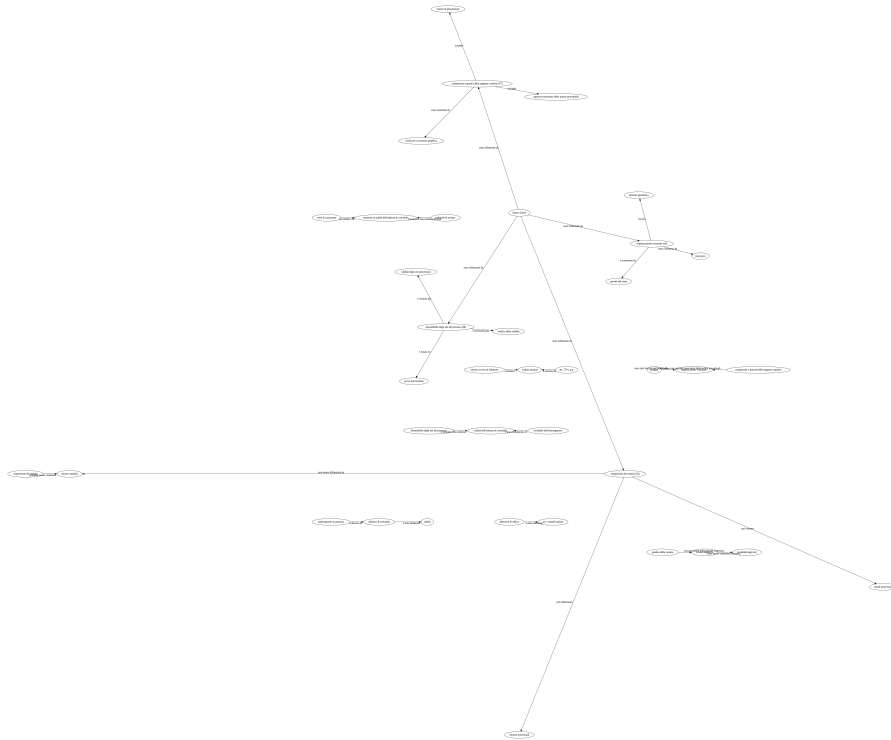


Figure 4.7: Graph obtained using factors

We require the prompt to be asked multiple times, instead of giving a fixed number of relationships to be made, we generally found that asking too many times leads the model to provide very general relationships.

Also, the number or types of examples provided does not change the outcome. Furthermore, the results are consistent between retries on the same judgement, although not completely.

For a better outcome, we used the summary obtained on the legal elements of the judgements, which can be seen in Figure 4.7

The results of the summaries generated after completing the initial task can be reviewed here: <https://hf.co/chat/r/Pn5y7Gx>.

4.3.1 Outcomes and Discussion

This approach further demonstrates that, while the list of elements generated by the LLM is substantively similar to those extracted using the previously outlined

methods, it is significantly more comprehensible to the end user.

However, there are both advantages and disadvantages to this method. A key drawback is that it is slower and more costly. Unlike the previous approach, where all cases were processed simultaneously, this method requires each case to be input individually, followed by the extraction of a summary and the creation of an ontology. This sequential process results in a clear time discrepancy.

Another potential issue lies in the summary itself, which could be a point of failure. While our experiments focused on verifying the presence of the same factors identified by the earlier methods, there is a possibility of errors in the summaries produced. By narrowing the focus of the summary to the legal elements specified in the prompt, the risk of errors is reduced; however, this approach is constrained by the limitations of the existing database.

The costliness of this method also stems from the specific capabilities of the LLM. While open-source LLMs exist and perform reasonably well in summary creation, the most effective models are typically proprietary and, therefore, require financial support to use. This makes the approach unnecessarily expensive.

As it has been shown, such methods, while they do not necessarily rely on existing techniques, they are nonetheless enhanced by it. This theme is, as has been shown, repeated throughout this thesis.

4.4 Explanation in Machine Learning Systems

This chapter demonstrates that automated decisions, derived from a variety of factors, can effectively explain the basis upon which these decisions are made by providing a list of the relevant factors.

The objective is to assess whether the extracted list of factors can be utilized to construct a 'reasonable' collection of legal elements that clarify how a particular decision was reached.

The factors and relationships identified throughout this chapter serve to explain the outcomes of various decisions. Users of such a system could be provided with a list of legal elements that support the outcome of a case or dispute. The connections between these elements can be further refined through the use of decision trees or other logic-based machine learning techniques, or by constructing

an ontology scheme rooted in the factual elements of the case.

Shapley Values and abductive reasoning also play a crucial role in establishing relationships between various factors. Their significance, as well as the identification of irrelevant factors, are vital components in building a strong appeal or defense in a case.

In theory, if a lawyer were to use such a system to verify the connection between facts and legal principles, it could be essential in focusing the defense on the most critical aspects of the case. Moreover, the connections between these elements could be more clearly highlighted using these approaches.

Chapter 5

Combining the two Approaches

5.1 Combining Symbolic and Sub-Symbolic Approaches - From Text To Code

This section will provide an overview on how to combine Logic Based expert systems with LLMs, in order to enhance the capabilities of such systems.

By combining these two approaches, this section aims to demonstrate how LLMs can be leveraged to overcome the Knowledge Representation Bottleneck, potentially transforming expert systems into more scalable and easily updatable platforms. For this study, we focus on the *Facilex* expert system¹, a rule-based system used in the application of grounds of refusal in mutual recognition instruments in the European Union, as a test-bed for evaluating the integration of LLMs [9].

5.2 Experimental Setup

First, we introduce the premises of the case. For the sake of simplicity and clarity, we will consider only the relevant legal rules, disregarding any other legal issues that might affect the applicability of the following norms. This chapter aims to demonstrate how argumentation can provide an explainable response representing

¹<https://facilex-tool.eu/>

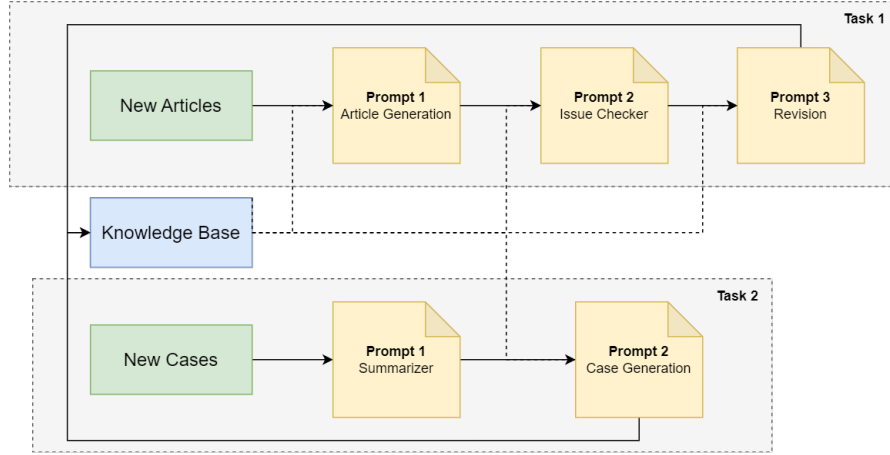


Figure 5.1: Pipeline for Article and Case generation.

the legal issue rather than offering a definitive solution.

One major point of contention between Polish national law and European Directives is the differing definitions of the term 'child.' Article 3, paragraph 1, of Directive 2016/800 defines a child as a person below the age of 18. Consequently, the directive applies to all individuals who committed an offense before reaching that age, with certain exceptions.

In this context, Polish criminal law predominantly employs the term "minor" (*nieletni*). This is legally defined in a separate Act on Proceedings in Juvenile Cases as a person who committed a punishable act between the ages of 13 and 17 at the time of the offense, until they reach the age of 18. This definition does not encompass individuals who engaged in criminal activity after turning 17 but were still under 18 during the criminal proceedings. However, certain provisions do afford specific rights to individuals under 18 who do not fall within the category of minors.

For this example, we shall examine the applicability of Article 15 of the directive, which guarantees children the right to be accompanied by the holder of parental responsibility. Additionally, we will consider its Polish counterpart, Article 32(f) of the Act on Proceedings in Juvenile Cases (hereinafter "the Act"), which ensures the same right for minors.

We assume that an individual engaged in a legally punishable act in Poland at the age of 17 (but before their 18th birthday). Following an investigation,

they were formally charged with a crime while still in Poland and subsequently summoned to court for a hearing. At this stage, the defendant decides to consult the decision-support system to determine their rights in this scenario.

5.2.1 Case 2

Maintaining the same assumptions as in the first focal case, this example illustrates the effectiveness of argumentation in identifying the requirements that determine the concrete application of rights.

As previously noted, the transposition of directives introduces a higher degree of complexity compared to the direct transposition of EU regulations into national law. This complexity is also reflected in the development of computable legal representations. Unlike regulations, which are generally self-contained, requiring minimal reference to other legal sources and being directly applicable in each Member State, directives establish only general principles and guidelines. These must be interpreted and implemented by national legislators, leading to significant variations in application. Moreover, legal culture further influences the definition of legal terms, adding another layer of complexity.

In this example, we examine the applicability of Article 16 of the directive, which guarantees children the right to be present at their trial. The Polish legislator has not directly transposed this article but instead asserts that the general right of accused persons (regardless of age) to participate in the main trial sufficiently guarantees the applicability of this right, as stipulated in Article 374 of the Code of Criminal Procedure. Specific rights exist for juveniles deprived of liberty, but these are subject to more stringent requirements, such as ensuring the proper exercise of their right to defense and submitting a formal request, as outlined in Article 62, paragraph 1, of the Act.

For this scenario, we assume that an individual engaged in a legally punishable act in Poland while under the age of 17. Following an investigation, they were formally charged with a crime while still in Poland and placed in a temporary detention center for juveniles. The defendant then requests to be summoned to court to explain their actions and consults the decision-support system to determine their rights in this instance.

5.3 Experimental Setup

This study utilizes the Facilex Advisory Module, a rule-based expert system created to support legal professionals in interpreting and applying European Union (EU) procedural directives and mutual recognition instruments. Facilex operates by encoding legal norms into computable rules using Prolog, a logic programming language well-suited for formal reasoning and rule-based logic. The system enables legal practitioners to define cross-border legal cases through a series of structured questions. Based on the responses, Facilex generates customized legal assessments, offering insights into relevant legal remedies and potential issues related to cross-border cooperation specific to each case.

The legal framework underpinning this study includes several key EU cases and legislative acts:

- Article 4 of the Council Framework Decision of 13 June 2002 on the European Arrest Warrant and the surrender procedures between Member States;
- Judgment of the Court (Grand Chamber), 16 November 2010, Case C-261/09;
- Judgment of the Court (Grand Chamber), 17 July 2008, Case C-66/08.

For the experiments, we utilize GPT-4o [64], version GPT-4o-2024-08-06², selected for its exceptional performance in natural language understanding and reasoning across various benchmarks [35]. GPT-4o’s ability to generate coherent, contextually relevant, and accurate outputs from limited prompts (i.e., few-shot learning) makes it particularly well-suited for generating Prolog rules based on legal texts.

The validation of the generated Prolog rules follows a two-tiered evaluation process: formal validation and juridical validation. These criteria are adapted from recent research on large language models (LLMs) and their evaluation metrics [16], with a human-in-the-loop approach used for assessing legal rule-application tasks, as described in [35].

²The latest stable model from <https://platform.openai.com/docs/models/gpt-4o>

Formal Validation focuses on the syntactic and executable correctness of the Prolog rules. This step ensures that the Prolog code generated by the model runs without errors, confirming that the rules are syntactically valid and executable within the Facilex system. However, it does not assess the legal relevance or accuracy of the rules.

Juridical Validation is conducted by legal experts, and involves the following criteria:

- **Accuracy:** The extent to which the generated Prolog rules capture the core elements of the input legal text, ensuring the rules are legally sound and comprehensive.
- **Relevance:** The generated rules must align with the expectations of legal experts and adhere to the intended legal reasoning.
- **Human Alignment:** This criterion ensures the system supports collaboration with human experts, facilitating ongoing interaction between the model and the knowledge engineer, rather than automating the expert system creation process entirely.
- **Fluency:** The Prolog rules must be clear, consistent, and coherent, ensuring they are readable and usable by legal professionals in line with the goals of the original expert system.

These criteria ensure that the generated rules are not only correct in form but also relevant and accurate within the specific expert system being addressed.

5.4 Article Generation

The first task employs the Chain of Prompts methodology to evaluate whether GPT-4o can assist in generating new Prolog rules within a legal expert system using few-shot learning. The process begins with existing rules and facts. It's crucial to differentiate between facts and legal conditions: a fact refers to a specific, discrete piece of information that can be input into the system, while a legal condition stems from a legal rule and may rely on multiple facts to be fulfilled.

The central aim of this task is to expand the Prolog knowledge base by adding new rules without introducing new facts. In this setup, the knowledge engineer retains oversight of the system’s domain, responsible for providing facts, while the model’s role is limited to generating rules based on the given examples, ensuring that the rules are both consistent and legally accurate.

5.4.1 Prompts

The task is formalized through a sequence of three prompts, which are to be used in subsequent order by the knowledge engineer³ (Figure 5.1):

- **Prompt 1:** The model is provided with a set of existing Prolog rules and facts, as well as a natural-language representation of a legal article. The model’s task is to generate new rules that formalize the article, ensuring that no new facts are introduced in the process.
- **Prompt 2:** The model then analyzes the generated rules to identify potential issues, such as inconsistent naming, discrepancies between the legal text and the generated rules, or the blending of different legal conditions.
- **Prompt 3:** After the knowledge engineer reviews the identified issues, a new prompt is issued for the model to revise the Prolog rules and correct the mistakes.

Regarding the examples given as input, we provide the first article of the legal source from where the article is taken. Through trial and error we evaluated 1 article to provide good enough guidance, without having the model over fit the examples given.

This structured approach guarantees that human experts remain integral to the process, ensuring that legal interpretations are carefully managed. While the Prolog rules generated by the model are syntactically sound, the focus is on ensuring the underlying legal reasoning aligns with the expectations and expertise of the domain expert.

³Full details are available at https://github.com/LegalMachineLab/JURIX24-fighting_krb.

Therefore, it should be remarked that the KE has full control, at this stage, over when and how to use the prompts.

5.4.2 Output

For the purpose of this thesis, we only focus on Article 4 from the Council Framework Decision 2002/584/JHA (European Arrest Warrant). The goal was to generate the rule for *Article 4(6)*, given comma 1 of the same article as shown in Listing 5.1.

Listing 5.1: Article 4(1) of the 2002/54 FD

```
optional_refusal(article4_1, ExecutingMemberState,
    europeanArrestWarrant):-
    eaw_matter(PersonId, IssuingMemberState,
        ExecutingMemberState, Offence),
    art2_4applies(Offence),
    national_law_not_offence(Offence, ExecutingMemberState).

eaw_matter(PersonId, IssuingMemberState,
    ExecutingMemberState, Offence) :-
    ...
```

Listing 5.2 shows the rules generated by the model using **Prompt 1**. They deal with the refusal of an arrest warrant if the requested person is a resident, national, or staying in the executing Member State and if that state undertakes to execute the custodial sentence or detention order.

Listing 5.2: Article 4(6) of the 2002/54 FD - Generated by GPT-4o

```
%6. if the European arrest warrant has been issued for the
    purposes of execution of a custodial sentence or
    detention order,
eaw_for_execution(PersonId, ExecutingMemberState):-
    executing_proceeding(ExecutingMemberState, PersonId,
        execution_custodial_sentence)
```

```

; executing_proceeding(ExecutingMemberState, PersonId,
    execution_detention_order).

%where the requested person is staying in, or is a national
    or a resident of the executing Member State
person_in_executing_state(PersonId, ExecutingMemberState):-
    person_staying_in(PersonId, ExecutingMemberState)
; person_nationality(PersonId, ExecutingMemberState)
; person_residence(PersonId, ExecutingMemberState).

%and that State undertakes to execute the sentence or
    detention order in accordance with its domestic law;
state_undertakes_execution(ExecutingMemberState, Offence):-
    executing_proceeding_status(Offence,
        ExecutingMemberState, execution_undertaken).

%optional refusal based on article 4(6)
optional_refusal(article4_6, ExecutingMemberState,
    europeanArrestWarrant):-
    eaw_matter(PersonId, IssuingMemberState,
        ExecutingMemberState, Offence),
    eaw_for_execution(PersonId, ExecutingMemberState),
    person_in_executing_state(PersonId, ExecutingMemberState
    ),
    state_undertakes_execution(ExecutingMemberState, Offence
    ).

```

Upon evaluation, it was observed that the expert's original representation of *Article 4(6)* did not incorporate the *state_undertakes_execution* condition, as it is considered implicit within the domain. This condition seems somewhat redundant when considering the conditions in listing 5.2, specifically in lines 3 and 4. The condition *eaw_matter* was introduced by the knowledge engineer to represent an ongoing legal proceeding between the issuing and executing states. One of the prerequisites for grounding this predicate involves confirming that proceedings have commenced in both states.

Moreover, GPT correctly modeled the subrule *eaw_for_execution*, while the knowledge engineer retained the two subconditions in the code. Upon further analysis, it was found that these subconditions were identical, meaning the rule is applied correctly.

Finally, the knowledge engineer can verify that the addition of these conditions does not introduce new information into the rule, allowing for the rejection of this condition if necessary.

Starting from this output, we apply Prompt 2 to obtain a list of possible issues, as shown in Listing 5.3:

Listing 5.3: Output Prompt 2 - Generated by GPT-4o

```
problem 1
Inconsistent naming: [...]
problem 2
Inconsistent naming: [...]
problem 3
Fail to capture relevant information: [...]
problem 4
No correspondence between natural language and rules: [...]
problem 5
Mixed legal conditions: The rule 'state_undertakes_execution
/2' uses the predicate 'executing_proceeding_status/3',
which may not clearly represent the legal condition of a
state undertaking to execute a sentence according to its
domestic law. This could lead to incorrect application of
legal conditions.
```

At this point the engineer may select any relevant problem he has encountered and can confirm from this list. We will now take a look at problem 5. The model will then revise and apply the revisions directly in the text, as shown in Listing 5.4.

Listing 5.4: Output Prompt 3 - Generated by GPT-4o

```
% Revised predicate to better represent the legal condition
of a state undertaking to execute a sentence according to
its domestic law
state_undertakes_execution(ExecutingMemberState, Offence):-
```

```
domestic_law_execution_commitment(ExecutingMemberState,
    Offence).

% New predicate to represent the commitment of the executing
% state to undertake execution according to its domestic
% law
domestic_law_execution_commitment(ExecutingMemberState,
    Offence):-
    executing_member_state(ExecutingMemberState),
    offence_type(Offence),
    member(Offence, [offence1, offence2, offence3]). %
    Replace with actual offences that the state commits
    to execute
```

In general, we found that predicates which did not fully align with the terminology in the natural language rule were often flagged as problematic. Additionally, we observed the creation of several sub-rules aimed at expanding the meaning of existing predicates and introducing new conditions. Another key finding was that the model struggles to maintain consistency, even when operating at a low temperature setting. While the overall structure of the rules tends to remain stable—meaning the core conditions are consistent—the names of predicates, as well as the number and scope of the sub-rules, varied significantly.

5.4.3 Discussion

The validation process results indicate that the generated Prolog rules successfully passed the formal validation step, with all rules being syntactically and grammatically correct. The facts used in the rules aligned with those from the expert's original system, confirming the rules' executability and structural alignment with the original framework.

During juridical validation, some minor issues with accuracy were identified. While the majority of the expert's formalized conditions were captured, there were a few discrepancies: two redundant conditions were added, two negative conditions were used instead of positive ones, and one instance where the system generated three separate rules instead of combining them into a single rule with multiple

conditions in an OR structure. Despite these issues, the overall accuracy remained high, as the core legal reasoning was correctly represented. Out of the 27 conditions modeled by the LLM, the human expert successfully represented 23.

The relevance of the generated rules was fully met, as all necessary legal conditions were included, satisfying the expert’s expectations. The logical connections between conditions were correctly applied, aligning with the intended legal reasoning. Relationships between conditions, such as AND/OR connections, were appropriately captured, preserving the integrity of the legal framework, even when the model introduced slight variations, such as separating multiple conditions into distinct rules.

Fluency was also satisfactory, as the Prolog rules adhered to the stylistic guidelines set by the expert. Predicate names and argument structures were consistent with the provided examples, and the system effectively explained new elements it generated. However, some issues arose regarding adherence to the knowledge engineer’s style: 1) the model tended to generate subrules to maintain syntactic and semantic alignment with the natural language source, and 2) the use of negation by failure in Prolog led to unintuitive code that may not align with the human programmer’s typical style.

Human alignment was facilitated through iterative prompting, which allowed an ongoing dialogue between the model and the knowledge engineer. This process enabled the expert to refine the output progressively, addressing any issues through successive revisions. However, one challenge was that multiple prompts were required to uncover all potential issues.

In conclusion, the experiments demonstrate that GPT-4o can generate coherent and consistent Prolog rules based on few-shot examples, although continuous human supervision remains essential to ensure legal correctness and proper alignment with the expert system’s domain.

5.5 Case Generation

Building on the *Chain of Prompts* approach, the second task centers on analyzing case law and extracting key elements pertinent to the legal expert system’s domain. The objective is to distill the fundamental legal principles and conditions from the

case law, which can then be translated into Prolog rules. The primary aim is to create Prolog rules that accurately reflect the identified elements, ensuring that the original structure is maintained and no new concepts are introduced beyond what the case law provides, as this would exceed the scope of the expert system. Within this framework, the knowledge engineer sets the guidelines and structure for rule creation, ensuring consistency across the system. The model's role is to generate rules that comply with these specifications, ensuring both legal accuracy and coherence with the system's established logic.

5.5.1 Prompts

To structure the task effectively, the prompt creation is divided into two subtasks to address different phases of the process (Figure 5.1):

- **Prompt 1:** The first step involves summarizing the relevant case law, focusing on the conditions associated with the application of a refusal ground under European law. The model is tasked with extracting the core legal principles, the grounds for refusal based on case law, and any influential case law precedents. The output from this prompt should include:
 - The name of the case law,
 - The relevant refusal ground article,
 - The specific conditions under which refusal applies,
 - A clear explanation of the key legal elements, and
 - Any case law influences that have shaped the interpretation of the relevant article.

This prompt ensures that the model condenses the case law into a focused summary, distilling the essential elements necessary for creating Prolog rules while maintaining fidelity to the original legal framework.

- **Prompt 2:** The second prompt focuses on converting the legal conditions identified in Prompt 1 into formal Prolog rules. This prompt guides the model to generate the rules based on the specific articles related to the ground

of refusal. However, it's crucial that the model avoids expanding the scope of the article or introducing new legislative concepts that aren't present in the case law. The output should include:

- The applied article from the legal text,
- The new Prolog rule that formalizes the case law conditions, and
- A brief explanation of how the rule relates to the identified conditions.

After generating the rules, the model must also verify whether the new output aligns with the existing legal rules. This process ensures that the integration of case law does not disrupt the underlying legal structure but rather enhances it by incorporating relevant judicial interpretation.

The Facilex system models the incorporation of case law conditions into existing legal frameworks by generating a representation of the case law and integrating it within the structure of relevant EU legal provisions. Initially, the system formulates a rule that captures the conditions derived from applicable case law. This rule is then merged with the pre-existing legal rule that governs the corresponding EU regulation for the ground of refusal. By linking these two rules, the system ensures that the case law conditions are applied in a way that aligns consistently with the article under which the case is classified, as demonstrated in Listing 5.6.

Alternatively, the system could modify the EU rule itself by incorporating specific exceptions drawn from case law, thereby introducing additional conditions to the EU legal provisions.

5.5.2 Output

To demonstrate the output, Listing 5.5 presents an excerpt of the summary generated by Prompt 1. This input is derived from the raw text of the Judgment in Case C-66/08, which provides clarity on the scope of Article 4(6). The resulting summary emphasizes the critical legal elements, offering a precise and coherent interpretation of the case law. This distilled summary can then be translated into Prolog rules, which will be integrated into the legal expert system for further processing and application.

Listing 5.5: Output Prompt 1 Task 2 - Generated by GPT-4o

- ****Ground of Refusal Article****
 - Article 4(6) of Council Framework Decision 2002/584/JHA
- [...]
- ****Case Law Influences****
 - ****Overall Assessment****: The executing judicial authority must make an overall assessment of various objective factors characterizing the situation of the requested person, including the length, nature, and conditions of their presence, and the family and economic connections they have with the executing Member State.
 - ****Non-Interruption and Compliance****: The fact that the requested person's stay in the executing Member State was not uninterrupted [...] determining whether the person is "staying" in the executing Member State.
 - ****Criminal Activity and Detention****: [...]

As seen in line 2, the model effectively identifies the relevant ground of refusal, ensuring that the core legal principle is accurately recognized. This identification serves as a foundational step for the next phase. In Prompt 2, the Prolog representation of the identified ground of refusal is provided as input, along with an example illustrating how case law is incorporated into the legal expert system. The integration of case law ensures that the conditions derived from judicial decisions are seamlessly embedded into the existing rules framework. Finally, a Prolog rule is generated that precisely reflects the identified conditions, maintaining both legal accuracy and consistency with the expert system's logic. This process is exemplified in Listing 5.6, where the final rule is shown alongside a detailed explanation of its application.

Listing 5.6: Output Prompt 2 Task 2 - Generated by GPT-4o

```
### Article Applied
Article 4(6) of the European Arrest Warrant Framework
Decision
```

New Prolog Rule

```
'''prolog
%%% Case Law Name
case_law(article4_6, CaseLawName):-
    overall_assessment(PersonId, ExecutingMemberState),
    non_interruption_and_compliance(PersonId,
        ExecutingMemberState).

%%% Prolog Article
case_law_refusal(article4_6, CaseLawName,
    europeanArrestWarrant):-
    optional_refusal(article4_6, ExecutingMemberState,
        europeanArrestWarrant),
    case_law(article4_6, CaseLawName).

% overall_assessment(PersonId, ExecutingMemberState) -
    overall assessment of the requested person's situation
overall_assessment(PersonId, ExecutingMemberState):-
    person_presence(PersonId, ExecutingMemberState, Length,
        Nature, Conditions),
    family_connections(PersonId, ExecutingMemberState),
    economic_connections(PersonId, ExecutingMemberState).

% non_interruption_and_compliance(PersonId,
    ExecutingMemberState) - assessment of the continuity and
    legality of the person's stay
non_interruption_and_compliance(PersonId,
    ExecutingMemberState):-
[...]
'''

### Explanation of the Conditions
[...]
```

5.5.3 Discussion

The formal validation of the generated Prolog rules confirmed their syntactical and grammatical correctness. The absence of constraints on the facts allowed the model to operate with greater flexibility, ensuring that the generated rules were executable without encountering structural issues.

In the juridical validation, the model demonstrated a strong grasp of legal accuracy. It successfully identified and represented all the critical legal elements within the Prolog formalization. Each influence from case law was meticulously captured by distinct predicates, and sub-rules were created to fully reflect the legal specifications of each influence. This approach ensured that the generated rules accurately encapsulated the necessary legal complexity.

The relevance of the generated rules was also well-achieved, though it required careful attention. All required legal conditions were correctly included, and the model adhered closely to the legal reasoning expected by the expert. Logical connections between conditions, such as boolean connectors, were implemented appropriately, preserving the integrity of the legal arguments.

However, similar to Task 1, we faced challenges with the model's tendency to introduce redundant or extraneous elements into the output. While these elements were legally accurate, they were not suitable for inclusion in the expert system. For example, concepts such as 'same acts', which are autonomously defined under European Union law for uniformity across Member States, and 'Cooperation and Information Exchange', which pertains to judicial communication between jurisdictions, were identified as unnecessary for this particular task. Although they hold legal significance, they fall outside the scope of the expert system and were excluded from both the summary and the Prolog representation.

Fluency was maintained throughout, with the model adhering closely to the expert's stylistic guidelines. The predicates were named in alignment with natural legal language, and the arguments were structured consistently with the provided examples. Furthermore, the model effectively generated distinct rules for both the case law and its integration with the original legal rule. To optimize performance, we also reduced the size of the input summary, as larger inputs led to structural errors and inaccuracies in Prolog rule generation.

In terms of human alignment, the predicates created to represent new legal concepts were well-explained, enhancing the accessibility of the output for the knowledge engineer. The output of the first prompt could be easily refined or replaced by the legal expert, facilitating a smooth and efficient collaboration process.

Overall, GPT demonstrated strong proficiency in extracting the essential legal elements from case law. The model consistently identified the core legal principles necessary for generating Prolog rules that accurately represent the ground of refusal. Moreover, the generated Prolog rules adhered to the stylistic and structural guidelines set by the knowledge engineer, ensuring consistency and alignment with the expert system’s framework.

5.6 From Code to Text

The following paragraphs entail a series of tasks aimed at enhancing accessibility and understanding of legal reasoning through Large Language Models (LLMs). We have used GPT-4 Turbo as the LLM of choice, as according to several benchmarks in legal tasks, it constantly scores as one of the best (<https://www.vals.ai/>).

The first task focuses on extracting relevant information from Prolog traces, summarizing norms, identifying user rights, and describing the inference process. This output serves as input for the second task, which aims to identify differences in legal reasoning and interpretations across sources, particularly regarding rights application. Building upon this, the third task involves recognizing, ranking, and explaining similarities and differences between cases, providing a list of common norms, and assessing their degree of similarity to the original case. Overall, the objective is to empower laypeople to navigate legal issues autonomously, facilitating comprehension and comparison of legal reasoning.

In our exploration of natural language translation for programming languages and legal texts, we faced challenges related to semantic extraction and multilingualism. To address the former issues, we implemented a segmentation of memory and context for clearer interpretation. Additionally, for multilingual content, we separated languages into distinct memory spaces to improve translation quality and relevance.

We implement the *Chain of Prompts* and *Chain of Memories* approach, which

refers to the sequential use of outputs from one task as inputs for another task in a continuous process. This procedure, applied to the CrossJustice case study, allows the system to chain together relevant legal rules based on the specific facts of a case. By referencing previous legal decisions, the solution to the specific case, and the summaries of case-law stored within the contextual memories, the system can guide the reasoning process towards a solution. This method builds upon existing legal knowledge and improves the efficiency and accuracy of finding an answer to the legal question at hand.

5.6.1 Memory

We will now describe how memory has been applied to the first two tasks of the CrossJustice case study.

During the course of our experiments into the natural language translation of programming languages, we encountered several challenges related to semantic extraction, multilingual representation, and data segmentation of the contextual information. To verify this hypothesis, we wanted to prove, that by having the original text of the legal articles, from which the Prolog is the transposition, the model would more efficiently use the correct terminology and provide a more consistent answer. Firstly, we grappled with the issue of deriving semantic meaning from large blocks of memory, which proved to be overly complex and confusing. Our solution was to implement smaller, more manageable segments that allowed for clearer semantic interpretation. This segmentation allowed for a more structured and focused approach to translation, ensuring that the semantic integrity of each segment was maintained.

Secondly, the integration of multilingual content posed a significant challenge. Initially, we attempted to combine two languages within the same memory context, but this approach was dominated by English, overshadowing the other language and leading to an imbalance. It became evident that our embedding model tended to perceive texts in the same language—or particularly in English—as more similar, regardless of their actual content. To address this, we opted to separate the languages, dedicating distinct memory spaces to each, which improved the translation quality and relevance.

Therefore, to build the memory section for the first two tasks, 5.6.2 and 5.6.3, we divided the Directive 2010/64 into 12 segments and the Dutch Code of Criminal Procedure and the Dutch Criminal Fees Act in around 600 segments, each segment being an article of the legal source. This way, we could ensure that each had only one topic for the model to focus on, and evaluate the relevance to the prompt.

5.6.2 Natural Language Translation

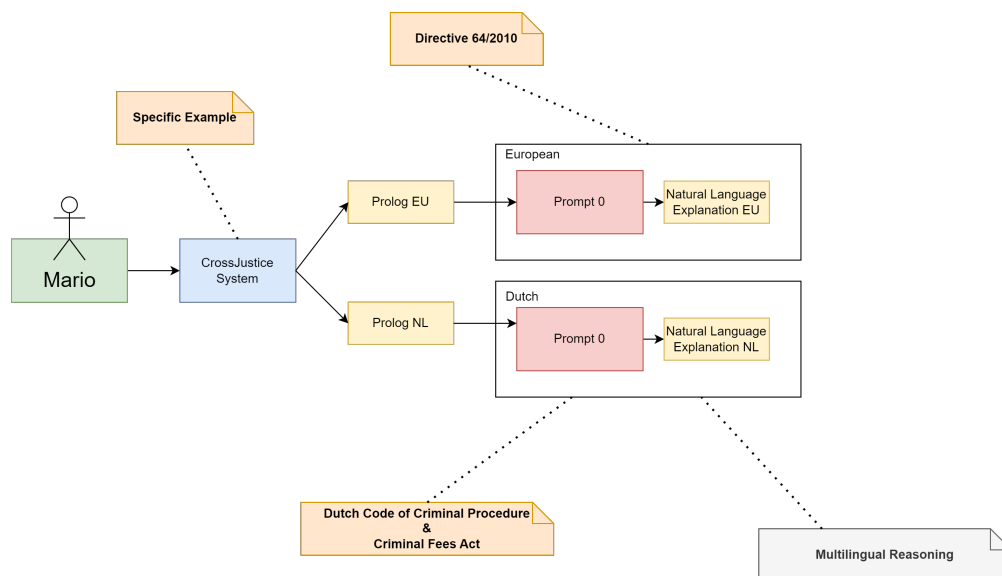


Figure 5.2: Natural Language Explanation

For the first task, our objective was to enable the model to extract the following critical information from the Prolog trace:

- A clear and concise summary of the relevant legal norms involved in the inference;
- A list of rights granted to the user, based on the input facts that represent the case;
- A detailed description of the inference process that led the system to its conclusion.

With these goals in mind, we focused on creating a highly adaptable and versatile prompt capable of applying to any inference produced by the Crossjustice system.

However, we encountered several challenges. The first difficulty was establishing a consistent output structure, as the responses were often presented in varying formats (e.g., bullet points, numbered lists, free-form text). We determined that in order to reduce variability between outputs and improve repeatability and reliability, it was essential to define a fixed structure for the large language model (LLM) to follow.

Secondly, we needed the LLM to account for all relevant legal terms and facts from the scenario and incorporate them into the explanation. The model showed a tendency to overlook key facts unless specifically prompted, posing a challenge for ensuring thoroughness.

The most significant challenge, however, was the model’s tendency to infer meaning from legal text where it was not explicitly provided. Specifically, the model often generalized the meaning of legal terms, disregarding the precision and specific context of legal norms and terminology.

Overall, the model struggled to integrate all the pertinent Prolog facts necessary for applying legal reasoning in a given case, which is critical for **substantial validation**. Additionally, it initially exhibited limitations in accurately representing legal inferences derived from the Prolog trace, particularly those involving the application of subordinate rules, thus falling short in **juridical validation**.

A key observation was the model’s difficulty in accurately interpreting Prolog terms. For instance, when dealing with a predicate concerning the qualification of an essential document, the model often misinterpreted the term “essential,” either broadening or narrowing its meaning inappropriately, instead of adhering to its specific legal definition.

To address these issues, we introduced two legal sources into the contextual information, enabling the model to refer to the original legal articles for the correct lexicon during the translation process, as outlined in section 5.6.1.

Additionally, the LLM was able to efficiently parse the Dutch text, extract the relevant legal norms, and provide responses in English, demonstrating its improved capacity to work with multilingual legal content.

In the prompt designed for this task, we first introduce the object and format of the input data. Next, we establish a fixed structure to follow, consisting of three sections:

- Summary,
- What Rights Do You Have, and
- Why Do You Have Them.

This structure ensures that the model adheres to a consistent format, enhancing repeatability and reliability.

Moreover, we explicitly instruct the system to incorporate all relevant Prolog terms in the explanation, with direct references to the original legal text. This approach aims to facilitate both the clarity of the response and its evaluation from both a substantial and juridical perspective.

Listing 5.7: Natural Language Translation - EU Directive

Summary:

`[...]right to have essential documents translated [...]`

What Rights do You Have:

1. Right to Translation of Essential Documents: `[...]`
2. Right to Challenge and Complain: `[...]`

Why do You Have Them:

1. Right to Translation of Essential Documents: This right is based on the fact that you do not understand the language of the proceedings (`not(person_understands(mario, dutch))`) and that there are documents considered essential for your defense (`essential_document(art3_3, mario, documents)`) as determined by the authority (`authority_decision(mario, essential_document)`).
2. Right to Challenge and Complain: This is derived from the procedural safeguards outlined in the directive,

```
ensuring that you can seek redress if you believe the
translation services provided are inadequate or
unnecessary.
```

```
[...]
```

As shown in Listing 5.7, the outcome is presented in a clear and readable format, adhering to the structure and formatting instructions provided (*formal validation*). It also includes all the relevant Prolog facts used to apply juridical reasoning to the specific case (*substantial validation*), and accurately represents all the legal inferences of the CrossJustice system (*juridical validation*).

Furthermore, the list of extracted memories demonstrates how the relevant legal articles have been correctly identified. This is particularly evident in line 11, where new information, such as the right to challenge and complain about the quality of the translation, is introduced. This right, not present in the Prolog source code, is explicitly granted by Article 3, paragraph 5, of the Directive. This highlights the system’s ability to extract relevant information from the legal source and apply it to the specific case, although the inclusion of information not directly found in the Prolog source was an unexpected outcome of the prompting process.

The same prompt was also applied to the Dutch legal source:

Listing 5.8: Natural Language Translation - Dutch

Summary

```
[...] if you are a suspect in a legal case and you don't
      fully understand or speak Dutch, you have the right to
      ask for important documents related to your case to be
      translated into a language you understand. [...]
```

What Rights do You Have:

1. Right to Translation: You have the right to request that
essential documents in your case be translated into a
language you understand.
2. Right to State-Covered Costs: [...]

Why do You Have Them:

- Right to Translation:

- [...]
 - You submitted a request for essential documents to be translated (`'person_request_submitted(mario, essential_document) [FACT]'`), which is a necessary action to invoke your right to translation under Article 32a(1) of the Dutch Code of Criminal Procedure (`'has_right(art32a_1, mario, right_to_translation, essentialDocument)'`).
- Right to State-Covered Costs:
[...]

The prompt was similarly tested with the Dutch legal source, where it once more met all three criteria. The system offered a coherent and precise explanation of the case study, correctly identifying all relevant facts and their implications for the user. It also accurately referenced the appropriate articles from the Dutch Code of Criminal Procedure. However, unlike the previous example, the system did not include any supplementary information not already provided in the Prolog code, despite the existence of a right to challenge a decision within Dutch law.

5.6.3 Comparison of legal sources

The aim is to develop a model that empowers non-experts to not only access and comprehend legal reasoning more effectively, but also to independently perform complex legal tasks, such as comparing different legal frameworks.

Our objective was for the model to identify the following:

- Variations and discrepancies in the legal reasoning and how norms are applied to the case across different sources.
- Differing interpretations of rights within the legal sources and how these variations might influence the application of the law to the specific case.

Building upon the results of Task 1, we followed by instructing the LLM to enact legal comparison between two sources. To reach a successful result, we experimented with several prompts.

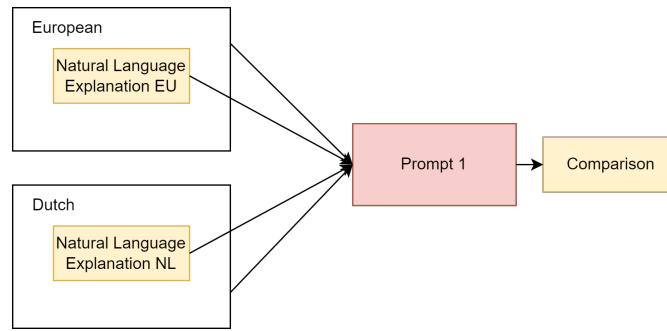


Figure 5.3: Comparison of Legal Sources

We first tested the ability of the LLM to perform legal comparisons directly on the text of legal sources. However, the results were very poor. The system couldn’t identify relevant differences and applied seemingly general and irrelevant legal norms to the case. We then experimented with comparing only the results from the previous task, without providing any additional memory or information. While the results improved, a foreseen obstacle emerged: the system struggled to differentiate between similar terms applied within the same case. For example, when the term “documents” appeared in both sources, even though based on different interpretations, the LLM often failed to recognize these distinctions and generalized the term. Similar to our attempt in Task 1 (as referenced in Section 5.6.1), we tested giving the model access to all legal sources without any pre-processing. This resulted in the model prioritizing English sources and neglecting Dutch references.

Therefore, we decided to provide the model with contextual information (memory) limited to the articles identified as relevant in the previous task, from both EU and Dutch outputs (as shown in Figure 5.3). The final prompt thus included the 20 extracted memories, 10 from each legal source. Doing so, we created a *Chain of Prompts* as the output from Task 1 would be used as input for Task 2. Moreover, it can also be called a *Chain of Memories*, as the memories obtained from Task 1 have been used as contextual information for Task 2.

This subsequent operation resulted in the application of the following prompt to the output obtained from the first step, plus the relevant memories:

Listing 5.9: Prompt Task 2

You have received two legal sources. Compare them according to the following steps:

- 1 - Make a summary of the legal case referencing the Prolog facts in the 'Why do You Have Them' section. Make one summary for each legal source;
- 2 - Compare which Prolog facts are in common between the two legal sources, and which are the source of differences;
- 3 - analyse consequences deriving from step 1 and 2. Explain your reasoning;

THE ANALYSIS MUST FOCUS ON THE CASE AND NOT ON THE GENERAL LEGAL SOURCES

Here we followed the same principles as per the initial prompt. First, we introduced the object of the text given as input, although now in natural language, so no clarification regarding the form was needed. Second, we fixed the structure to be followed when providing an answer. In order to minimise the risk of misinterpretation and hallucinations we asked it to provide a new summary for each legal source, as it also helped us in the evaluation of the answer. We asked the model to create the summary from the 'Why do You Have Them' section, as it contains the Prolog facts needed for step 2. The evaluation of the common facts should therefore be based on the previous step, together with the contextual information provided. Finally, the analysis of the consequences is needed to verify the correct understanding of the legal principles, and whether the previous steps have been correctly processed.

We also explicitly require the system to focus on the case at hand, otherwise we verified that the system tends to focus on general principles and rights.

An example of the final results gathered using the *Chain of Prompts and Memories* (CoPM) in discussion follows:

Listing 5.10: Comparison of Legal Sources

Step 1: Summary of the Legal Case Based on Prolog Facts
[...]

Step 2: Comparison of Prolog Facts

Common Prolog Facts:

- Both legal sources recognize the right of a suspect who does not understand the language of the proceedings to have essential documents translated (`'has_right(art3_1, mario, right_to_translation, essentialDocument)'`).
- [...]

Source of Differences:

- [...]
- The Dutch law explicitly requires a written request for translation, detailing the documents or parts thereof, which seems more procedural in nature.

Step 3: Analysis of Consequences

[...]

The Dutch law's emphasis on written requests for translations introduces a procedural step that could either ensure more precise communication of the suspect's needs or potentially create a barrier if the suspect finds it challenging to specify their request adequately.

[...]

All 3 criteria have again been fully satisfied, although in different degrees. First, with regards to the *formal validation* criteria, the output is presented in an accessible and readable way, following the given structure.

Second, with regards the *substantial validation*, all elements have been correctly identified, i.e., the relevant common rights, as well as the missing property right and difference in conditions/premises. Even if the Articles from two legal sources are mixed together in the contextual information, the model can reliably infer that the Code of Criminal Procedure belongs to the Netherlands, and keep the European Directive separate. It is interesting to note that, as seen at line 11 and 15, the model introduces additional information (the request for information shall be submitted in written form), extracted from Article 32, paragraph 2, Code of Criminal Procedure, that was not present in the output of the previous task. However, it only mentions what the dutch law states should be done, and does not

compare it with the European source, although it can be inferred from the first section of the answer.

Finally, with regards the *juridical validation*, by adding contextual information the unpredictability of the model decreased immensely. However, as can be seen from line 6 of Listing 5.9, we had to specifically request the model to compare the result of the previous input, and not the entire legal source, as it kept moving towards a more general topic instead of being case-specific. Specifically in the last section, *Analysis of Consequences*, this is needed in order to have the LLM be precise in the interpretation of “oral summary” and “request for translation”, as it would otherwise be focusing on general rights such as “fairness” or “defense”.

5.6.4 Analysis of Case Similarity

Building upon the results of Task 2, our goal was for the model to achieve the following:

- Recognize and rank similar cases.
- Provide a list of common norms and premises for each case.
- Explain the degree to which each case is similar or different from the original.

Memory

In our exploration of translating programming languages into natural language, we encountered complex issues when processing extensive textual data from diverse documents. Our analysis, particularly in the Tasks described in section 5.6.2 and section 5.6.3, revealed that mixing text from different sources frequently led to confusion within the model. While splitting legal texts into articles proved a viable strategy, this approach was unsuitable for case law, as it risked severing critical informational links between sections.

The first issue involved the necessity to identify and extract secondary yet significant elements from cases that encompassed a multitude of topics. The second issue pertained to the challenge of capturing connections across disparate parts of the text, which were sometimes too distant for the model to consider within a

single segment, resulting in low similarity scores for larger files. Our solution to these intertwined issues was to create summaries that contained only references to the legal references and factual premises, thereby capturing the essence of the cases, leading the model towards a more case-specific analysis, as seen in Figure 5.4.

Furthermore, maintaining a direct link to the original case was imperative, necessitating specificity to the subject matter at hand. To achieve this goal, the aforementioned summaries will then act as “contextual memory” for the system. When presented with a new legal issue, the system will first filter potentially relevant cases based on the summarized information. Finally, it will analyse the full text of the pre-selected cases and the output of the previous Task to perform a more in-depth similarity evaluation. This approach leverages both the summarized knowledge base and the full case text, leading to a more focused and accurate search process.

Due to the inability to achieve our objectives through a singular prompt, we found it necessary to decompose the tasks into multiple sub-tasks, each connected to the previous one, to achieve the best possible result. This segmentation allowed for a more targeted and effective translation process, although it remains an area for further refinement and optimization.

Out of the 10 total court cases we used, 5 were similar to the original.

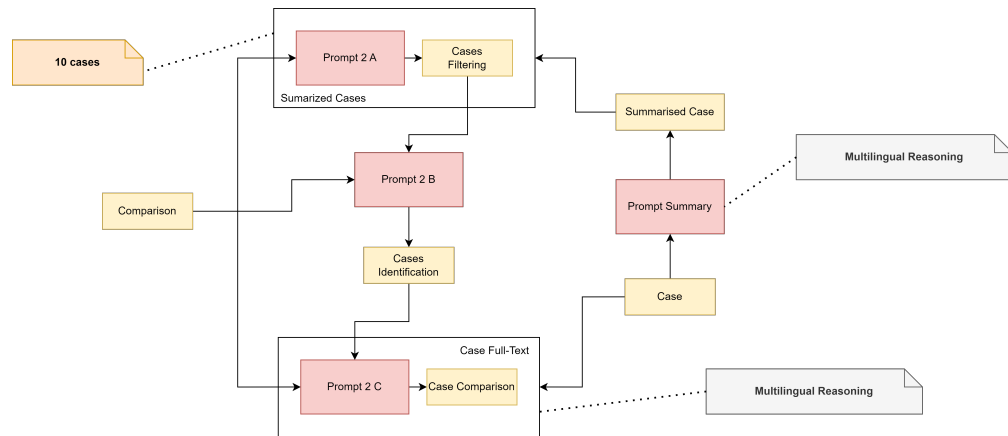


Figure 5.4: Analysis of Case Similarity

We will now take a look at the way we processed the legal cases and the way

the various tasks have been linked to each other.

Case Summarization

An initial challenge consisted in the system returning the wrong case identifier, such as stating that the Directive 2010/64 is a case. Second, we had to find a way to have the model focus only on the relevant topics of each case, which are those connected to the right of translation in cross-border proceedings. A common theme throughout the paper, the tendency of the LLM to focus on general topics, such as the right to fair trial or humane treatment of the person, “forgetting” the specific application of the case study. Third, we also had to verify that the summaries would work no matter the language of the original case, as we tested our approach on cases from the Italian, Dutch, and European courts, although all summaries would be in the English language. The result suggests that without a stable prompting process, the model would often give a wrong ranking, finding similarities where there were none, looking at the wrong cases, and in an extremely unreliable way.

The first problem lied with identifying the relevant cases, according to the similarities with the case study. We decided to first summarize each case, focusing on providing a constant and easy way to verify and evaluate the outcome of each summary.

In the prompt we instruct the system to not go over 2000 characters⁴, in order to counterbalance the tendency of the LLM to give unrequested additional information on the case itself. Moreover, we give a fixed structure, made up of a normative reference, followed by a description of the factual elements that justify the application of the norm. We experimented with providing the full summary in the contextual information, but the model would struggle with identifying the relevant part of the summary. We subsequently divide each summary into several sub-summaries, each containing a norm and a description of that norm.

⁴The amount of characters to be respected was found after a few trial and error runs

Case Identification

The identification of the most similar cases has been divided into two further sub-tasks. First, we ask the system to retrieve from the fully summarized and divided summaries a list of new court cases, each containing only the main factual elements. Already, a first filtering process occurs in this first step, as in the contextual information part we added the output from Task 2, described in section 5.6.3, as well as all the summaries obtained from the earlier Section 5.6.4. Therefore, the extracted information focuses on how similar each of the sub-summaries have in common with the original case, as it's being led towards that conclusion by both the prompt and the contextual memory.

The documents in the memory of the model are identified by a label stating where the section of the text has been extracted from, thus we explicitly remind the system to keep that information for each similar section. The first step involves creating a comprehensive list of all cases. Then, for each individual case, the model is instructed to extract a list of the core factual assertions, based on the summary. This breakdown allows for efficient identification of new cases and a clear understanding of the key factual issues at stake in each one.

At this stage, the output of the model only presents 7 cases out of the 10 introduced as summaries. Out of these, 5 are the relevant ones⁵. Furthermore, the brief description of each case is focusing well on the main applicable rights, as a quick way to immediately identify potential mistakes. This helps the user evaluate the outcome of the summarization stage, without having to read through each case and summary.

While the right in question is the same, in both the European Court of Human Rights case of *Bokhonko vs Georgia*, and the European Directive, they reference different articles (Article 6(1) and Article 6(3)(e) of the European Convention on Human Rights vs Article 3(2) of the Directive 2010/64). This highlights that the model can recognize legal interpretations stemming from different sources, under the same underlying issue, even if they use distinct legal provisions.

The next step involves a final filtering stage, discarding less relevant cases and ranking them according to their relevance with the original. As input, we have the

⁵sleutjes, Case 216/14, hermi vs italy, bokhonko vs georgia, Cassazione 2022/48330

output of Task 2, as well as the output from the first half of the Task 3, as seen above, in Figure 5.4.

We require the highest number of norms and premises in common to verify the relevancy of the new court cases, as we have already filtered through the list of topics, therefore, only those concerning the original case should be picked up as being similar.

Listing 5.11: Case Id2

```
[...]
```

```
2. Ranking Court Cases:
```

- ```
- sleutjes_case: Directly addresses the directive on
 interpretation and translation rights in the EU,
 closely aligned with the original case's focus.
```

```
[...]
```

- ```
- Cassazione_2022_48330: Deals with procedural  
  requirements for translations in the Italian criminal  
  process, somewhat relevant due to its focus on  
  procedural aspects.
```

```
[...]
```

```
4. List of Cases Identifiers (Most to Least Relevant):
```

- ```
- sleutjes_case
- Case_C_216_14
- hermi_vs_italy
- bokhonko_vs_georgia
- Cassazione_2022_48330
```

Finally, we see from Listing 5.11 that the cases have been successfully identified, and the least relevant cases have been discarded. The ranking is also quite good, although one could argue *Bokhonko vs Georgia* should be listed above *Hermi vs Italy*. However, the rationale for keeping the *Cassazione 48330/2022* from the Italian Court is wrong, as it is similar not due to its focus on procedural aspects, but due to the requirements for translation in criminal trial, as identified in the first half of line 6.

Overall, we can state that the outcome is optimal, although we had more issues with its repeatability, as the model tended to change the place of a few of the cases, ranking them slightly higher or lower (especially Case 216/14). This is (probably) an aspect of unpredictability of the LLM we were not fully successful in solving.

### Similarity Evaluation

For the next and final part, we want to highlight the similarities between the original case, and one of the cases identified above. As input, we have the result from Task 2, and the full text in the original language of one of the identified cases, for this example we will use the *Sleutjes* Case (C-278/16).

The goal is to obtain:

- A list of common elements
- A list of diverging interpretations of the normative elements, and a list of different factual elements.
- A brief overview of why the differences may be relevant

#### Listing 5.12: Prompt

```
You have received two legal sources: an analysis of legal
dispute and parts of a court case. Compare them according
to the following steps:
1 - Make a summary of the legal court case;
2 - Compare which facts are in common between the two legal
sources, and which are the source of differences;
3 - analyse consequences deriving from step 1 and 2. Explain
your reasoning.
```

As can be seen from Listing 5.12 we require the model to, first, provide a summary of the case. We have verified that, to enhance the repeatability of the operation, it is useful to have a brief summary of the case at hand. It is also helpful for evaluating the outcome of the model and to check if the information collected is correct. Second, we want to highlight both similarities and differences between the cases, and analyse them based on the original case.

### Listing 5.13: Case Analysis

Summary of the Legal Court Case

[...] The central issue is whether a "Strafbefehl" (criminal order) should be considered an "essential document" that must be translated for a defendant who does not understand German.  
[...]

Comparison of Facts Between the Legal Sources

Common Facts:

- Both legal sources emphasize the right to translation and interpretation in criminal proceedings for individuals who do not understand the language of the court.
  - The necessity of translations being of sufficient quality to ensure fair proceedings and the right of the defendant to challenge decisions regarding translations are acknowledged in both sources.
- [...]

Source of Differences:

- The court case provides specific insights into the application of Directive 2010/64/EU within the German legal system, particularly concerning the classification of a Strafbefehl as an essential document.
- [...]
- The court case details the procedural aspects and legal reasoning behind determining what constitutes an "essential document" and the implications of such a designation on the rights to translation and interpretation.

Analysis and Reasoning

Consequences:

- [...]
- The specificity of the court case in addressing the

translation of a *Strafbefehl* as an essential document reveals the nuanced challenges in ensuring the Directive's implementation does not merely follow the letter of the law but also its spirit, ensuring defendants truly understand the charges against them and can effectively participate in their defense.  
[...]

Overall, our criteria have been fully satisfied. The result effectively simplifies legal vocabulary and presents the information in a clear and readable format. It avoids legalese, making it accessible to everyday users. The structure of the analysis aligns with the input, presenting the common facts, source of differences, and analysis and reasoning sections clearly. This demonstrates strong *formal validation*.

Furthermore, the analysis has grasped the key points of the court case, as requested in the prompt. It correctly identifies the central issue of the court case (classification of a “*Strafbefehl*” for translation) and the Directive it revolves around (EU Directive 2010/64/EU). However, the model focuses mainly on the implementation of the Directive, and it overlooks the fact that the original dispute had to deal with the Dutch law, although it also highlighted that the German court is a new legal system. It is interesting to note that, in this instance, the model did not translate the word *Strafbefehl*, although it did indicate that its meaning is closely related to a criminal order. We can therefore state the system has fully considered all key points (*substantial validation*).

Finally, the analysis correctly identifies the central issue of the court case (classification of a “*Strafbefehl*” for translation). It emphasizes the specific contribution of the court case in clarifying the translation requirement for *Strafbefehl*, and that there is debate over what constitutes an essential document. Here, we can see that the model compares mainly the reasoning of the Directive, as it does not mention anywhere the need for the defendant to explicitly request a document to be considered for translation, as it was stated in the Dutch law. This, although not perfectly, nevertheless demonstrates a valid grasp of the legal issues (*juridical validation*).

In conclusion, we can state that, even if the model may miss key facts, it avoids

presenting demonstrably false information or misconstruing facts and established principles, which is one of the main goals we aimed to achieve. To further refine the last details, further experiments are required.

This section demonstrates the improved accessibility of legal tasks by combining symbolic legal models with generative AI. This creates a new system that is justifiable, multilingual, and functionally versatile.





---

## Chapter 6

# Conclusions

After reviewing the various experiments presented throughout this thesis, we can now finalize our findings on the integration of AI systems in the legal domain. One of the key insights relates to explainability, particularly in how we can seamlessly translate complex legal texts into executable code. This movement from text to code is critical for ensuring transparency in AI systems, allowing users to understand how the system processes input data and reaches conclusions. By maintaining this transparency, we address a fundamental concern in AI applications: the need for users to trust and comprehend the system's underlying processes.

Expert systems, in particular, excel in promoting transparency by offering a clear, step-by-step reasoning process. These systems enable users to follow the logical progression that leads to a conclusion, mirroring the way human experts might analyze legal issues. This not only enhances the trustworthiness of the system but also empowers users to better understand the rationale behind legal decisions. By embedding both statutory law (positive law) and relevant case law into these systems, we ensure that the outcomes are grounded in legal precedent and doctrine, providing a robust foundation for decision-making.

However, it is important to recognize the limitations of machine learning (ML) systems in this context. Unlike expert systems, ML models do not 'reason' through facts in the traditional sense, which can result in black-box issues where the reasoning behind an outcome is opaque. Despite this, ML models still offer significant advantages, such as simplifying user interaction and easing the workload of knowl-

---

edge engineers in constructing legal models. Their primary role in this framework is to enhance accessibility and streamline the modeling process, rather than serving as the core mechanism for legal reasoning.

Moreover, the interaction between the user and the system is a central feature of these AI models. Users can input their specific legal circumstances, and the system will generate outputs tailored to their situation. This dynamic interaction helps to personalize legal support, making it more accessible and user-friendly.

Future research in this area will be aimed at advancing the automation of legal reasoning through several key initiatives.

A primary objective is to enhance the automatic formalization of legal rules, reducing reliance on human knowledge engineers. This involves developing frameworks that can autonomously interpret and represent legal texts in formal logic languages, facilitating automated reasoning processes. For instance, integrating deep learning with symbolic reasoning methods has shown promise in creating more robust and flexible legal reasoning systems capable of handling natural language ambiguities and rigid logical representations.

Another focus is the improved integration of case law into our models. By extracting rules from decision trees and ontologies, we aim to transition from merely identifying factors to formulating actionable legal rules. This approach seeks to enhance the system's ability to apply precedents effectively, thereby enriching the model's reasoning capabilities. Research indicates that combining case-based reasoning with object-oriented rule-based systems can lead to more effective legal reasoning models.

Lastly, we plan to develop chatbot-like systems capable of automatically identifying relevant elements from factual case descriptions. These systems would utilize natural language processing to parse and analyze legal texts, providing users with accessible and understandable legal information. Such advancements could democratize access to legal knowledge, enabling individuals to navigate complex legal landscapes more effectively. The integration of large language models with logic programming is a promising avenue for enhancing the strategic capabilities of legal reasoning systems.

A critical aspect of our future work is to enhance the explainability of our legal reasoning systems. Providing clear and understandable explanations of the

---

system's conclusions is essential for users to comprehend the rationale behind legal decisions. This transparency is particularly important in contexts where users may need to appeal outcomes, as it enables them to identify specific areas of concern or misunderstanding. By focusing on explainability, we aim to empower users with the necessary information to challenge decisions effectively, thereby promoting fairness and accountability in legal processes.

---

---

# Bibliography

- [1] Charu C. Aggarwal. *Machine learning for text*. Springer, Cham, 2018.
- [2] Charu C Aggarwal and ChengXiang Zhai. A survey of text classification algorithms. In Charu C Aggarwal and ChengXiang Zhai, editors, *Mining text data*, pages p. 163–222. Springer, New York, 2012.
- [3] Ayyoub Ajmi. Revolutionizing access to justice: The role of ai-powered chatbots and retrieval augmented generation in self-help. *Brief*, 53(3), 2024.
- [4] Matthew Armitage. Trust, confidence, and automation: the isda master agreement as a smart contract. *Business Law Review*, 43(2), 2022.
- [5] Katie Atkinson, Trevor Bench-Capon, and Danushka Bollegala. Explanation in ai and law: Past, present and future. *Artificial Intelligence*, 289:103387, 2020.
- [6] Patrizio Bellan, Mauro Dragoni, and Chiara Ghidini. Process knowledge extraction and knowledge graph construction through prompting: A quantitative analysis. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, volume 1 of *SAC '24*, page 1634–1641. ACM, April 2024.
- [7] Emily M Bender and Alexander Koller. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198, 2020.
- [8] Floris Bex and Henry Prakken. On the relevance of algorithmic decision predictors for judicial decision making. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 175–179, 2021.
- [9] Marco Billi and Alessandro Parenti. Access to justice through ai. In *Facilitating Judicial Cooperation in the EU. A computable approach to mutual recognition in criminal matters*, pages 1–15. Brill Nijhoff, forthcoming.
- [10] Marco Billi, Alessandro Parenti, Giuseppe Pisano, and Marco Sanchi. A hybrid approach for accessible rule-based reasoning through large language models. *18th International Workshop on Juris-Informatics*, Jurisin, 2024. Preprint Version.
- [11] Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph P. Turian. Experience grounds language. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8718–8735. Association for Computational Linguistics, 2020.

- [12] Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. Can gpt-3 perform statutory reasoning? In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 22–31, 2023.
- [13] P Bojanowski et al. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [14] Giorgio Bongiovanni, Gerald Postema, Antonino Rotolo, Giovanni Sartor, Chiara Valentini, and Douglas Walton. *Handbook of legal reasoning and argumentation*. Springer, 2018.
- [15] Roberta Calegari, Giuseppe Pisano, Andrea Omicini, and Giovanni Sartor. Arg2P: An argumentation framework for explainable intelligent systems. *Journal of Logic and Computation*, 32(2):369–401, March 2022. Special Issue from the 35th Italian Conference on Computational Logic (CILC 2020).
- [16] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *ACM Trans. Intell. Syst. Technol.*, 15(3):39:1–39:45, 2024.
- [17] Haihua Chen, Lei Wu, Jiangping Chen, Wei Lu, and Junhua Ding. A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, 59(2):102798, 2022.
- [18] Pawel Cichosz. *Data mining algorithms: explained using R*. John Wiley & Sons, Chichester, 2015.
- [19] Giuseppe Contissa and Giulia Lasagni. When it is (also) algorithms and ai that decide on criminal matters: In search of an effective remedy. *European Journal of Crime, Criminal Law and Criminal Justice*, 28(3):280–304, 2020.
- [20] Thiago Dal Pont, Federico Galli, Andrea Loreggia, Giuseppe Pisano, and Riccardo Rovatti. L’ia generativa per la sintesi automatica dei testi giuridici. *ParadoXa: XVIII, 1, 2024*, pages 121–134, 2024.
- [21] Thiago Raulino Dal Pont et al. Classification and association rules in brazilian supreme court judgments on pre-trial detention. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 131–142. Springer, 2021.
- [22] A. De Bellis. Structuring the unstructured: an llm-guided transition. In C. d’Amato and J.Z. Pan, editors, *Proceedings of the Doctoral Consortium at ISWC 2023 co-located with 22nd International Semantic Web Conference (ISWC 2023)*, volume 3678 of *CEUR Workshop Proceedings*, Athens, Greece, 2023.
- [23] J Demšar et al. Orange: data mining toolbox in python. *Journal of Machine Learning Research*, 14(1):2349–2353, 2013.
- [24] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. Ensemble methods for improving extractive summarization of legal case judgements. *Artificial Intelligence and Law*, pages 1–59, 2023.
- [25] Haoling Dong and Bin Wu. Enhancing named entity recognition in safety hazard analysis through gbd and llms. In *2024 7th International Conference on Information and Computer Technologies (ICICT)*, page 13–19. IEEE, March 2024.
- [26] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*, 2017.

- [27] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.
- [28] Mauro Dragoni, Serena Villata, Williams Rizzi, and Guido Governatori. Combining natural language processing approaches for rule extraction from legal documents. In Ugo Pagallo, Monica Palmirani, Pompeu Casanovas, Giovanni Sartor, and Serena Villata, editors, *AI Approaches to the Complexity of Legal Systems - AICOL International Workshops 2015-2017: AICOL-VI@JURIX 2015, AICOL-VII@EKAW 2016, AICOL-VIII@JURIX 2016, AICOL-IX@ICAIL 2017, and AICOL-X@JURIX 2017, Revised Selected Papers*, volume 10791 of *Lecture Notes in Computer Science*, pages 287–300. Springer, 2017.
- [29] RA Duff. Pre-trial detention and the presumption of innocence. *Oxford University Press, Minnesota Legal Studies Research Paper*, 12-31, 2012.
- [30] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council.
- [31] Alexandre Goossens and Jan Vanthienen. Integrating gpt-technologies with decision models for explainability. In *World Conference on Explainable Artificial Intelligence*, pages 428–448. Springer, 2023.
- [32] Thomas F. Gordon, Guido Governatori, and Antonino Rotolo. Rules and norms: Requirements for rule interchange languages in the legal domain. In *Lecture Notes in Computer Science*, pages 282–296. Springer Berlin Heidelberg.
- [33] Guido Governatori, Michael J Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14(5):675–702, 2004.
- [34] Morgan A. Gray, Jaromir Savelka, Wesley M. Oliver, and Kevin D. Ashley. Empirical legal analysis simplified: reducing complexity through automatic identification and evaluation of legally relevant factors. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 382(2270), February 2024.
- [35] Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, K. Aditya, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John J. Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael A. Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- [36] Joseph F. Hair, Bill Black, William C. Black, Barry J. Babin, and Rolph E. Anderson. *Multivariate data analysis*. Cengage Learning, Andover, 2019.
- [37] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000.

- [38] John Horty. Reasoning with dimensions and magnitudes. *Artificial Intelligence and Law*, 27(3):309–345, 2019.
- [39] John F Horty and Trevor JM Bench-Capon. A factor-based definition of precedential constraint. *Artificial intelligence and Law*, 20(2):181–214, 2012.
- [40] Weifeng Hu, Siwen Zhao, Qiang Zhao, Hao Sun, Xifeng Hu, Rundong Guo, Yujun Li, Yan Cui, Long Ma, et al. Bert\_lf: A similar case retrieval method based on legal facts. *Wireless Communications and Mobile Computing*, 2022, 2022.
- [41] Xia Hu and Huan Liu. Text analytics in social media. In Charu C. Aggarwal and Cheng Xiang Zhai, editors, *Mining text data*, pages 385–414. Springer, New York, 2012.
- [42] Samyar Janatian, Hannes Westermann, Jinzhe Tan, Jaromír Savelka, and Karim Benyekhlef. From text to structure: Using large language models to support the development of legal expert systems. In Giovanni Sileno, Jerry Spanakis, and Gijs van Dijck, editors, *Legal Knowledge and Information Systems - JURIX 2023: The Thirty-sixth Annual Conference, Maastricht, The Netherlands, 18-20 December 2023*, volume 379 of *Frontiers in Artificial Intelligence and Applications*, pages 167–176. IOS Press, 2023.
- [43] Jørgen Karpf. *Quality Assurance of Legal Expert Systems*. Pre-proceedings of the third international conference on Logica, Informatica, Diritto. CNR, 1989.
- [44] Daniel Martin Katz, Michael J. Bommarito, and Josh Blackman. A general approach for predicting the behavior of the supreme court of the united states. *PLOS ONE*, 12(4):e0174698, April 2017.
- [45] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270):20230254, 2024.
- [46] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
- [47] Vijay Kotu and Bala Deshpande. *Data Science*. Morgan Kaufmann (Elsevier Science), Cambridge, MA, 2 edition, 2019.
- [48] Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE international test conference (ITC)*, pages 1–8. IEEE, 2018.
- [49] Marco Lippi, Przemysław Pałka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, and Paolo Torroni. Claudette: an automated detector of potentially unfair clauses in online terms of service. *Artificial Intelligence and Law*, 27:117–139, 2019.
- [50] Shuaiqi Liu, Jiannong Cao, Yicong Li, Ruosong Yang, and Zhiyuan Wen. Low-resource court judgment summarization for common law systems. *Information Processing & Management*, 61(5):103796, September 2024.
- [51] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.
- [52] Andrew Lymer and Ken Richards. A hybrid-based expert system for personal pension planning in the uk. *Intelligent Systems in Accounting, Finance and Management*, 4(1):71–88, 1995.



## BIBLIOGRAPHY

---

- [53] Naser Masri, Yousef Abu Sultan, Alaa N Akkila, Abdelbaset Almasri, Adel Ahmed, Ahmed Y Mahmoud, Ihab Zaout, and Samy S Abu-Naser. Survey of rule-based systems. *International Journal of Academic Information Systems Research (IJAIRS)*, 3(7):1–23, 2019.
- [54] Michael EJ Masson and Mary Anne Waldron. Comprehension of legal contracts by non-experts: Effectiveness of plain language redrafting. *Applied cognitive psychology*, 8(1):67–85, 1994.
- [55] Robert H Michaelsen. An expert system for federal tax planning. *Expert Systems*, 1(2):149–167, 1984.
- [56] Hans-W Micklitz, Oreste Pollicino, Amnon Reichman, Andrea Simoncini, Giovanni Sartor, and Giovanni De Gregorio. *Constitutional challenges in the algorithmic society*. Cambridge University Press, 2021.
- [57] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [58] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.
- [59] Sanjay Modgil and Henry Prakken. The aspic+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.
- [60] Luca Nannini. Habemus a right to an explanation: so what?—a framework on transparency-explainability functionality and tensions in the eu ai act. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 1023–1035, 2024.
- [61] John J. Nay, David Karamardian, Sarah B. Lawsky, Wenting Tao, Meghana Bhat, Raghav Jain, Aaron Travis Lee, Jonathan H. Choi, and Jungo Kasai. Large language models as tax attorneys: a case study in legal capabilities emergence. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 382(2270), February 2024.
- [62] Ha Thanh Nguyen, Randy Goebel, Francesca Toni, Kostas Stathis, and Ken Satoh. Lawgiba - combining gpt, knowledge bases, and logic programming in a legal assistance system. In Giovanni Sileno, Jerry Spanakis, and Gijs van Dijck, editors, *Legal Knowledge and Information Systems - JURIX 2023: The Thirty-sixth Annual Conference, Maastricht, The Netherlands, 18-20 December 2023*, volume 379 of *Frontiers in Artificial Intelligence and Applications*, pages 371–374. IOS Press, 2023.
- [63] Ha-Thanh Nguyen, Fumihito Nishino, Megumi Fujita, and Ken Satoh. An interactive natural language interface for proleg. In *JURIX*, pages 294–297, 2022.
- [64] OpenAI. Hello gpt-4o, May 2024. Accessed: 2024-09-12.
- [65] Stanisław Osiński, Jerzy Stefanowski, and Dawid Weiss. Lingo: search results clustering algorithm based on singular value decomposition. In Mieczysław A Kłopotek, Sławomir T Wierzchoń, and Krzysztof Trojanowski, editors, *Intelligent information processing and web mining*, pages 359–368. Springer, Berlin, Heidelberg, 2004.
- [66] Stanisław Osiński and Dawid Weiss. Carrot2, 2019.
- [67] Cecilia Panigutti, Andrea Beretta, Fosca Giannotti, and Dino Pedreschi. Understanding the impact of explanations on advice-taking: a user study for ai-based clinical decision support systems. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–9, 2022.

- [68] Christos H Papadimitriou et al. Latent semantic indexing. *Journal of Computer and System Sciences*, 61(2):217–235, 2000.
- [69] Henry Prakken and Giovanni Sartor. Law and logic: A review from an argumentation perspective. *Artificial Intelligence*, 227:214–245, 2015.
- [70] Marc Queudot, Éric Charton, and Marie-Jean Meurs. Improving access to justice with legal chatbots. *Stats*, 3(3):356–375, 2020.
- [71] Karen McGregor Richmond, Satya M Muddamsetty, Thomas Gammeltoft-Hansen, Henrik Palmer Olsen, and Thomas B Moeslund. Explainable ai and law: an evidential survey. *Digital Society*, 3(1):1, 2024.
- [72] Isabela Cristina Sabo, Marco Billi, Francesca Lagioia, Giovanni Sartor, and Aires José Rover. Unsupervised factor extraction from pretrial detention decisions by italian and brazilian supreme courts. In *Advances in Conceptual Modeling: ER 2022 Workshops, CMLS, EmpER, and JUSMOD, Hyderabad, India, October 17–20, 2022, Proceedings*, pages 69–80. Springer, 2023.
- [73] Giovanni Sartor, Michał Araszkiewicz, Katie Atkinson, Floris Bex, Tom van Engers, Enrico Francesconi, Henry Prakken, Giovanni Sileno, Frank Schilder, Adam Wyner, et al. Thirty years of artificial intelligence and law: the second decade. *Artificial Intelligence and Law*, 30(4):521–557, 2022.
- [74] Jaromir Savelka, Kevin D Ashley, Morgan A Gray, Hannes Westermann, and Huihui Xu. Explaining legal concepts with augmented large language models (gpt-4). *arXiv preprint arXiv:2306.09525*, 2023.
- [75] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.
- [76] M. J. Sergot, F. Sadri, R. A. Kowalski, F. Kriwaczek, P. Hammond, and H. T. Cory. The british nationality act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986.
- [77] Juanming Shi, Qinglang Guo, Yong Liao, Yuxing Wang, Shijia Chen, and Shenglin Liang. *Legal-LM: Knowledge Graph Enhanced Large Language Models for Law Consulting*, page 175–186. Springer Nature Singapore, 2024.
- [78] Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, and Silas Franco dos Reis Alves. *Artificial neural networks*. Springer, Switzerland, 2018.
- [79] Zhongxiang Sun. A short survey of viewing large language models in legal aspect. *arXiv preprint arXiv:2303.09136*, 2023.
- [80] Dietrich Trautmann, Alina Petrova, and Frank Schilder. Legal prompt engineering for multilingual legal judgement prediction. *arXiv preprint arXiv:2212.02199*, 2022.
- [81] William van Melle. Mycin: a knowledge-based consultation program for infectious disease diagnosis. *International Journal of Man-Machine Studies*, 10(3):313–322, 1978.
- [82] Anne L Washington. How to argue with an algorithm: Lessons from the compas-propublica debate. *Colo. Tech. LJ*, 17:131, 2018.
- [83] Hannes Westermann and Karim Benyekhlef. Justicebot: A methodology for building augmented intelligence tools for laypeople to increase access to justice. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 351–360, 2023.

## BIBLIOGRAPHY

---

- [84] Xiaoxian Yang, Zhifeng Wang, Qi Wang, Ke Wei, Kaiqi Zhang, and Jiangang Shi. Large language models for automated q&a involving legal documents: a survey on algorithms, frameworks and applications. *International Journal of Web Information Systems*, 20(4):413–435, April 2024.
- [85] Yanru Zhang and Ali Haghani. A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58:308–324, 2015.
- [86] Wei Zhao, Qinghui Chen, and Junling You. Llmre: A zero-shot entity relation extraction method based on the large language model. In *Proceedings of the 2023 7th International Conference on Electronic Information Technology and Computer Engineering*, volume 35 of *EITCE 2023*, page 475–480. ACM, October 2023.