PhD-FSTM-2020-54
The Faculty of Sciences, Technology and Medicine

DISSERTATION

Defence held on 05/10/2020 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN *INFORMATIQUE*

by

Amin SLEIMI
Born on 23 February 1992 in Tabarka, (Tunisia)

AN NLP-BASED FRAMEWORK TO FACILITATE THE DERIVATION OF LEGAL REQUIREMENTS FROM LEGAL TEXTS

Dissertation defence committee

Dr. Mehrdad Sabetzadeh, dissertation supervisor
Senior Research Scientist, Université du Luxembourg

Dr. Travis Breaux
Associate Professor, Carnegie Mellon University (US)

Dr. Jacques Klein, Chairman
Associate Professor, Université du Luxembourg

Dr. Jennifer Horkoff
Associate Professor, Chalmers University of Technology
Senior Lecturer, University of Gothenburg (Sweden)

Dr. Lionel Briand, Vice Chairman
Professor, Université du Luxembourg

Acknowledgements

The African proverb says that if you want to go fast go alone but if you want to go far go together. This work would not have been possible without the support and encouragement of a number of people in my life.

First of all, I would like to thank my supervisor Mehrdad Sabetzadeh for your patient guidance, enthusiastic encouragement, and useful critiques of this research work. Thank you for being a great mentor both professionally and personally. I quite appreciate your thoughtful guidance throughout my Ph.D.

I am grateful to my co-supervisor Lionel Briand for his support and guidance. Thank you for giving me the opportunity to conduct research under your direction in this exciting line of research. Your advice and recommendations have been a great help towards this milestone.

I am particularly grateful for the assistance of Marcello Ceci. Your advice on both research and philosophy of life has been invaluable. Thank you for your support through the ups and downs of this past couple of years.

I would like to offer special thanks to Nicolas Sannier for his continued support. I learned a lot from working with you.

I also wish to acknowledge the help and recommendations of Travis Breaux throughout my doctoral studies. Our meetings were vital in inspiring me to think outside of the box.

I would like to express my gratitude to the Central Legislative Service in Luxembourg for this opportunity. Special thanks to John Dann for always being ready to help and for providing valuable insights.

I am grateful for the members of my defense committee for their valuable insights and suggestions.

My special thanks are extended to my colleagues in the Software Verification and Validation lab.

Finally, I would like to thank my friends and family for their unyielding support. Thank you for believing in me. Thank you for always being there for me.

Abstract

Information systems in several regulated domains (e.g., healthcare, taxation, labor) must comply with the applicable laws and regulations. In order to demonstrate compliance, several techniques can be used for assessing that such systems meet their specified legal requirements. Since requirements analysts do not have the required legal expertise, they often rely on the advisory of legal professionals. Hence, this paramount activity is expensive as it involves numerous professionals. Add to this, the communication gap between all the involved stakeholders: legal professionals, requirements analysts and software engineers. Several techniques attempt to bridge this communication gap by streamlining this process. A promising way to do so is through the automation of legal semantic metadata extraction and legal requirements elicitation from legal texts. Typically, one has to search legal texts for the relevant information for the IT system at hand, extract the legal requirements entailed by these legal statements that are pertinent to the IT system, and validate the conclusiveness and correctness of the finalized set of legal requirements.

Nevertheless, the automation of legal text processing raises several challenges, especially when applied to IT systems. Existing Natural Language Processing (NLP) techniques are not built to handle the peculiarities of legal texts. On the one hand, NLP techniques are far from perfect in handling several linguistic phenomena such as anaphora, word sense disambiguation and delineating the addressee of the sentence. Add to that, the performance of these NLP techniques decreases when applied to foreign languages (other than English). On the other hand, legal text is far from being identical to the formal language used in journalism. We note that the most prominent NLP techniques are developed and tested against a selection of newspapers articles. In addition, legal text introduces cross-references and legalese that are paramount to proper legal analysis. Besides, there is still some work to be done concerning topicalization, which we need to consider for the relevance of legal statements.

Existing techniques for streamlining the compliance checking of IT systems often rely on code-like artifacts with no intuitive appeal to legal professionals. Subsequently, one has no practical way to double-check with legal professionals that the elicited

legal requirements are indeed correct and complete regarding the IT system at hand. Further, manually eliciting the legal requirements is an expensive, tedious and error-prone activity. The challenge is to propose a suitable knowledge representation that can be easily understood by all the involved stakeholders but at the same time remains cohesive and conclusive enough to enable the automation of legal requirements elicitation.

In this dissertation, we investigate to which extent one can automate legal processing in the Requirements Engineering context. We focus exclusively on legal requirements elicitation for IT systems that have to conform to prescriptive regulations. All our technical solutions have been developed and empirically evaluated in close collaboration with a government entity.

Acknowledgments. Financial support for this work was provided by Luxembourg National Research Fund (FNR) under grant number PUBLIC2-17/IS/11801776.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Premise

Written language contains knowledge that makes several aspects of human activities possible or renders them simpler. The information contained in texts is however often complex and nested. Several activities in the modern world have an inherent need to efficiently handle information expressed in texts, and as a consequence text processing is at the core of such activities. Because of its potential to automate text processing, Natural Language Processing (NLP) has been hailed as the efficient future technology to enable or simplify all those activities that rely heavily on the extraction of information from written texts.

The past few years have witnessed important advances in NLP. Driven by the rapid increase in available data and computational resources, NLP became a staple technology in tackling fundamental tasks ranging from Machine Translation to Question Answering and summarization. This was enabled by the remarkable developments in computational semantics and semantic parsing. Many of these fundamental tasks are relevant to a much broader array of domains beyond computational linguistics. In particular, NLP has a tremendous potential in automating several monotonous tasks (given the appropriate reformulation) with performance metrics reaching those of a human expert. Nowadays, we rely on NLP more and more for different tasks: this technology is replacing humans in the processing of non-narrative texts, with 2.5 quintillion bytes of data created each day [1].

A legal text is a particular type of text and as such all activities that revolve around its understanding are likely to be revolutionized by the above-mentioned advances in NLP. Legal texts, however, pose specific challenges because of the effects that

different interpretations have on the external reality, and the principles that oversee such alternative interpretations.

We note that law is a social phenomenon which emerged very early in human societies. Although primitive law was not formally structured, it shares the same roots of our modern days' legal systems. In fact, law always relies on the concept of a legitimate order expressing the will of a recognized authority, be that the strongest man in the tribe, the deity's chosen one or a democratically elected Parliament.

Since the code of Hammurabi (1760 BC), legislative acts have been an integral part of early human societies, turning individuals into subjects by bringing new implications to their actions that do not only depend on the tenets of nature and physics. Later on, legal systems evolved beyond the mere imposition of duties and protection of rights, turning subjects into citizen and taking a central place in human societies as "a framework for the conduct of almost every social, political, and economic activity" [2].

Throughout history, legislative acts have been always expressed in words, which have been kept in writing for purposes of documentation and publicity. However, human language is often ambiguous, in that it is prone to multiple interpretations which are often in contradiction with each other, and thus mutually exclusive. In the case of the legal language, this vagueness (or open texture) is not only a consequence of the limitations of the physical conveyor of the message, i.e., written language: it is rather an intrinsic characteristic of the law (we can call it "vagueness by design"), that derives from different reasons:

- As mentioned before, the law is essentially an order expressing the will of an authority [3]. As such, it always has an element of arbitrariness in order to allow the authority to adapt an order in the presence of new or unexpected situations, so that it always matches his or her will. For this reason, legislative texts always have an in-built vagueness that allows the authority to enforce its will across a variety of concrete situations, without constraining itself a-priori: an authority totally constrained by rules, unable to enforce its will, is not a sovereign but only a bureaucrat. Modern societies, with the creation of the so-called "rule of law", introduced limits to the original arbitrariness of the authority. This arbitrariness was however not removed altogether, and modern laws are often left vague in order to reach agreement within legislative bodies, or to defer contrasts to other places (e.g. courts of justice, administrative bodies, or successive legislative processes for local or delegated acts) which can then exert the "arbitrariness" or "enforcement of the will" previously granted to sovereigns.

- The Roman philosopher Cicero affirms that for a just legal text: "it shall be of universal application, unchanging and everlasting". Also in modern legal systems the law is meant to be generic and abstract, in order to guarantee that situations that are similar in substance are treated similarly. Legislative texts are thus written in a general and abstract format, simplifying reality by means of approximations and generalizations. This means that legislative texts do not represent all the subtleties present in specific cases, in order to leave room for adapting its implications to new or unexpected situations.

- The law is a conceptualization of reality, but the law does not describe the reality that is: it prescribes the reality that ought to be. Legal acts create new entities that do not exist in the physical world, but that arise out of phenomena in the physical world [4]. Because as humans we don't have a way to represent reality in a complete and unique way, the effects of a legal prescription on the real world cannot be outlined precisely and therefore can only be expressed as an approximation of reality.

We can say that law is controversial by nature, imperfect but indispensable as a primitive, spontaneous and pervasive social phenomenon. There are gaps and overlaps in the prescriptions expressed by legal texts, and they cannot be always sorted with mathematical certainty. In addition, modern-day law is highly technical in practice, with its obscure legal jargon, specific semantics, articulated procedures, complex hierarchy of sources, and proliferating jurisprudence. All these characteristics make it very difficult to provide a legal interpretation of a law, which means to predict the way in which the law will be applied to a concrete fact. The vagueness and unpredictability have remained, even though it is not anymore (or at least less than before) to the advantage of a selected few.

Modern legal systems are divided into two categories: civil law systems and common law systems. In civil law systems, the legislative text is seen as a complete set encompassing all possible real-world situations, and therefore the judge can only fill gaps and resolve conflicts by interpreting existing laws, not by creating new ones. In common law systems, in case of gaps or conflicts in the "statutes" (legislative texts), the judge has the power to create a new ruling, which however should have a certain degree of analogy to previous authoritative rulings. In practice, however, both civil- and common-law systems share the common characteristic of relying on the interpretation of legal sources (legislative texts and judicial decisions) to infer implications for every possible real-world situation.

In addition to the interpretation of the law performed by judges, other people such as lawyers and politicians often have to argue for or against certain interpretations of a legal text, in order to foster the interpretation that best suits their (or their clients') interests. Interpreting a legal text is always a complex and delicate activity, whether it is done in a lawyer's office (in an extra-judicial and/or pre-judicial phase) or in court (in the judicial phase). Globalization added further strain to these challenges: rapid advances in technology and its integration in everyday's life, together with the sheer amount of administrative regulations issued to keep up with the multiplication of interactions across the world, raise the need for contemporary legal professionals to enhance and rationalize their approach to the interpretation of (the management of the knowledge contained in) legal texts.

NLP can help untangle this complexity, but it must be understood that the challenges posed by legal texts are inherent to the very nature of the law, and cannot be downplayed to the basic text processing challenge of understanding the semantics of a written sentence.

1.2 Objectives

In this research, we investigate selected facets of the automated processing of legal texts. As noted in the previous section, the law is not a complete and consistent body of rules and doctrine. In addition, the legal domain covers a vast variety of domains ranging from criminal offences, environment, commerce, finance, health, up to civil matters such as marriage and succession. These challenges entail that an over-arching solution cannot be found that tackles all aspects of legal interpretation.

In the field of automated processing of legal texts, a lot of work has been devoted to providing automated support to the judicial phase, with most work focusing on applications to assist the judge: examples include applications for the automatic adjudication of cases, for pre-sorting of cases of similar nature, and for e-discovery. To the best of our knowledge, research related to the non-judicial phase (applications for automated legal advice and e-government) did not go beyond the proof-of-concept state to yield a usable approach in the real world.

In this work, we focus on the non-judicial phase, examining the feasibility of standardized legal advice. As a matter of fact, legal advice is non-binding. When provided by legal professionals, this advice is specialized, to enable the clients to evaluate the legal implications of a specific course of action. When provided by governments, legal advice is more generic as it clarifies the way in which the law is

going to be applied, thus increasing the accessibility of the law for the citizens. In the past, circulars and brochures have been used for this latter type of legal advice. In recent times, governments provide advice in their institutional web portals, through the use of Frequently Asked Questions pages (FAQs) and summary sheets. Another example of legal advice is the elicitation of legal requirements for IT systems. Information systems in several regulated domains (e.g., healthcare, taxation, labor) must comply with the applicable laws and regulations. In order to ensure compliance, several techniques can be used for assessing whether such systems meet their specified legal requirements. Automation of legal requirements elicitation can not only solve practical problems in the software engineering field, but also provide the basis for automating other instances of legal advice, such as the advice brought by governments to their citizen.

In the field of requirements engineering, since requirements analysts do not have the required legal expertise, they often rely on the advice of legal professionals when dealing with legal requirements. This renders the activity of legal requirements elicitation rather expensive, as it involves numerous professionals. Furthermore, the collaboration is prone to misunderstandings due to the communication gap that exists between the involved stakeholders (legal professionals, requirements analysts and software engineers). Several techniques attempt to bridge this communication gap by streamlining the requirements elicitation process. Typically, this activity involves (1) browsing legal texts for the relevant legal statements applicable to the IT system at hand, (2) extracting the legal requirements expressed by these legal statements, and (3) validating the completeness and correctness of the resulting set of legal requirements.

The goal of this research is to automate the extraction of semantic metadata for the automatic recommendation of legal requirements for IT systems. The research is based on recent advances in NLP, on established approaches for requirements engineering and on modelling approaches provided by AI and Law.

1.3 Challenges

In practice, the automation of legal text processing raises several challenges, especially when applied to IT systems.

1.3.1 Issues related to NLP

Natural Language Processing (NLP) has proven to be a suitable toolset for the automation of handling information in textual format. These techniques have been

iteratively developed over annotated datasets to capture specifically tailored bits and pieces of linguistics in the underlying text. However, existing NLP techniques are not built to handle the peculiarities of legal text. Actually, multi-lingual NLP techniques are far from perfect in handling several linguistic phenomena such as anaphora, word sense disambiguation and delineating the addressee of the sentence. More specifically, we have similar phenomena in the legal text such as exceptions, party to the law, inclusion and extension of the addressee for different provisions. Without a model of these concepts, we cannot have the semantics of the legal texts. In addition, some the techniques like topicalization are not at a state where we can immediately use their intermediary results (not a plug and play). However, these techniques can be used to support the automated processing of the legal text for the relevance of legal statements to a specific IT-related topic. Add to that, the performance of these NLP techniques decreases when applied to foreign languages (other than English). We note that the most prominent NLP techniques are developed and tested against a selection of newspapers articles. Additionally, legal text is far from being identical to the formal language used in journalism. Hence, the issues of NLP above are more of a challenge specifically for the legal text.

1.3.2 Issues specific to legal texts

As explained in the beginning of this chapter, extracting the semantics of a legal text is an act of legal interpretation, which does not have the predictability or precision of formal sciences. More specifically, legal texts introduce cross-references that need to be analyzed in order to understand the meaning of a legal provision. Also, the legal rules expressed by the text make use of deontic modalities and logical formulations. The modality expressed in the rule may affect the meaning of the sentence, but it is also possible that the action and cross-references affect the modality. As a result, identifying the modality and interpretation of a specific modality in the context of the provision (how the modality affects the meaning of the sentence) is peculiar to the legal text.

1.3.3 Issues related to the complexity of legal interpretation

Existing techniques for streamlining the compliance checking of IT systems often rely on code-like artifacts with no intuitive appeal to legal professionals. There is a gap between the legal expertise and the technical expertise. This gap introduces the following challenges:

- Translating a rule expressed by the law into a component inside the software is not a straightforward activity. Subsequently, one has no practical way to double-check with legal professionals that the elicited legal requirements are indeed correct and complete regarding the IT system at hand. As a matter of fact, previous research attempted to fill this communication gap, using different approaches to address the lack of information: Some efforts were directed at reformulating the law into code; others tried the opposite, namely to write software in a similar way to the law. Other approaches devised a controlled language as an intermediary, or protocols of collaboration to streamline the collaboration between these two communities.

- Manually eliciting the legal requirements is an expensive and complex activity. Overall, the legal domain is very large. Hence, manual support is very expensive (manual annotation) and prone to interpretation (different judges and different jurisdictions). As a matter of fact, a new legal concept does not translate to a single new legal rule. Take the example of a new legislation of "electronic scooters". We note that legislation already accounts for other pre-established categories of vehicles like bikes or mopeds. These concepts do have their own assigned legal rules. The decision to assign the new concept to one of these categories entails several legal rules (those of the assigned category alongside other rules that apply by means of inference). Actually, mopeds are also categorized as cars and therefore some but not all regulations for cars apply to these vehicles. Shall we assign these regulations to the new concept as well?

The challenge here is to propose a suitable knowledge representation that can be easily understood by all the involved stakeholders but at the same time remains cohesive and conclusive enough to enable the automation of legal requirements elicitation.

1.4 Contributions

In this dissertation, we investigate to which extent one can automate legal processing in the Requirement Engineering context. We focus on legal requirements elicitation for IT systems that have to conform to prescriptive regulations. All our technical solutions have been developed and empirically evaluated in close collaboration with a government entity.

Concretely, the technical solutions presented in this dissertation include:

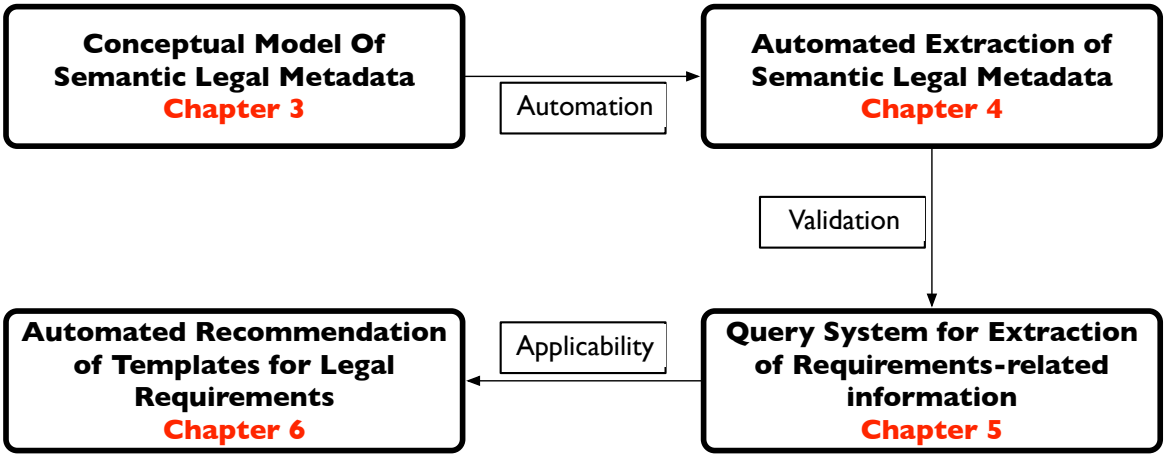

Fig. 1.1 Research Overview and Organization

- A conceptual framework for semantic legal metadata: We propose a conceptual model for the abstract building blocks of legal text. While the research community acknowledges the importance of semantic legal metadata, there is no consensus on the metadata types that are beneficial for legal compliance analysis. Indeed, these conceptualizations are at different levels of abstraction, depending on the targeted analysis as well as on the desired degree of interpretation. By looking at the literature, we have identified these conceptualizations and performed a mapping that reconciles these works into a general, high-level conceptual model that we deem general enough to be domain-independent, along with a precise definition for each of its elements.

- Automated extraction of semantic legal metadata: Given the established conceptual model, we devised extraction rules for the elements of the conceptual model through several qualitative studies and case studies performed over six legislative domains, including: traffic law, commerce law, environmental law, health law, penal law, and labor law. The extraction of semantic metadata is realized through subjecting individual legal statements to automated analysis, leveraging Natural Language Processing (especially constituency parsing and dependency parsing) and Machine Learning.

- Query system for extracting requirements-related information from legal text: We built a query system to streamline the validation of the automatically extracted semantic legal metadata. This is an advanced search facility over regulations. We showcase that semantic legal metadata can be successfully leveraged to answer

requirements engineering-related questions. Hence, this query system enables resolving the relevance challenge. At the same time, the experience pinpoints for further improvements to the conceptual framework of semantic legal metadata.

- Automated recommendation of templates for legal requirements: We propose an approach to automatically recommend templates for legal requirements based on legal statements, thus assisting requirements analysts with legal requirements elicitation. We investigate the use of requirements templates for the systematic elicitation of legal requirements. Subsequently, we conduct a qualitative study to define NLP rules for template recommendation.

Chapter 2

Background and Related Work

This chapter provides information about the background of the research and the related work. The work in this chapter is organized in relation to three different research communities: (1) Requirements engineering, (2) Automated linguistics, and (3) Legal informatics.

2.1 Natural Language Requirements

Software requirements come in different forms and representations. The most well known one is the IEEE-830 style "shall" requirements. This practice describes recommended approaches for the specification of software requirements. We note that the underlying software behavior shall be prescribed in an ambiguous and complete Software Requirements Specification (SRS). SRS serve two essential activities:

1. Assist software users (software customers) in explaining their needs in terms of software functionalities. SRS accurately describe what the customers wish to obtain from the software.

2. Assist software engineers (software suppliers) in understanding the customers' needs in terms of software functionalities and inner-workings. SRS determine exactly what shall be implemented for the software.

2.1.1 Requirements Templates

The requirements engineering community researched the real-world outcome and pitfalls of software specification. Several IT projects failed due to inconsistencies and gaps within the software. The Arianne 5, which cost 8 Billion $ and blew in space due to a

software error, is still a warning to the failing methodologies of specifying requirements. Researchers attempted to formalize the prescription of requirements in a complete, consistent and understandable format through the use of templates. In this subsection, we present a selection of work that attempted to formalize the use of templates within the requirements engineering community.

Considerable work has been devoted to structuring requirements through template suggestion. Palomares et al. [5] report on the use of patterns in RE in a comprehensive survey. First attempts include Robertson's study [6] on "how event/use case modelling can be used to identify, define and access requirements patterns" and Dwyer et al.'s set of templates [7] for the specification of verifiable requirements through state machines. The latter involves manual mapping and transformation of requirements into logical expressions. More recently, Mavin et al. [8] present the Easy Approach to Requirements Syntax (EARS). EARS templates have a high-level perspective, and do not adequately account for actors and stakeholders other than the IT system.

2.1.2 Legal Natural Language Requirements

SRS is a legally binding document that is agreed upon by the customer and the supplier. However, IT systems themselves in several domains have been regulated by different jurisdictions. This raises the question of software compliance, which concerns the legal validity and scope of the software to be implemented. Therefore, legal requirements are an inclusive part of software requirements specifications. Several work attempted to formally characterize the legal aspects and components of SRS. In this subsection, we give an overview of the multiple strands of research in the requirements engineering community dealing with legal implications of software.

Several contributions are specifically aimed at capturing legal requirements. Breaux & Gordon [9] present a list of generic templates to highlight information within legal provisions in the Legal Requirement Specification Language (LRSL). LRSL is aimed at encoding legal provisions for developers and policy makers. It accounts for conditions, actions, the syntactic structure of the legal provision, and the different stakeholders of the IT system. In the previously mentioned work, Breaux et al. [10] present a methodology for extracting rights and obligations from regulations using a semantic model. They define a list of patterns for such rights and obligations. Young & Anton [11] present a list of templates for translating provisions into legal requirements. These templates have IT systems as their main focus and take into account the different stakeholders' viewpoints. Yoshida et al. [12] update the templates proposed by Young & Anton by adding templates for definitions and processing data objects as first-class

components. Although they present a method for automatically suggesting templates, the implementation has important limitations, the most notable being its exclusive focus on functional requirements, thus not accounting for non-functional and quality requirements.

Contributions from AI and Law focus on representing legal requirements with logical rules rather than templates. LegalRuleML [13] is a rule language that classifies statements into facts and norms, further specialized into constitutive, prescriptive, and penalty statements. LegalRuleML provides a solution to accurately express complex legal rules, but it is not supported by automatic extraction of concepts.

2.2 Natural Language Processing

In the last few years, Artificial Intelligence took the world by a storm through a set of advances that enabled the automation of several tasks in business and even entertainment. Natural Language Processing (NLP) is one of the prominent fields of AI. NLP concerns the automated processing of text and has been integrated into many frameworks we use on daily basis, from the automatic translation on social media to the automated email replies suggestions. All of these accomplishments and several others have been the result of digitization of linguistic tools that enabled to a certain extent the computers to process textual data in a similar way to humans. In what follows, we briefly present the body of research within the computerized linguistics community that we investigated and used throughout the the different phases of this project.

2.2.1 Constituency and Dependency Parsing in RE

As mentioned already, the main enabling NLP techniques we employ for metadata extraction are constituency and dependency parsing. In recent years, advanced NLP techniques, including constituency and dependency parsing, have generated a lot of traction in RE. Examples of problems to which these techniques have been applied are template conformance checking [14], model extraction [15, 16], feature extraction [17], and ambiguity and defect detection [18, 19].

In relation to legal requirements specifically, Bhatia et al. [20, 21] and Evans et al. [22] apply constituency and dependency parsing for analyzing privacy policies. These threads of work have provided us with useful inspiration. Nevertheless, our objective is different. Bhatia et al. and Evans et al. focus on detecting ambiguities in privacy

policies via the construction of domain-specific lexicons and ontologies. Our work, in contrast, addresses the extraction of metadata for facilitating the identification and specification of legal requirements. Our work aligns best with the GaiusT and NomosT initiatives discussed earlier. What distinguishes our work from these initiatives is providing wider coverage of metadata types and using NLP techniques that can more accurately delineate the spans for metadata annotations.

2.2.2 Semantic Role Labeling

As noted before, the potential of NLP technologies has increased with recent advancements. *Semantic Role Labeling* [23, 24] is the activity of assigning semantic roles to each of the predicate's arguments in a sentence. These roles usually capture the semantic commonality between instantiations of actors or artifacts across the language. The most notable contribution in the field is FrameNet [25], rooted in the theory of frame semantics. *Deep language analysis* [26] consists of using knowledge of linguistics to extract knowledge from text. It is a type of analysis that takes into account the nuances and complexities of linguistic constructs such as negation and conditionality. A *verb lexicon* is a lexical database of the different variations of syntactic representations of verbs in a sentence. VerbNet [27] is a verb lexicon that incorporates both semantic and syntactic information about verb types following Levin's classification of verbs [28].

POS-tagging. Part-of-Speech (POS) tagging [29] is the process of assigning part-of-speech tags (e.g., *noun, verb, propositional phrase*) to the different words in a sentence. The process is done following rule-based algorithms or stochastic techniques. In this work, we use a lexical resource in French called Lefff to automatically assign POS tags for the different tokens in a provision.

Tree pattern matching. Pattern matching is the process of verifying and locating the occurrence of a sequence of tokens, which follows a specified pattern, in a larger sequence. For tree pattern matching, we use Tregex [30] which validates and retrieves the sequences of trees or sub-trees that follow a specified pattern of POS tags within a parse tree of a sentence. Tregex is based on the inner-tree relationships among the different constituents of the parse tree.

Named Entity Recognition. Named Entity Recognition (NER) [29] is the process of locating and classifying the different named entities present in an input sentence into a pre-established set of categories. In this work, an named entity can be a person, organization, or place. This technique can be performed following rule-based

classification of entities or stochastic techniques (i.e., techniques based on the statistical probability of a token being an entity from one of the pre-established categories).

10-fold cross validation. 10-fold cross validation is the process of randomly dividing the dataset into 10 subdatasets. Nine of these subdatasets are used for ML training resulting in predictions that are eventually evaluated against the 10th subdataset. This process is repeated 10 times until the ML technique has been evaluated against all 10 subdatasets. This evaluation technique is widely used within the ML community to decrease the effects of over-fitting the model to the training set. We note that the overall evaluation of the ML technique in our work occurred against a new set of actors, as presented in the evaluation section for both case studies.

ML techniques for classification. Classification [31] is the process of assigning a category (a class) for a given data point. Classification has been a focus of the machine learning community since the inception of the field. Over the years, several techniques involving both statistical algorithms and symbolic rules have been proposed as supervised learning approaches, and incrementally enhanced.

In our work, we focus on four techniques used for classification: Naive Bayes classifiers, Decision Trees, Random Forests, and Support Vector Machines. A Naive Bayes classifier is a probabilistic algorithm based on Bayes's theorem. Decision Trees partition the feature space for optimal decision making. Random Forest is a technique relying on a set of diverse decision trees. Support Vector Machine (SVM) is a technique to classify data inputs in a high-dimensional space through a hyper plane that divides the training dataset into the categories of classification.

2.3 Semantic Legal Metadata and Legal Ontologies

Law was not spared from the AI revolution. Researchers investigated the use of Natural Language Processing tools, semantic web technologies, machine learning and visualization techniques to automate the mundane tasks performed by legal practitioners. This community of legal informatics consisted in a close collaboration between experts in the different fields of Computer science and experts from the legal domain (academics, lawyers and judges). The need for extracting the bits and pieces of information from the legislation or jurisprudence for the different legal activities was paramount, yet expensive and tedious at times. The solution was to extract these legal metadata and assemble (store/save) them in a standardized representation with formal naming, definition of clustered categories of these entities along with their properties and the intra- and inter-relationships between the different legal concepts. Hence, the

introduction of legal ontologies and the subsequent investigation of their practical uses and continuous enhancements made legal ontologies in general and semantic legal metadata in particular a cornerstone of the practice of law in the new era of digitization. In what follows, we discuss the different strands of work that we studied and considered for integration to our framework.

2.3.1 Preliminaries

When trying to interpret and analyze the semantics of the law, most existing research takes its roots in either deontic logic [32] or the Hohfeldian system of legal concepts [33]. Deontic logic distinguishes "what is permitted" (permission or right) from "what ought to be" (obligation) and their negations: what is "unpermitted" ("prohibition") and what "not ought to be" ("omissible" or non-obligatory), respectively.

The Hohfeldian system [33] distinguishes eight terms for legal rights: claim (claim right), privilege, power, immunity, duty, no-claim, liability, and disability. Each term in the Hohfeldian system is paired with one opposite and one correlative term. Two rights are opposites if the existence of one excludes that of the other. Hohfeldian opposites are similar to how permissions and obligations are negated in deontic logic. Two rights are correlatives if the right of a party entails that there is another party (a counter-party) who has the correlative right. For example, a driver has the (claim) right to know why their vehicle has been stopped by the police; this implies a duty for the police to explain the reason for stopping the vehicle.

2.3.2 Semantic Metadata in Legal Requirements

Deontic logic and the Hohfeldian system introduce a number of important legal concepts. Several strands of work leverage these concepts for the elicitation and specification of legal requirements, and the definition of compliance rules. Below, we outline these strands and the legal concepts underlying each. Examples for many of the legal concepts can be found in Fig. 3.1. However, we note that not all publications provide precise definitions for the concepts they use. Further, for certain concepts, the provided definitions vary in different publications. Consequently, while Fig. 1 is useful for illustrating existing work, the definitions used by others may not be fully aligned with ours. Our definitions for the concepts in Fig. 1 are based on the conceptual model that we propose in Section 3.2.

Early foundations. Two of the earliest research strands in RE on extracting information from legal texts are by Giorgini et al. [34] and Breaux et al. [10]. These

approaches target the elicitation of rights and permissions following the principles of deontic logic. Breaux et al. provide a proof-of-concept example of how structured information may be extracted from legal texts. Extending the generic Cerno information extraction framework [35], Kiyavitskaya et al. [36] develop automation for the approach of Breaux et al.'s. The automation addresses *rights*, *obligations*, *exceptions*, *constraints*, *cross-references*, *actors*, *policies*, *events*, *dates*, and *information*.

The above strands lay the groundwork for two different branches of research on legal requirements. The first branch is oriented around goal modeling, and the second around formal rules specified in either restricted natural language or logic.

Goal-based legal requirements. The initial work of Kiyavitskaya et al. with Cerno was enhanced by Zeni et al. in the **GaiusT tool** [37]. GaiusT pursues an explicit objective of identifying metadata in legal texts and using this metadata for building goal-based representations of legal requirements. GaiusT is centered around the concepts of: (1) *actors* who have *goals*, *responsibilities* and *capabilities*, (2) *prescribed behaviors* according to the deontic logic modalities of *rights*, *obligations* and their respective opposites, (3) *resources*, specialized into *assets* and *information*, (4) *actions* that describe what is taking place, and (5) *constraints*, either *exceptions* or *temporal conditions*, which affect the *actors*, *resources* or *prescribed behaviors*. GaiusT further addresses structural legal metadata which we are not concerned with here.

In tandem with GaiusT, the different versions of the **Nomos framework** [38–41] provide a complementary angle toward metadata extraction with a more pronounced alignment with goal models. Nomos models are built around five core concepts: *roles* (the holder or beneficiary of provisions), *norms* (either *duties* or *rights*), *situations* describing the past, actual or future state of the world, and *associations* describing how a provision affects a given situation. Zeni et al. propose **NomosT** [39] to automate the extraction of Nomos concepts using GaiusT. While still grounded in Nomos' original concepts, NomosT reuses several other concepts from GaiusT, including *actors*, *resources*, *conditions*, and *exceptions*.

The above work strands follow the principles of deontic logic. Another strand of work on goal-based analysis of legal requirements is LegalGRL [42, 43] which, in contrast to the above, follows the Hohfeldian system. The main legal concepts in LegalGRL are: *subjects*, *modalities* (based on Hohfeld's classifications of rights), *verbs*, *actions*, *cross-references*, *preconditions*, and *exceptions*. LegalGRL does not yet have automated support for metadata extraction.

Formal legal requirements. Following up on their earlier work [10] and motivated by deriving compliance requirements, Breaux et al. [44, 45] propose an **upper ontology**

for formalizing *frames* in legal provisions. This ontology has two tiers. The first tier describes statement-level (sentence-level) concepts. These concepts are: *permissions*, *obligations*, *refrainments*, *exclusions*, *facts*, and *definitions*. The second tier describes the concepts related to the constituent phrases in legal statements (phrase-level concepts). In this second tier, *actions* are used as containers for encapsulating the following concepts: *subjects*, *acts*, *objects*, *purposes*, *instruments* and *locations*. For actions that are *transactions*, one or more *targets* need to be specified. Breaux et al. further consider *modalities*, *conditions* and *exceptions* at the level of phrases.

Maxwell and Antón [46] propose a classification of semantic concepts for building formal representations of legal provisions. These representations are meant at guiding analysts throughout requirements elicitation. At the level of statements, the classification envisages the concepts of *rights*, *permissions obligations* and *definitions*. At a phrase level, the concepts of interest are the *actors* involved in a provision and the *preconditions* that apply to the provision.

Massey et al. [47, 48] develop an approach for mapping the terminology of a legal text onto that of a requirements specification. The goal here is to assess how well legal concerns are addressed within a requirements specification. Massey et al. reuse the concepts of *rights*, *obligations*, *refrainments* and *definitions* from Breaux et al.'s upper ontology, while adding *prioritizations*. At a phrase level, the approach uses *actors*, *data objects*, *actions* and *cross-references*.

2.3.3 Semantic Metadata in Legal Knowledge Representation

There is considerable research in the legal knowledge representation community on formalizing legal knowledge [49]. Several ontologies have been developed for different dimensions of legal concepts [50, 51]. Our goal here is not to give a thorough exposition of these ontologies, because our focus is on the metadata types (discussed in Section 2.3) for which clear use cases exist in the RE community.

An overall understanding of the major initiatives in the legal knowledge representation community is important for our purposes: First, these initiatives serve as a confirmatory measure to ensure that we define our metadata types at the right level of abstraction. Second, by considering these initiatives, we are able to create a mapping between the metadata types used in RE and those used in these initiatives; this is a helpful step toward bridging the two communities.

We consider two major initiatives, LKIF [52–54] and LegalRuleML [55, 56], which are arguably the largest attempts to date on the harmonization of legal concepts.

LKIF is a rule modeling language for a wide spectrum of legal texts ranging from legislation to court decisions. LKIF's core ontology includes over 200 classes. At a statement level, LKIF supports the following deontic concepts: *rights*, *permissions*, *obligations*, and *prohibitions*. At a phrase level, LKIF's most pertinent concepts are: *actors*, *objects*, *events*, *time*, *locations*, *trades*, *transactions*, and *delegations* (further specialized into *mandates* and *assignments*). LKIF further provides concepts for the *antecedents* and *consequents* of events.

LegalRuleML [55, 56] – a successor of LKIF – tailors the generic RuleML language [57] for the legal domain. LegalRuleML classifies statements into *facts* and *norms*. Norms are further specialized into *constitutive statements* (definitions), *prescriptive statements*, and *penalty statements*. The modality of a prescriptive statement is, at a phrase level, expressed using one of the following deontic concepts: *right*, *permission*, *obligation* or *prohibition*. Penalty statements have embedded into them the concepts of *violations* and *reparations*. LegalRuleML further introduces the following concepts directly at the level of phrases: *participants*, *events*, *time*, *locations*, *jurisdictions*, *artifacts*, and *compliance* (opposite of *violation*). The participants may be designated as *agents*, *bearers* or *third parties*, who may have *roles* and be part of an *authority*.

All the above-mentioned concepts from LKIF and LegalRuleML have correspondences in the RE literature on legal requirements, reviewed in Section 2.3. In Section 3.2, we reconcile all the RE-related legal concepts identified in an attempt to provide a unified model of legal metadata for RE.

Chapter 3

A Conceptual Model of Semantic Legal Metadata

Semantic legal metadata provides information that helps with understanding and interpreting the meaning of legal provisions. Such metadata is important for the systematic analysis of legal requirements. Our work in this chapter is motivated by the observation that the existing requirements engineering (RE) literature does not provide a harmonized view on the semantic metadata types that are useful for legal requirements analysis. Our objective is to take steps toward addressing this limitation. We review and reconcile the semantic legal metadata types proposed in RE. We propose a harmonized conceptual model for the semantic metadata types pertinent to legal requirements analysis.

3.1 Motivations and Contributions

Legal metadata provides explicit conceptual knowledge about the structure and content of legal texts. The requirements engineering (RE) community has long been interested in legal metadata as a way to systematize the process of identifying and deriving legal requirements and compliance rules [45, 48, 37]. There are several facets to legal metadata: *Administrative metadata* keeps track of the lifecycle of a legal text, e.g., the text's creation date, its authors, its effective date, and its history of amendments. *Provenance metadata* maintains information about the origins of a legal text, e.g., the parliamentary discussions preceding the ratification of a legislative text. *Usage metadata* links legal provisions to their applications in case law, jurisprudence, and doctrine. *Structural metadata* captures the hierarchical organization of a legal text (or legal corpus). Finally, *semantic metadata* captures fine-grained information about the

meaning and interpretation of legal provisions. This information includes, among other things, modalities (e.g., permissions and obligations), actors, conditions, exceptions, and violations. This fine grained information is useful for understanding the content of the provision per se.

Among the above, structural and semantic metadata have been studied the most in RE. Structural metadata is used mainly for establishing traceability to legal provisions, and performing such tasks as requirements change impact analysis [58, 59] and prioritization [48, 47]. Semantic metadata is a prerequisite for the systematic derivation of compliance requirements [45, 44, 38, 20], and transitioning from legal texts to formal specifications [46] or models [37–39].

In this chapter, we concern ourselves with *semantic legal metadata*. In Fig. 3.1, we exemplify such metadata over three illustrative legal statements. These statements come from the traffic laws for Luxembourg, and have been translated into English from their original language. Statement 1 concerns the management of public roads by the municipalities. Statement 2 concerns penalties for violating the inspection processes for vehicles. Statement 3 concerns the interactions between the magistrates in relation to ongoing prosecutions on traffic offenses. In these examples, we provide metadata annotations only for the phrases within the statements (*phrase-level metadata*). Some of these phrase-level annotations induce annotations at the level of statements (*statement-level metadata*). For example, the "may" modality in Statements 1 and 3 makes these statements permissions. The modal verb "shall" in Statement 2, combined with the presence of a sanction, make the statement a penalty statement. In Section 3.3, we will further explain the metadata types illustrated in Fig. 3.1.

The example statements in Fig. 3.1 entail legal requirements for various governmental IT systems, including road and critical infrastructure management systems, as well as case processing applications used by the police force and the courts.

The metadata annotations in Fig. 3.1 provide useful information to requirements analysts. Indeed, and as we argue more precisely in Section 2.3, the RE literature identifies several use cases for semantic legal metadata in the elicitation and elaboration of legal requirements. For instance, the annotations of Statement 1 help with finding the conditions under which a road restriction can be put in place. The annotations of Statement 2 may lead the analyst to define a compliance rule made up of an antecedent (here, absence of an agreement), an action (here, performing vehicle inspections) and a consequence (here, a range of sanctions). Finally, the annotations of Statement 3 provide cues about the stakeholders who may need to be interviewed during requirements

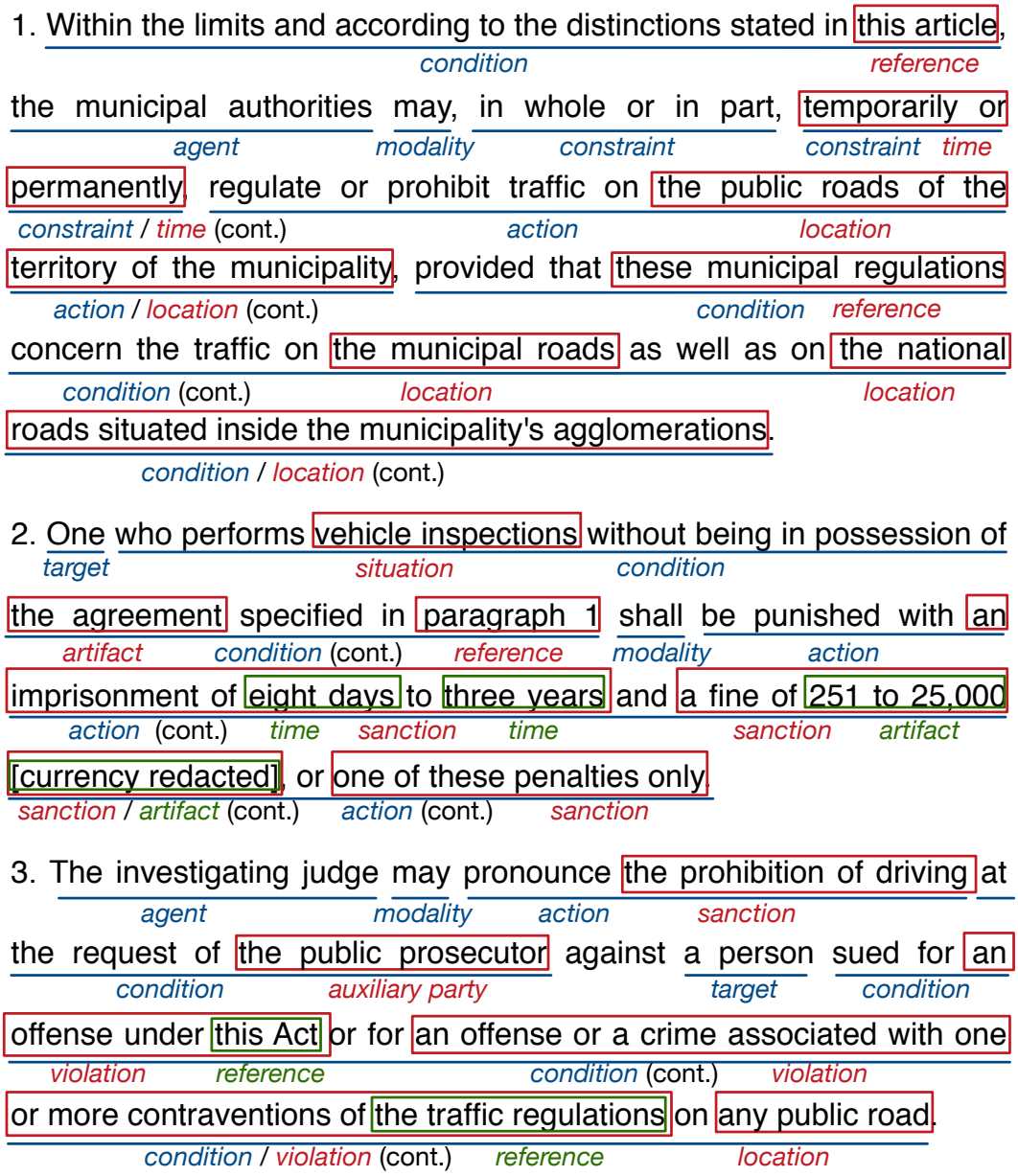

Fig. 3.1 Examples of Manually-annotated Legal Concepts

elicitation (agents and auxiliary parties), as well as the way these stakeholders should interact, potentially using computer systems.

Our work in this chapter is motivated by two observed limitations in the state-of-the-art on semantic legal metadata:

Lack of a harmonized view of semantic legal metadata for RE. While the RE community acknowledges the importance of semantic legal metadata, there is no consensus on the metadata types that are beneficial for legal requirements analysis. Different work strands propose different metadata types [44, 37, 46, 36, 40], but no strand completely covers the others.

Research Question (RQ). Throughout the chapter, we investigate the following research question which tackles the above mentioned limitation.

RQ: What are the semantic legal metadata types used in RE? RQ aims at developing a harmonized specification of the semantic metadata types used in legal RE. To this end, we review and reconcile several existing classifications. Our answer to RQ is the first contribution of the chapter: *a conceptual model of semantic metadata types pertinent to legal requirements analysis. The model defines six metadata types for legal statements, and 18 metadata types for the phrases thereof. A glossary alongside mappings to the literature are provided as an online annex [60].*

Overview and Structure. We refer the reader to Section 2.3 which reviews the background and related work. Section 3.3 describes our conceptual model for semantic legal metadata. Section 3.4 discusses threats to validity. Section 3.5 concludes the chapter.

3.2 Approach for the Harmonization

In this section, we present the mapping of the concepts elicited in Section 2.3. This mapping constitutes the basis for the elaboration of our conceptual model for legal concepts.

The main challenge in reconciling the above proposals is that they introduce distinct but overlapping concepts. We present our mapping of legal concepts in Table 3.1. For the elaboration of the mapping, we consider the concepts elicited in the related work (columns of the table), analyze their definition (when provided) and compare them to the definitions contained in other work, in an attempt to reconcile the concepts and the terminology.

In the table, concepts in bold and shaded green have a definition that is close to our own related concept. For instance, the concepts of *exception*, *right*, *obligation* and *definition* are shared and are similar across most of the examined work.

Concepts in italic and shaded orange are related to the concepts in our own taxonomy, but the alignment between the concepts is weaker than for the ones in bold and shaded green. This is largely due to variations in the granularity level adopted across the examined work. For instance, while one can find notions for actors in most of the work, the granularity at which they are described, i.e., their role, varies from one strand of work to another. It varies from an explicit list of roles in Breaux et al.'s upper ontology [45], to the simple notions of role in Nomos [40] and of actor in Cerno [35] and GaiusT [37], and to the fine-grained taxonomy in LKIF [52]. Another example are obligations and prohibitions, which in Nomos [40] are represented by the single concept of *norm (duty)*.

Empty cells correspond to concepts that are not described in a given work, or whose abstraction is too far from our interpretation. For instance, only few strands of work are concerned with *definition*, *violation*, *penalty* or *sanction*.

In the first column, there are concepts (highlighted in red) that correspond to concepts in the literature that we initially characterized but eventually decided not to retain. The rationale for these decisions is discussed during the elaboration of our conceptual model.

Our conceptual model for semantic legal metadata is presented in Fig. 3.2. It leverages the mapping that we have performed and described in the previous Section. The dashed boundaries in the figure distinguish statement-level and phrase-level metadata types. Our conceptual model brings together existing proposals from the literature [45, 46, 40, 48, 42, 37]. The model derives the vast majority of its concepts (83.3% or 20/24) from Breaux et al.'s upper ontology [45] and GaiusT [37].

Our model includes six concrete concepts at statement level. Aside from *penalty*, all statement-level concepts are from Breaux et al.'s upper ontology [45]. Penalty comes from LKIF [52]: we found this concept to be a necessary designation for statements containing sanctions. The model envisages 18 concrete concepts for phrases, most of which are illustrated in the statements of Fig. 3.1. We propose a definition for each of these concepts in Table 3.2 and we further discuss them below, starting with statement-level concepts.

Fact is something that is known or proved to be true and comes from Breaux et al.'s upper ontology [45]. This is in line with the classifications from LegalRuleML [55] and LKIF [52].

Table 3.1 Mapping of the Various Legal Concepts Elicited in Selected Work from the Literature

Our Approach	LegalRuleML [30, 31]	LKIF [27]	Cerno [20]	Nomos [8, 13, 21]	GaiusT [3]	NomosT[11]	Breaux's Upper Ontology [1, 7]	Massey's Classification [2]	Maxwell's Classification [10]	Legal GRL [22, 23]
Agent	**Agent**	**Agent**	*Actor*	*Role (Holder)*	**Agent**	*Role (Holder) / Actor*	**Subject**	*Actor*	*Actor*	*Subject (Actor)*
Artifact		**Artifact**	*Information*		*Resource / Information / Asset*			*Data Object*		
		Natural Object			*Resource / Information / Asset*			*Data Object*		
		Document / evidence	*Information*		*Resource / Information / Asset*					
Authority	**Authority**	**Public Body / Legislative Body**								
Auxiliary Party	**Auxiliary Party**	*Organisation/ Co-operative ...*			*Actor*	*Actor*	*Target*	*Actor*	*Actor*	
	Auxiliary Party	*Person*	*Actor*	*Role*	*Actor*	*Role*	*Target*	*Actor*	*Actor*	
Compliance	**Compliance**			**Satisfies Norm/situation**		**Satisfies Norm/situation**				
Constraint / Condition	*Context*	*Cause*	**Constraint**		**Constraint**	**Condition**	**Condition**		**Precondition**	**Precondition**
Definition	**Constitutive Statement**						**Definition**	**Definition**	**Definition**	
Delegation		**Assignment / Delegation / Mandate / Trade/**					**Transaction**			
Exception		**Exception**	**Exception**	**Block norm**	**Exception**	**Block norm / Derogation /**	**Exception**	**Exception**		**Exception**
Fact	**Factual Statement**	**Fact**					**Fact**			
Location / *Jurisdiction*	*Jurisdiction*	*Place*					**Location**			
Modality							**Modality**			**Modality**
Obligation	**Obligation**	**Obligation**	**Obligation**	**Norm (Duty)**	**Obligation**	**Norm (Duty)**	**Obligation**	**Obligation**	**Obligation**	*Duty/Claim -- Immunity/Disability**
Exclusion		*Immunity*	*Anti-obligation*		*Anti Obligation*		*Exclusion*			*Immunity/Disability*
Penalty	**Penalty Statement**									
Permission	**Permission**	**Permission / allowed**					**Permission**		**Permission**	*Privilege/NoClaim -- Power/Liability**
Prescriptive provision	**Prescriptive Statement**									*Hohfeldian Statement*
Prohibition	**Prohibition**	**Prohibition**	**Anti-right**	*Norm (Duty)*	**Anti Right**	*Norm (Duty)*	**Refrainment**	**Refrainment**		
Reason		**Reason**			*Goal*		**Purpose**			
Reference	**Legal Source / Reference**	**Legal Source**	**Policy / Cross-Reference**					**Cross-Reference**	**Cross-Reference**	**Cross-Reference**
Result				**Activate norm / Consequent**		**Activate norm / Consequent**				
Sanction	**Reparation**									
Right	**Right**	**Right**	**Right**	**Norm (Right)**	**Right**	**Norm (Right)**		**Right**	**Right**	**Power/Liability**
Role	**Role**	**Role**		**Role**	*Actor*	*Role / Actor*		*Actor*	*Actor*	
Action / Situation		**Action / Process**	*Event*	**Situation / Antecedent**	**Action**	**Situation / Antecedent**	**Act / Instrument**	**Action**		**Clause (verb)/ Clause (action)**
Target	**Bearer**			*Role (beneficiary)*	*Actor*	**Role (Beneficiary) /**	*Object / Target*	*Actor*	*Actor*	*Subject (Exception Actor)*
Time	**Time / Temporal**	**Time**	*Date*		*Temporal Condition*					
Violation	**Violation**	*Disallowed*		**Break Norm/situation**		**Break Norm/situation**				
--	Override									
--		Application / Efficacy					Quality	Prioritization		

* according to the authors

Table 3.2 Glossary for Our Legal Concepts

Concept	**Definition**
Action	the process of doing something
Actor	an entity that has the capability to act
Agent	an entity that is the main actor performing the action
Artifact	a human-made object involved in an action
Auxiliary Party	an actor that in some way participates in an action but is neither the agent nor the target
Condition	a constraint stating the properties that must be met
Definition	a legal provision defining the meaning of concepts
Constraint	a restriction placed on the applicability of a legal provision
Exception	a constraint indicating that a legal provision takes precedence over another legal provision
Fact	something that is known or proved to be true
Location	a place where an action is performed
Modality	a verb indicating the modality of the action (e.g may, must, shall)
Obligation	a provision imposing mandatory action to be performed by an agent
Penalty	a provision indicating the result of breaking an obligation or prohibition
Permission	a provision indicating the possibility to perform an action without an obligation or a prohibition
Prohibition	a provision forbidding an action to happen or take place
Reason	the rationale for an action
Reference	a mention of other legal provision(s) or legal text(s) affecting the current provision
Result	the outcome of an action
Sanction	a punishment imposed in a penalty
Statement	a (well-formed) sentence within a legal text
Situation	a description of something that has happened or can happen
Target	an entity that is directly affected by an action
Time	the moment or duration associated with the occurence of an action
Violation	a condition that indicates explicit criteria for non-compliance (denial of a provision)

Definition is a legal provision that defines the meaning of a concept. We adopted this statement-level concept from Breaux et al.'s upper ontology [45] as we found it in other classifications [48, 46] as well. This is also aligned with the concept of constitutive statement that is present in the Knowledge Representation community [61].

Obligation, *Permission* and *Prohibition* are three modal statement-level concepts that can be found across the legal literature at different granularity levels, as we previously described in Section 2.3. Traditionally, these statements are related to the use of modal verbs. For example, in *permission* statements such as Statements 1 and 3 in Fig. 3.1, the "may" modality induces the statement-level concept *permission*. However, in French, modal verbs are not always used for stating obligations or prohibitions, and other cues are needed. For example, the statement "the police officer notifies the driver ..." describes an implicit obligation for the police officer to perform an action.

Finally, we adopted the statement-level concept *penalty* from the Knowledge Representation community. A penalty statement is a provision that imposes sanctions in case of non-compliance. Statement 2 in Fig. 3.1 provides an example of a penalty statement.

We now continue the presentation of our conceptual model with phrase-level concepts.

Agent is an actor performing an action, whereas *target* is an actor affected by the action stated in the provision. A third form of actor is *auxiliary party*, which is neither an agent nor a target, but rather an intermediary. Examples of agents and targets are provided in Statements 1 and 2, respectively. An example of co-occurrence of all three actor types is provided in Statement 3.

The concept of *artifact* captures human-made objects (physical or virtual). An example is "the agreement" in Statement 2.

The concept of *situation* describes a state of affairs, similarly to Nomos [40]. A situation may be a *result*, and a result may in turn be classified as a *sanction*. An example is "the prohibition of driving" in Statement 3.

The description of "what is happening" is considered as a norm in Nomos [40], an action in GaiusT [37], an act in Breaux et al.'s upper ontology [45], and a clause in LegalGRL [42]. In our model, we follow GaiusT's terminology thus adopting the term *action*. As illustrated by our statements in Fig. 3.1, an *action* can be associated to a *modality* (often expressed via a modal verb) and to *constraints*. Constraints may be further classified as *exceptions* or *conditions*. Conditions may in turn be classified as *violations*, when they describe the circumstances under which the underlying statements are breached (violated). Statement 2 provides an example of violation. Violations,

alongside sanctions discussed earlier, provide information that is necessary for inferring the consequences of non-compliance.

We capture the purpose of a statement using the concept of *reason* (not illustrated in Fig. 3.1). This concept corresponds to *purpose* in Breaux et al.'s upper ontology, to *goal* in GaiusT, and to *reason* in LegalRuleML. Finally, a statement may contain information in the form of *references*, *times* and *locations*. These concepts are all illustrated in the statements of Fig. 3.1.

As a final remark, we note that not all the concepts discussed in Section 2.3 have been retained in our model. The decision not to retain was made when we deemed that a concept could be expressed using other concepts, or when the concept could not be directly captured as metadata. For example, *compliance* results from the satisfaction of one or more conditions; *delegation* is a particular type of action involving an auxiliary party; *exclusion* is an implicit type, difficult to infer without additional reasoning.

3.3 Conceptual Model

Our conceptual model for semantic legal metadata is presented in Fig. 3.2. The dashed boundaries in the figure distinguish statement-level and phrase-level metadata types. Our conceptual model brings together existing proposals by Breaux et al. [45], Maxwell and Antón [46], Siena et al. [40], Massey et al. [48], Ghanavati et al. [42] and Zeni et al [37]. The model derives the majority – 83.3% (20/24), to be precise – of its concepts from the work of Breaux et al.'s [45] and Zeni et al.'s [37]. Due to space, we do not present the full mapping we have developed between the above proposals. This mapping is available in an online annex [60]. The annex further provides a glossary for our conceptual model.

The main challenge in reconciling the above proposals is that they introduce distinct but overlapping concepts. When dealing with overlapping concepts in the RE literature, we favored concepts that aligned better with LKIF [52] and LegalRuleML [55], outlined in Section 2.3.3. This decision was driven by the desire to define our concepts at a level of abstraction that allows interoperability with initiatives in the legal knowledge representation community.

Our model has six concrete concepts at the level of statements. Aside from *penalty*, all statement-level concepts are from Breaux et al. [45]. Penalty comes from LKIF; we found this concept to be a necessary designation for statements containing sanctions. The model envisages 18 concrete concepts for phrases. Most have been illustrated in the statements of Fig. 3.1. *Agent* is an actor performing an action, whereas *target* is

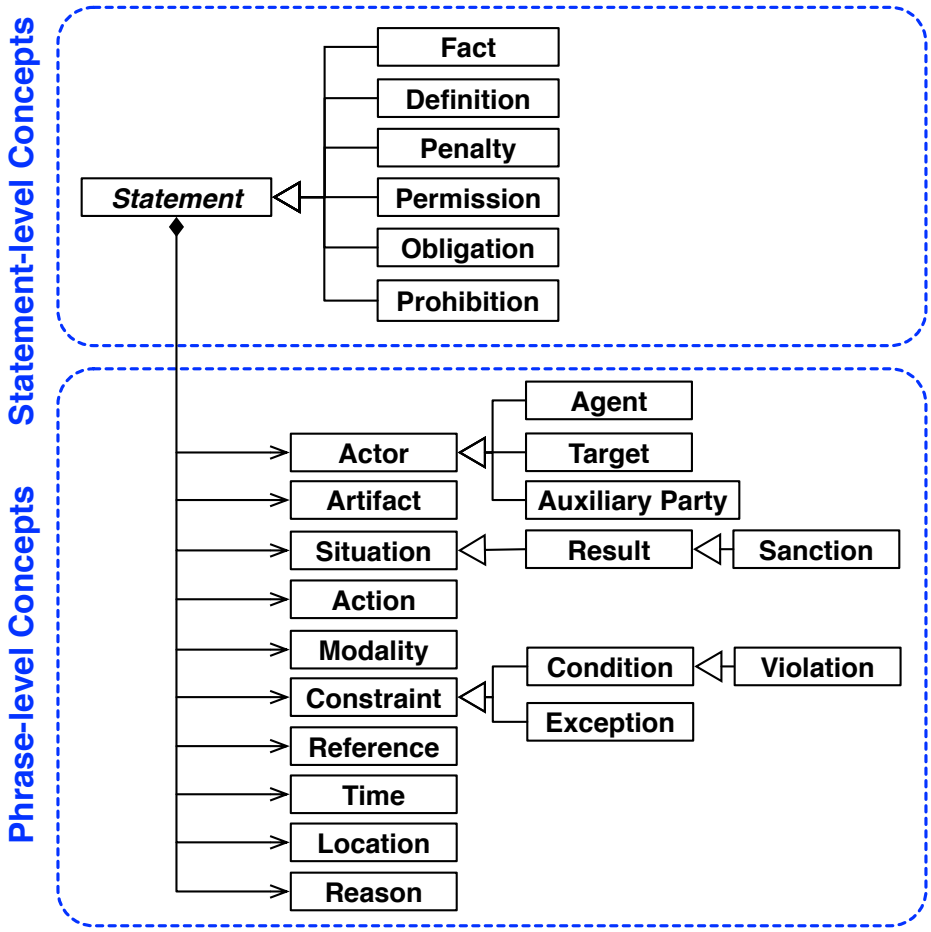

Fig. 3.2 Conceptual Model for Semantic Legal Metadata Relevant to RE

an actor affected by the enforcement of a provision. A third form of actor is *auxiliary party*, which is neither an agent nor a target, but rather an intermediary. Examples of agents and targets are given in Statements 1 and 2, respectively. An example of all actor types together is given in Statement 3.

The concept of *artifact* captures human-made objects (physical or virtual). An example artifact is "the agreement" in Statement 2. The concept of *situation* describes a state of affairs, similarly to Nomos [40]. A situation may be a *result*; a result may be further classifiable as a *sanction*. An example situation is "the prohibition of driving" in Statement 3. This situation also happens to be a sanction (and thus a result too).

The description of "what is happening" is considered as a *norm* in Nomos [40], an *action* in GaiusT [37], an *act* in Breaux et al.'s upper ontology [45] and a *clause* in LegalGRL [42]. In our model, we follow GaiusT's terminology. As illustrated by our statements in Fig. 3.1, an action can be linked to a *modality* (often expressed via a modal verb), as well as to *constraints*. Constraints may be further classifiable as *exceptions* or *conditions*. Conditions may be further classifiable as *violations*; this is when a condition describes the circumstances under which the underlying statement is denied (violated). Statement 2 provides an example of a violation. Violations, alongside sanctions discussed earlier, provide information that is necessary for inferring the consequences of non-compliance.

We capture the purpose for a statement using the concept of *reason* (not illustrated in Fig. 3.1). This concept corresponds to *purpose* in Breaux et al.'s upper ontology and to *goal* in GaiusT. The term "reason" comes from LegalRuleML. Finally, a statement may contain information represented in the form of *references*, *time* and *locations*. These concepts are all illustrated in the statements of Fig. 3.1.

As a final remark, we note that not all the concepts discussed in Section 2.3 have been retained in our model. A decision not to retain was made when we deemed a concept expressible using other concepts, or when the concept did not readily lead to metadata. For example, *compliance* results from the satisfaction of one or more conditions. *Delegation* is a particular type of action involving an auxiliary party. *Exclusion* is an implicit type and difficult to infer without additional reasoning.

3.4 Threats and Limitations

The most pertinent threats to the validity of our work concern internal validity, as we discuss below.

Internal validity. A potential threat to internal validity is that the researchers interpreted the existing legal metadata types. To mitigate the threat posed by subjective interpretation, we tabulated all the concepts identified in the literature and established a mapping between them. By doing so, we helped ensure that no concepts were overlooked, and that the correspondences we defined between the different metadata types were rooted in the existing definitions. While we cannot rule out subjectivity, we provide our interpretation in a precise and explicit form [60]. This is thus open to scrutiny.

As a final remark, the selection of the appropriate semantic metadata heavily depends on the final use case. Specific use cases such as automated elicitation of requirements or answering legal-related questions utilize different elements of the conceptual model as detailed in Chapters 5 and 6. In these chapters, we investigate the usability of our harmonized conceptual model of semantic legal metadata in practice for these specific use cases.

3.5 Conclusion

Metadata about the semantics of legal statements is an important enabler for legal requirements analysis. In this chapter, we described an attempt at reconciling the different types of semantic legal metadata proposed in the RE literature.

Chapter 4

Automated Extraction of Semantic Legal Metadata

Semantic legal metadata provides information that helps with understanding and interpreting legal provisions. Such metadata is therefore important for the systematic analysis of legal requirements as we discussed in the previous chapter. However, manually enhancing a large legal corpus with semantic metadata is prohibitively expensive. Our work is motivated by the observation that automated support for the extraction of semantic legal metadata is scarce, and it does not exploit the full potential of artificial intelligence technologies, notably natural language processing (NLP) and machine learning (ML). Our objective is to take steps toward overcoming this limitation. To do so, we devise an automated extraction approach for the identified metadata types in the previous chapter using NLP and ML. We evaluate our approach through two case studies over the Luxembourgish legislation. Our results indicate a high accuracy in the generation of metadata annotations. In particular, in the two case studies, we were able to obtain precision scores of 97.2% and 82.4%, and recall scores of 94.9% and 92.4%.

4.1 Motivations and Contributions

Legal metadata provides explicit conceptual knowledge about the content of legal texts. The requirements engineering (RE) community has long been interested in legal metadata as a way to systematize the process of identifying and elaborating legal compliance requirements [45, 48, 37].

Semantic metadata is a prerequisite for the systematic derivation of compliance requirements [45, 44, 38, 20] and for transitioning from legal texts to formal specifications [46] or models [37–39].

In this chapter, we concern ourselves with the extraction of *semantic legal metadata*. In Fig. 4.1, we exemplify such metadata over three illustrative legal statements. These statements come from the traffic laws for Luxembourg, and have been translated into English from their original language, French. Statement 1 concerns the management of public roads by the municipalities. Statement 2 concerns penalties for violating the inspection processes for vehicles. Statement 3 concerns the interactions between the magistrates in relation to ongoing prosecutions on traffic offenses. In these examples, we provide metadata annotations only for the phrases within the statements (*phrase-level metadata*). Some of these phrase-level annotations, however, can also induce annotations at the level of statements (*statement-level metadata*): for example, the "may" modality in Statements 1 and 3 makes these statements permissions, and the modal verb "shall" in Statement 2, combined with the presence of a sanction, makes the statement a penalty statement. The metadata types illustrated in Fig. 4.1 will be further explained in Section 3.2.

The example statements in Fig. 4.1 entail legal requirements for various governmental IT systems, including road and critical infrastructure management systems, as well as case processing applications used by the law enforcement agencies and the courts. In this regard, the metadata annotations in Fig. 4.1 provide useful information to requirements analysts: as we argue more precisely in Section 2.3.2, the RE literature identifies several use cases for semantic legal metadata in the elicitation and elaboration of legal requirements. For instance, the annotations of Statement 1 may help with finding the conditions under which a road restriction can be put in place. The annotations of Statement 2 may lead the analyst to define a compliance rule composed of an antecedent (i.e., absence of an agreement), an action (i.e., performing vehicle inspections) and a consequence (i.e., a range of sanctions). Finally, the annotations of Statement 3 provide cues about the stakeholders who may need to be interviewed during requirements elicitation (agents and auxiliary parties), as well as the way in which these stakeholders should interact, potentially using IT systems.

Our work in this chapter is motivated by an observed limitation in the state-of-the-art on semantic legal metadata, discussed below.

Lack of coverage of recent advances in NLP and ML. If done manually, enhancing a large corpus of legal texts with semantic metadata is extremely laborious. Recently, increasing effort has been put into automating this task using natural language

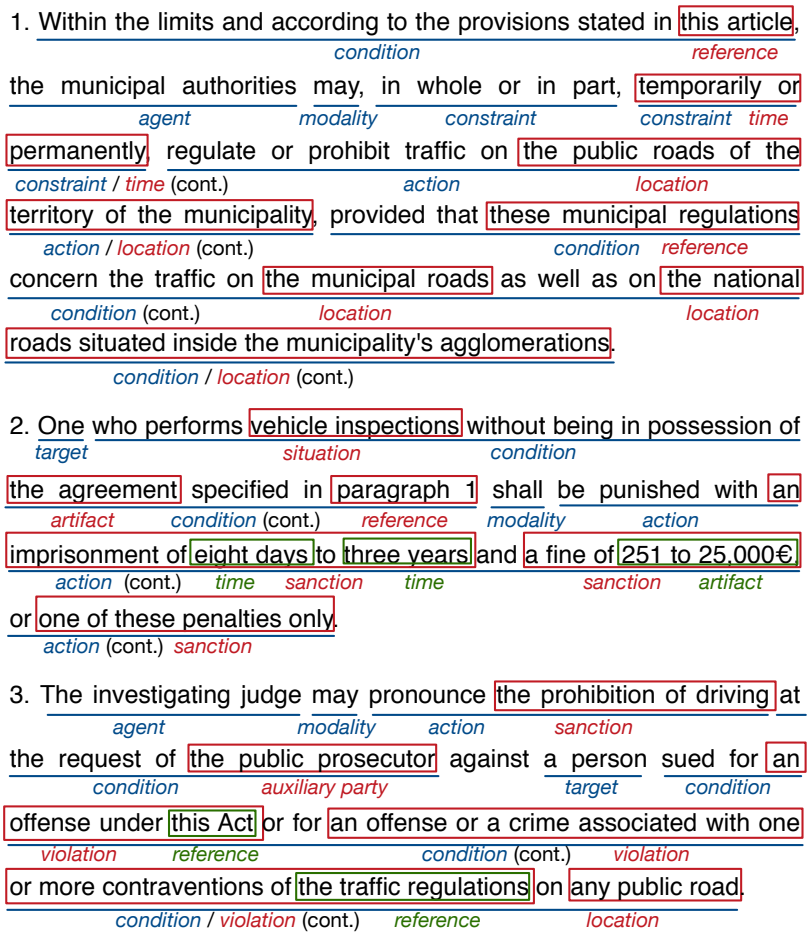

Fig. 4.1 Examples of Semantic Legal Metadata Annotations

processing (NLP). Notable initiatives aimed at providing automation for metadata extraction are GaiusT [37] and NomosT [39]. These initiatives do not handle the broader set of metadata types proposed in the RE literature, e.g., locations [44], objects [48], and situations [40]. Besides, they rely primarily on simple NLP techniques, e.g., tokenization, named-entity recognition, and part-of-speech (POS) tagging. Although these techniques have the advantage of being less prone to mistakes, they cannot provide detailed insights into the complex semantics of legal provisions.

With recent developments in NLP, the robustness of advanced NLP techniques, notably constituency and dependency parsing, has considerably improved [29]. This raises the prospect that these more advanced techniques may now be accurate enough for a deep automated analysis of legal texts. Dependency parsing is important for correctly identifying constituents whose roles are influenced by linguistic dependencies. For instance, in Statement 3 of Fig. 4.1, the roles of the "(sued) person", the "investigating judge" and the "public prosecutor" can be derived from such dependencies. Constituency parsing is instead important for delineating phrases out of simpler nouns or chunks in a statement. For instance, in Statement 1 of Fig. 4.1, annotating "the national roads situated inside the municipality's agglomerations" as one segment requires the ability to recognize this segment as a compound noun phrase. Without a parse tree, one cannot readily mark this segment in its entirety.

In addition, machine learning (ML) provides a potentially useful mechanism for distinguishing metadata types that NLP-based rules cannot handle with sufficient accuracy. For instance, in Statement 3 of Fig. 4.1, the "investigating judge", the "public prosecutor" and the "sued person" hold three closely related stakeholder roles, differentiated by whether they are acting, are a third party or are the target of the action described in the statement. Articulating explicit rules for distinguishing such metadata types proved very difficult. This prompted us to investigate whether ML can be employed for telling apart such metadata types.

In this chapter, we take a step toward addressing this limitation outlined above by developing a framework for automated semantic legal metadata extraction. In the previous chapter, we started by reviewing and reconciling several existing metadata classifications in order to devise a conceptual model of semantic metadata types pertinent to legal requirements analysis. This model defines six metadata types for legal statements and 18 metadata types for the phrases contained therein. In this chapter, we perform a qualitative study over 200 legal statements from the traffic laws of Luxembourg in order to define rules that can automatically detect the metadata contained in legal statements. This qualitative study results in a set of NLP-based

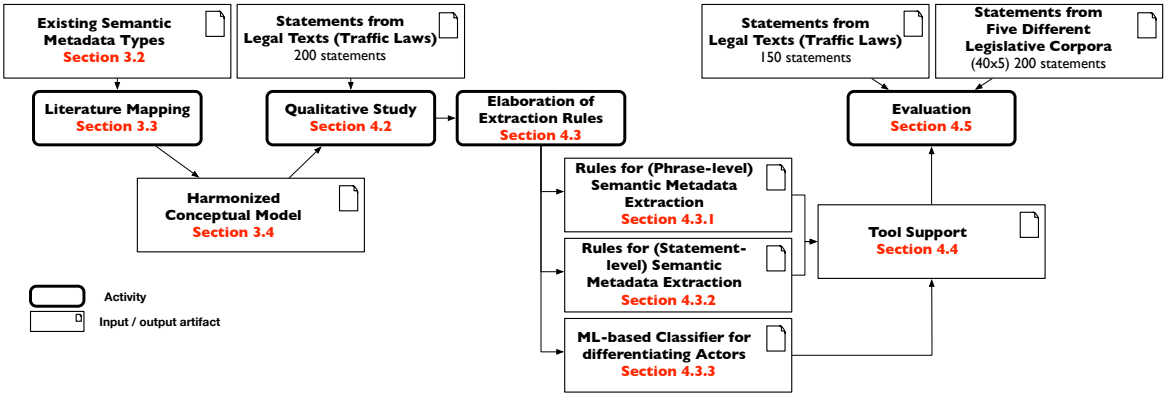

Fig. 4.2 Approach Overview for the Automated Extraction of Semantic Legal Metadata

rules for automated extraction of semantic legal metadata, covering the majority of the phrase-level metadata types and the statement-level metadata types of the conceptual model. These rules are complemented by using ML classification techniques in order to further distinguish stakeholders' roles in the statements. In our evaluation, we analyze 350 different statements from six different legislative domains in order to assess the accuracy of our metadata extraction rules.

Overview and Structure. Fig. 4.2 summarizes our approach.

Section 4.2 introduces a qualitative study aimed at validating the metadata types of the conceptual model by observing their occurrence in the legal statements. An additional objective of this qualitative study is to determine cues that could help in automating the extraction of the metadata types. Section 4.3 addresses the elaboration of extraction rules for the legal concepts in our conceptual model using a combination of NLP and ML. Section 4.4 describes the implementation of the extraction rules into a tool for semantic legal metadata extraction. Section 4.5 empirically assesses, through two case studies, the accuracy of our extraction rules. Section 4.6 discusses threats to validity, and Section 4.7 concludes the chapter.

4.2 Qualitative Analysis of legal Concepts

In this section, we report on the qualitative study aimed at defining extraction rules for semantic legal metadata.

Study context and data selection. We conducted our study in collaboration with Luxembourg's Central Legislative Service (*Service Central de Législation*, hereafter SCL). SCL's main mandate is the publication and dissemination of national legal texts. SCL already employs a range of semantic web technologies for legal text processing,

and has considerable prior experience with legal metadata. In recent years, SCL has been investigating the use of legal metadata for two main purposes:

1. Assisting IT engineers in identifying legal provisions that are likely to imply software requirements (Section 4.1 describes several use cases of semantic legal metadata for requirements analysts).

2. Providing an online service that enables lay individuals and professionals alike to interactively query the law, e.g., ask questions such as: "What would be the consequences of driving too fast on a road with a speed limit of 30 km/h?".

Our work is motivated by the former use case.

Our study focuses on the traffic laws for Luxembourg. They consist of 74 separate legal texts, including state-level legislation, regulations, orders and jurisprudence. Collectively, the texts are 1075 pages long and contain ≈12000 statements. The oldest (and main) text is from 1955 and the most recent from 2016.

The choice of traffic laws was motivated by two factors. First, due to these laws being intuitive and widely known, SCL found them convenient for showing the benefits of legal metadata to decision makers in Luxembourg. Second, the provisions in traffic laws are interesting from an RE perspective, due to their broad implications for the IT systems used by the police force, the courts, and the public infrastructure management departments.

Our study is based on 200 randomly selected statements from the traffic laws. As it is the case with most legal texts, the source texts in our study contain statements with enumerations and lists embedded in them. To treat these statements appropriately, we took the common legal text preprocessing measure of merging the beginning of a statement with its individual list items to form complete, independent sentences [62].

Analysis procedure. Our analysis procedure follows *protocol coding* [63], a method for collecting qualitative data according to a pre-established theory, i.e., a set of codes. In our study, the codes are the phrase-level concepts of the model of Fig. 5.2. The first researcher, who is a native French speaker and expert in NLP, analyzed the 200 selected statements from the traffic laws, and annotated the phrases of these statements. Throughout the process, difficult or ambiguous situations were discussed between the researchers (including a legal expert) and decisions were made based on consensus.

To assess the overall reliability of the coding, the second researcher – also a native French speaker, with background in NLP and regulatory compliance – independently annotated 10% of the selected statements, prior to any discussion among the researchers. Inter-annotator agreement was then computed using Cohen's κ [64]. Agreement was

reported when both annotators assigned the same metadata type to the same span of text. Other situations counted as disagreements. We obtained $\kappa = 0.824$, indicating "almost perfect agreement" [65].

Coding results. The coding process did not prompt the use of any concepts beyond what was already present in the conceptual model of Fig. 5.2. In other words, we found the concepts of the model to be adequately expressive.

Table 4.1 presents overall statistics about the studied statements by indicating the occurrences of each type of statement-level and phrase-level concept. In the majority of cases, we were able to assign a unique annotation to a given phrase. However, we noted that in some cases different interpretations of the same phrase would result in different annotations. The last column of the table provides information about such phrases. For instance, we annotated 73 phrases with the unique concept of *artifact*; in addition, we annotated seven phrases as both *artifact* and *sanction*, five phrases as both *artifact* and *situation*, and so on. We note that phrases are hierarchical and nested: consequently, nested annotations are prevalent, as illustrated by the statements in Fig. 4.1. What we show in the last column of Table 4.1 excludes nesting, i.e., it covers only phrases where more than one annotation is attached to exactly the same span. An example phrase is "temporarily or permanently" found in Statement 1 of Fig. 4.1: here two annotations, namely *constraint* and *time*, have been attached to the same span.

In total, we identified 1339 phrases in the 200 selected statements. Of these phrases, 1299 ($\approx$ 97%) have a single annotation, and the remaining 40 ($\approx$ 3%) have two annotations (no case was observed, where more than two annotations were possible).

With regard to the coverage of statement-level concepts, we observed at least nine occurrences of each concept, except for *facts*, for which we have none. In the Luxembourguish system, facts mostly concern generic assertions of little value to RE, such as the details of the publication in the official gazette or the contents of the preamble of the legislative act. However, case law is likely to contain more instances of fact, providing a deeper understanding of the law and of its interpretation, and the concept is thus important for RE, as described by Breaux et al. [45].

With regard to the coverage of the phrase-level concepts, we have at least 20 occurrences for each concept, with two exceptions: *constraint* has only five occurrences, and *result* has none. Despite our study not having identified any occurrences of *result*, the concept is still to be considered as important. Feedback from legal experts indicated in fact that there is a gap between *situation* and *sanction*. Consider for instance the following example statement (from outside our qualitative study): "If the defect is

Table 4.1 Metadata Annotations Resulting from Qualitative Study

Concept	Unique Classification	Multiple Classifications
Definition	9	
Fact	0	
Obligation	120	
Penalty	20	
Permission	36	
Prohibition	15	
Subtotal	**200**	
Action	187	
Agent	42	
Artifact	73	+7 sanctions, +5 situations, +3 times, +1 violation
Auxiliary Party	34	
Condition	230	+18 times, +1 violation
Constraint	5	+1 time
Exception	22	
Location	52	
Modality	68	
Reason	21	
Reference	111	
Result	0	
Sanction	91	+7 artifacts
Situation	162	+5 artifacts, +2 times, +2 violations
Target	73	
Time	90	+3 artifacts, +18 conditions, +1 constraint, +2 situations
Violation	38	+1 artifact, +1 condition, +2 situations
Subtotal	**1299**	**40**

fixed, the car is not subject to a new inspection." Here, "the defect is fixed" is a regular *situation* appearing as part of a *condition*. What follows, i.e, "the car is not subject to a new vehicle inspection" is the consequence of the first situation; however, this consequence is not a sanction. *Result* is thus a general notion for consequences that are not sanctions.

As for constraints that are unclassifiable as any of the specializations of *constraint* in the model of Fig. 3.2, consider the following statement: "Drivers of transport units [...] must observe, *with respect to the vehicles ahead of them*, a distance of at least 50 meters [...]." The italicized segment in this statement restricts the interpretation of distance. This constraint, however, qualifies neither as a *condition* nor as an *exception*.

We next describe the extraction rules that we derived from our qualitative study. We exclude *result* and *constraint* from the concepts, since the study did not yield a sufficient number of occurrences for these two concepts.

4.3 Approach for the Extraction of Semantic Legal Metadata

In this section, we present the extraction rules that we have derived based on the outcomes of the qualitative study described in the previous section. We start by presenting the rules for phrase-level metadata in Section 4.3.1, followed by the rules for statement-level metadata in Section 4.3.2, noting that the rules for statement-level metadata make use of phrase-level metadata. Finally, we present an extension of the NLP rules for classifying the specializations of actor, namely *agent*, *target* and *auxiliary party*. With regard to these, we observed that distinguishing them is highly context-dependent, and we were not able to derive rules that were simple enough and yet accurate. We therefore devised an alternative strategy for this particular classification using ML. We describe this strategy in Section 4.3.3.

4.3.1 Phrase-level Metadata Extraction Rules.

Table 4.2 presents the extraction rules that we derived by analyzing the 1339 manual annotations in our study. The rules were iteratively refined to maximize accuracy over these annotations. Our rules cover 12 out of the 18 phrase-level concepts in the model of Fig. 5.2. The concepts that are not covered are: *result*, *constraint* (both due to the scarce observations, as noted above), the three specializations of *actor*, and *(cross-)reference*.

Table 4.2 NLP-based Rules for Extracting Semantic Legal Metadata

Concept	Rule(s)
Action	• VP with modality, condition, exception and reason annotations removed
Actor	• subject dependency and NP < (actor marker) • object dependency and passive voice and PP < P $ (NP < (actor marker)) • object dependency and active voice and NP < (actor marker)
Artifact	• NP < (artifact marker) • NP !<< (violation marker) | !<< (time marker) | !<< (situation marker) | !<< (sanction marker) | !<< (reference marker) | !<< (location marker) | !<< (actor marker)
Condition	• Srel << (condition marker) • Ssub < (condition marker) • PP << (condition marker) • NP < (VPinf !<< (exception marker) & !<< (reason marker)) • NP < (VPart !<< (exception marker) & !<< (reason marker))
Exception	• Srel << (exception marker) • Ssub << (exception marker) • NP < (VPart << (exception marker)) • PP << (exception marker) • NP << (P < (exception marker) $ VPinf)
Location	• NP < (location marker)
Modality	• VN < (modality marker)
Reason	• Srel << (reason marker) • Ssub << (reason marker) • PP << (reason marker) • NP < (VPart << (reason marker)) • NP << (P < (reason marker) $ VPinf)
Sanction	• NP < (sanction marker)
Situation	• NP < (situation marker)
Time	• NP < (time marker) • PP < (P < (time marker)) $ NP
Violation	• NP < (violation marker)

NP: noun phrase, **PP**: prepositional phrase, **Srel**: relative clause, **Ssub**: subordinate clause, **VN**: nominal verb, **VP**: verb phrase, **VPinf**: infinitive clause, **Vpart**: VP starting with a gerund

With regard to *reference*, we made a conscious choice not to cover it in our extraction rules. Legal cross-references are well covered in the RE literature, with detailed semantic classifications already available [66, 67], and so is the automated extraction of cross-reference metadata [59, 67].

In Table 4.2, the element highlighted in blue in each rule is the target of annotation for that rule. All rules use constituency parsing, except the rules for *actor*, which use both constituency and dependency parsing. Aside from the rules for *action* and *actor*, all the rules are expressed entirely in Tregex [30], a widely used pattern matching language for (constituency) parse trees. The rule for *action* annotates every verb phrase (VP), excluding from the span of the annotation any embedded segments of type *modality*, *condition*, *exception*, and *reason*. Note that, to work properly, the rule for *action* has to be run after those for the four aforementioned concepts.

We do not provide a thorough description of Tregex which is already well-documented [30]. Below, we illustrate some of our rules to facilitate understanding and to discuss some important technicalities of the rules in general.

Consider Statement 1 in Fig. 4.1. A simplified parse tree for an excerpt of this statement is shown in Fig. 4.3. The *condition* annotation in this statement is extracted by the following Tregex rule: PP << (condition marker). This rule matches any prepositional phrase (PP) that contains a condition marker (in our example, the term "limit"). Initial sets of markers for all the concepts in Table 4.2, including conditions, were derived from our qualitative study on the 200 annotated statements from traffic laws. With these initial sets in hand, we followed different strategies for different concepts in order to make their respective sets of markers as complete as possible. We present these strategies next. Table 4.3 illustrates the markers for different concepts. We note that the terms in Table 4.3 are translations of the original markers in French. We also note that, for simplicity, the table provides a single set of markers per concept. In practice, different rules for extracting the same concept use different subsets of markers. For instance, "who" and "whose" are treated as concept markers in the first *condition* rule in Table 4.2 (Srel << (condition marker)), but not in the other four rules.

We observed that *actor* and *situation* have broad scopes, thus leading to large sets of potential markers. To identify the markers for these concepts in a way that would generalize beyond our study context, we systematically enumerated the possibilities found in a dictionary. More precisely, we analysed all the entries in Wiktionary [68]. Any entry classified as a noun and with a definition containing "act", "action", or "process" (or variations thereof) counts as a marker for *situation*. For instance, consider the term "inspection", defined by Wiktionary as "*The act of* examining something,

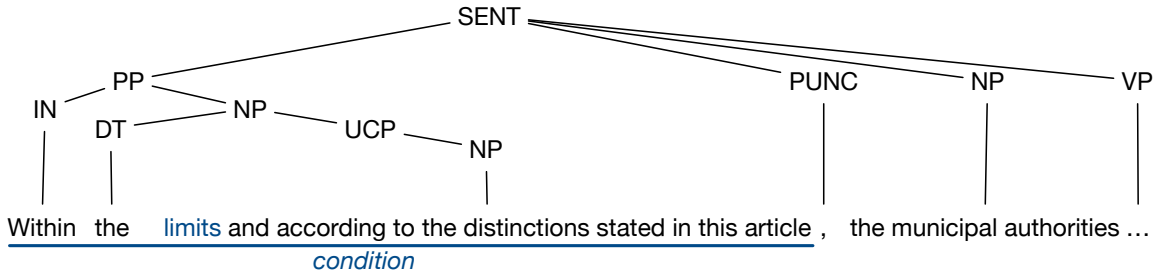

Fig. 4.3 Simplified Parse Tree for an Excerpt of Statement 1 from Fig. 4.1

Table 4.3 Markers for Different Metadata Types

Concept	Examples of Markers (Non-exhaustive)
Fact	
Definition	*include, exist, comprise, consist, designate, …*
Penalty	*is comdemned, is punished, is punishable, pronounces, is liable, …*
Permission	*may, can, is permitted, is authorized, …*
Obligation	*shall, must, is obliged, is compelled, is required, …*
Prohibition	*can not, is forbidden, is prohibited, is illegal, is not authorized, …*
Actor*	*physician, expert, company, judge, prosecutor, driver, officer, inspector, …*
Artifact§	*document, agreement, certificate, licence, permit, warrant, pass, …*
Condition†	*if, in case of, provided that, in the context of, limit, who, whose, which …*
Exception†	*with the exception of , except for, derogation, apart from, other than, …*
Location‡	*site, place, street, intersection, pedestrian crossing, railway track*
Modality†	*may, must, shall, can, need to, is authorized to, is prohibited from, ...*
Reason†	*in order to, for the purpose of, so as to, so that, in the interest of, in view of, …*
Sanction†	*punishment, jail sentence, imprisonment, prison term, fine, …*
Situation*	*renewal, inspection, parking, registration, deliberation, …*
Time†	*before, after, temporary, permanent, period, day, year, month, date, ...*
Violation†	*offence, crime, misdemeanor, civil wrong, infraction, transgression, …*

* The markers are not generic but are automatically derivable from a simple dictionary.
§ The markers are not generic but can be derived automatically if an ontology like WordNet's with an explicit classification of objects (human-made and natural) is available.
† The markers are mostly generic and expected to saturate quickly.
‡ The markers are in part domain-specific. Domain-specific markers need to be specified by subject-matter experts or be derived from an existing domain model (ontology).

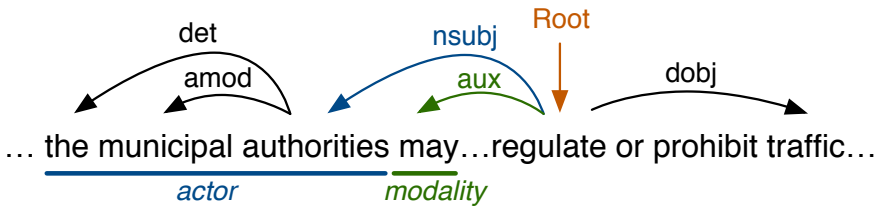

Fig. 4.4 Simplified Dependency Graph for an Excerpt of Statement 1

often closely". With "inspection" included in the situation markers, the rule for *situation* in Table 4.2, NP < (situation marker) would mark the noun phrase "vehicle inspections" in Statement 2 of Fig. 4.1 as a *situation*.

In a similar way, any Wiktionary entry classified as a noun and with a definition containing "person", "organization", or "body" (or variations thereof) counts as a marker for *actor*. For example, "authority" is an actor marker since Wiktionary defines it as "A *body* that enforces law and order [...]." As shown by the rules in Table 4.2, the mere presence of an actor marker does not necessarily induce an *actor* annotation: the candidate phrase must also appear in a subject or object dependency as defined by the rules. To illustrate, let us again consider Statement 1 of Fig. 4.1. A simplified dependency graph for an excerpt of this statement is provided in Fig. 4.4. Here, the actor annotation is extracted by the rule: subject dependency and NP < (actor marker). This rule classifies a noun phrase as an *actor* if the noun phrase contains an actor marker and if it has a subject dependency (*nsubj*) to the main (root) verb within the statement.

For *artifacts*, we need the ability to identify human-made objects. It is possible to develop generalizable automation for this purpose in the English language, with support from ontologies such as WordNet [69], providing a classification of objects. In place of such an ontology for French, we derived an initial set of markers from the 200 statements in our qualitative study. We then enhanced these markers by inspecting their synonyms in a thesaurus, and retaining what we found relevant. In addition, we implemented a heuristic (the second rule under *artifact* in Table 4.2) which classifies as *artifact* any noun phrase that is otherwise unclassifiable.

For *conditions*, *exceptions*, *modalities*, *reasons*, *sanctions*, *times*, and *violations*, the markers were derived from our study and later augmented with simple variations suggested by legal experts. As one can see from Table 4.3, noting the nature of the markers for these seven concepts, the number of possibilities is limited. While our qualitative study did not necessarily capture all the possibilities, we anticipate that the list of markers for these concepts will saturate quickly if enriched further.

Table 4.4 Rules for Extracting Statement-Level Semantic Legal Metadata

Statement Concept	Rule	Priority
Fact	Statement < (Fact Marker)	6
Definition	Statement < (Definition Marker)	5
Penalty	Statement < (sanction) OR Statement < (violation) OR Statement < (Penalty Marker)	1
Permission	Statement < (modality < (Permission Marker))	2
Obligation	Statement < (modality < (Obligation Marker))	3
Prohibition	Statement < (modality < (Prohibition Marker))	4

Finally, and with regard to the markers for *location*, we followed the same process as described above for *artifacts*, i.e., we derived an initial set of markers from the qualitative study and enhanced the results using a thesaurus. The resulting markers for *location* contain a combination of generic and domain-specific terms. For example, "site" and "place" are likely to generalize to legal texts other than traffic laws. In contrast, designating a "railway track" as a location is specific to traffic laws. The markers for *location* will therefore need to be tailored to the specific legal domains.

4.3.2 Statement-level Metadata Extraction Rules

Table 4.4 presents the rules that we have developed for classifying the statements as *facts*, *definitions*, *obligations*, *penalties*, *permissions* or *prohibitions*. The third column indicates the order in which the rules are applied. This ordering is meant at avoiding additional incorrect statement classifications when several rules apply to the same statement.

As we already explained in Section 3.2, it is quite common to rely on modal verbs like "shall" or "may" to determine the actual deontic nature of a legal statement [10, 37, 42]. Therefore, the classification of the three metadata types *obligation*, *permission* and *prohibition* is highly related to the phrase-level metadata *modality*. Since the use of modal verbs is not systematic in French, other cues or markers (see Table 4.3) are required, notably the interpretation of the main verb of the statement and the implicit modality it implies.

For *penalty*, we mainly rely on the presence of the metadata types *sanction* and *violation*. There also exist markers that directly indicate a *penalty* (see Table 4.3). For instance, cues such as "punishable" – and its synonyms – are good indicators.

For the remaining two metadata types, namely *fact* and *definition*, we found no obvious markers and therefore the classification relies on the interpretation of the main verb of the statement. From the initial observations, we initiated two lists of markers

(one for *fact* and one for *definition*). These lists are limited to the markers found in the qualitative study and their synonyms, and are therefore not exhaustive. We expect the list to saturate quickly over a limited set of qualitative studies on legal acts. For example, a main verb such as “include” can be a good indicator for a *definition*.

In addition to the rules, we also devised a prioritization among the different statement-level concepts. Indeed, *penalty* statements may also have a modal verb such as ”may” or ”shall”, but the presence of a sanction and a penalty cue plays a decisive role in classifying the statement as a *penalty*. Therefore, *penalty* has a higher priority than *obligation*, *permission*, and *prohibition*. An example of such prioritization being triggered for a penalty can be found in Statement 2 of Fig. 4.1. Finally, if none of the previous rules apply, we attempt to trigger the rules for *definition* and *fact*.

During the qualitative study, we observed that obligations are the most common statements in legal texts. Often (especially in French), the modal verb for expressing obligations is left implicit and therefore cannot be detected by NLP parsers. For this reason we took a design decision to classify by default as *obligation* all the declarative statements that do not contain any cue or marker previously discussed.

4.3.3 Actor’s Role Extraction using Machine Learning

The rules we presented earlier do not account for the roles (subtypes) of *actor*, namely *agent*, *target* or *auxiliary party*. When analyzing a statement, a human annotator is able to further classify actors into its subtypes, thanks to their knowledge of the language and their ability to consider various semantic characteristics of sentences as well as the phrases in the proximity of the actors themselves. Translating such knowledge into intuitive rules, similar to those described above, is possible only for simple and straightforward statements. However, the task becomes challenging when facing the subtleties and complexity of legal language. Nevertheless, an ML algorithm can learn the logic applied by the experts by extracting, from a set of examples, a combination of linguistic elements that a “simple” NLP extraction rule would fail to capture. A good candidate to determine these three *actor* roles is semantic role labeling (SRL) [70]. SRL aims at automatically finding the semantic role for each argument of each predicate in a sentence. However, the intrinsic limitation of SRL is that it is built to work on relatively simple sentences [71], and is thus not adapted to long and complex sentences such as those that can be found in legal texts. A second limitation is that the labeling of the inferred roles is based on the analysis of the main verb and provides specific semantic roles (e.g., the agent for the main verb ”buy” is labelled ”buyer”), whereas our conceptual model revolves around three general concepts (*agent*,

target, *auxiliary party*). Besides, semantic role labelling frameworks for French such as Verbenet [72] are not likely to yield accurate results for legal texts. Given the arguments above, building an ML-based classifier is a natural course of action. We also want to investigate if we can encapsulate the actor's role extraction into a set of features that an ML technique could combine to yield correct classifications.

Feature selection. To train our classifiers, we rely on 31 features, grouped under three categories. These features are shown in Table 4.5 and described below. They are derived from the linguistic characteristics of the sentence as well as from phrase- and statement-level annotations.

Sentence-level features. This set of five features provides information about the statement itself. The first feature is the (active or passive) voice of the main verb of the statement, which has influence over the identification of the semantic subject(s) and object(s) of the statement. The second feature concerns another aspect of the main verb of the statement, namely its transitivity. A verb can be transitive, intransitive or both. This information is extracted from an open source dictionary for French (the French version of Wiktionary) and gives information about the subject and potential object. The third feature returns the modal verb if the statement contains one. A modal verb can be categorized as a marker for permission, obligation or prohibition. The list of modal verbs was manually constructed from the ground truth, and validated by legal experts. An example of such list for obligations is available in Annex A. This feature returns an enumeration of the modal verbs found in the statement, or NULL if the statement does not contain any. The fourth feature is the number of actors in the statement. This information is used in combination with the fifth feature, which identifies the actor that is analyzed in the remainder of the features by indicating its position within the list provided by the fourth feature. When combined with the other features, this indication can provide insights over the role of the actor under analysis.

Actor's Neighborhood features. This set of four features categorizes the neighborhood relations of the actor to be classified. The first feature specifies if the actor is contained within another phrase-level annotation. This feature is based on the results of our qualitative study, which showed that agents are usually not contained in other metadata types, while targets are usually contained within the action, and auxiliary parties are easily recognizable if found inside a condition, exception, reason or reference. The second and third features concern the preceding and following phrase-level annotations, respectively. Our qualitative study showed in fact that an empty actor neighbourhood would hint to an agent classification. The fourth feature specifies the type of the preceding POS tag in the sentence (e.g., a preposition). This

Feature Group	Feature	Description
Sentence information	active voice	indicates the voice of the statement (passive (false) or active(true))
	transitivity of the main verb	indicates the transitivity of the main verb as determined from Wiktionary {'transitive'; 'intransitive'; 'both'}
	modal verb	returns the modal verb if retrieved from a modal verb list, 'null' otherwise
	number of actors	number of actors in the statement
	position of the actor	position of the actor in the list of actors
Actor Annotation Neighborhood	Annotation Container	indicatesthe annotation that contains the Actor annotation, 'null' otherwise
	Preceding Annotation	indicates the annotation that precedes the actor annotation
	Following Annotation	indicates the annotation that follows the actor annotation
	Preceeding POS tag	POS tag that preceedes the actor annotation
Main Verb Relationship	distance to the main verb	distance in terms of annotations to the main verb
	dependency chain	dependency chain from the actor to the main verb
	SUJ	number of instances of the dependency subject in the dependency chain
	OBJ	number of instances of the dependency direct object in the dependency chain
	ATS	number of instances of the dependency predicative complement of a subject in the dependency chain
	ATO	number of instances of the dependency predicative complement of a direct object in the dependency chain
	MOD	number of instances of the dependency modifier or adjunt in the dependency chain
	A-OBJ	number of instances of the dependency indirect complement introduced by à in the dependency chain
	DE-OBJ	number of instances of the dependency indirect complement introduced by de in the dependency chain
	P-OBJ	number of instances of the dependency indirect complement introduced by another preposition in the dependency chain
	DET	number of instances of the dependency determiner in the dependency chain
	DEP	number of instances of the dependency in the dependency chain
	PONCT	number of instances of the dependency punctuation in the dependency chain
	ROOT	number of instances of the dependency root in the dependency chain
	DEPCOORD	number of instances of the dependency coordination in the dependency chain
	COORD	number of instances of the dependency coordination in the dependency chain
	AUXPASS	number of instances of the dependency passive auxiliary in the dependency chain
	AUXCAUS	number of instances of the dependency causative auxiliary in the dependency chain
	AUXTPS	number of instances of the dependency tense auxiliary in the dependency chain
	AFF	number of instances of the dependency affix in the dependency chain
	ARG	number of instances of the dependency argument in the dependency chain
	MODREL	number of instances of the dependency relative modifier in the dependency chain

Table 4.5 Classification Features

makes it possible to exclude the classification as agent of actors that are preceded by a proposition.

Main-verb Relationship features. This set of 22 features categorizes the relationship between the actor to be classified and the main verb of the legal statement. The first feature is the distance, in terms of number of annotations, between the actor to be classified and the main verb of the statement. For a legal statement, the main verb is recognized through dependency parsing, as previously explained. The next feature is the dependency chain connecting the actor to the main verb, expressed using the information from the dependency graph. This results in an ordered list of dependency types. The remaining 20 features concern the number of instances of a given dependency type in the dependency chain. These dependencies are those that are relevant and that we can extract through the dependency parser. They are similar to the dependencies used, for instance, for Verbenet [72], but adapted to our own parser (the implementation details are elaborated in Section 5.3.1).

The presence or absence of a given dependency type and its number of occurrences in the dependency chain, in combination with other features from the actor's neighborhood, can direct the classifier to classify the actor as agent, target, or auxiliary party. For example, the presence of a subject dependency in the dependency chain attached to the *actor* annotation, in addition to an active voice from the sentence level features, is likely to trigger the classification of the actor in question as *agent*. On the other hand, the presence of a multitude of dependency types with a high number of occurrences of modifiers and object dependencies might lead to assigning the role *auxiliary party* to the actor.

Dataset. A key consideration in ML-based classification is the size of the dataset that is needed for training a classifier. In addition to the initial 200 statements of our qualitative study, which contained 149 actors (including 42 agents, 34 auxiliary parties, and 73 targets – see Table 4.1), we annotated 503 additional statements in order to obtain an enhanced dataset of 1000 actors. This enhanced dataset is composed of 183 agents, 481 targets, and 336 auxiliary parties.

Training. We use our dataset to train three classifiers to classify *actors* as *agent*, *target*, and *auxiliary party*, respectively. We use WEKA, which is an open source platform for data mining activities [73]. We employ ten-fold cross validation over our dataset to evaluate prediction performance of our classifiers. We evaluate several classification algorithms including Naive Bayes classifier, Decision Trees, Random Forest, and Support Vector Machine (SVM). In order to effectively identify the most suitable ML algorithm and optimize the hyper-parameter settings, we run Auto-WEKA [74], a tool

to automatically select the optimal classification algorithm among those implemented in the WEKA package using Bayesian optimization.

Overall, Naive Bayes did not yield accurate predictions. This was expected, since Naive Bayes treats features as independent, whereas our features are strongly interconnected. Similarly, Decision Trees did not perform well in our classification task. SVM is not suited for a mix of numerical and categorical features, as it is the case in our dataset. Random Forest (RF) returned actor classifications of a much higher quality than the other ML techniques evaluated by Auto-WEKA. Overall, the ten-fold cross-validation of the RF classifiers led to accuracy results, i.e., the ability to correctly classify annotations as being or not of a given type, of 90.2% for the agent classifier, 79.0% for the target classifier and 78.3% for the auxiliary party classifier, respectively.
Making the final classification decision. In our approach, each actor annotation in a statement is submitted for classification over the three classifiers. The final classification decision algorithm is described in Fig. 4.5. Essentially, our final classification decision favors the highest confidence score, with some additional heuristics being applied:

1. the classifier will favor agent if its confidence score for agent is high;
2. a direct classification is made if the best score is already high and there is a sufficient difference between the best and second best classification scores;
3. when having to choose between *target* and *auxiliary party*, if the difference between the two scores is small, then the classifier will not make a decision ("cannot_classify").

Our heuristics are based on the accuracy measured during the initial cross-validation step, which makes us confident about the classifier for agent, but less so about the classifiers for target and auxiliary party. These two roles are in fact often ambiguous and therefore hard to distinguish. For instance, in statement 3 of Fig. 4.1, the roles of the public prosecutor and the sued person are not straightforward to establish.

4.4 Tool Support

Implementation. Our metadata extraction rules are implemented using Tregex [30] and Java. These rules utilize the outputs of the classic NLP pipeline for syntactic analysis. The pipeline described in Fig. 4.6 has the following modules: Tokenizer,

Alg. 1: Generate an actor classification

Inputs:

(1) an actor_annotation to be classified,
(2) SA is the score returned from the agent classifier;
(3) ST is the score returned from the target classifier;
(4) SAux is the score returned from the auxiliary party classifier;
(5) an acceptance threshold T1 for SA;
(6) an acceptance threshold T2 for SA when SA is not the maximum confidence score among the three scores;
(7) an acceptance threshold T3 for uncertainty when we cannot make a classification decision;

Output: a classification of the actor annotation

function GenerateClassification.
if *Agent_Acceptance* **then return** "Agent" .
else if *Uncertainty_Condition* **then return** "Cannot_Classify"
else if *Target_Acceptance* **then return** "Target".
else return "Auxiliary_Party"
end function

function Agent_Acceptance // *generates a Boolean to decide if the actor annotation should be classified as an agent.*
return (SA>T1 or SA==S1 or (SA==S2 and (S1−S2<T2))).
end function

function Target_Acceptance //*generates a Boolean to decide if the actor annotation should be classified as a target.*
return ST == S1.
end function

function Uncertainty_Condition // *generates a Boolean to decide if the actor annotation can not be classified among the three classes*
return S1 − S2 < T3
end function

function S1 // *returns the best confidence score among the three classification models.*

function S2 // *returns the second-best confidence score among the three classification models.*

Fig. 4.5 Final Classification Decision Algorithm

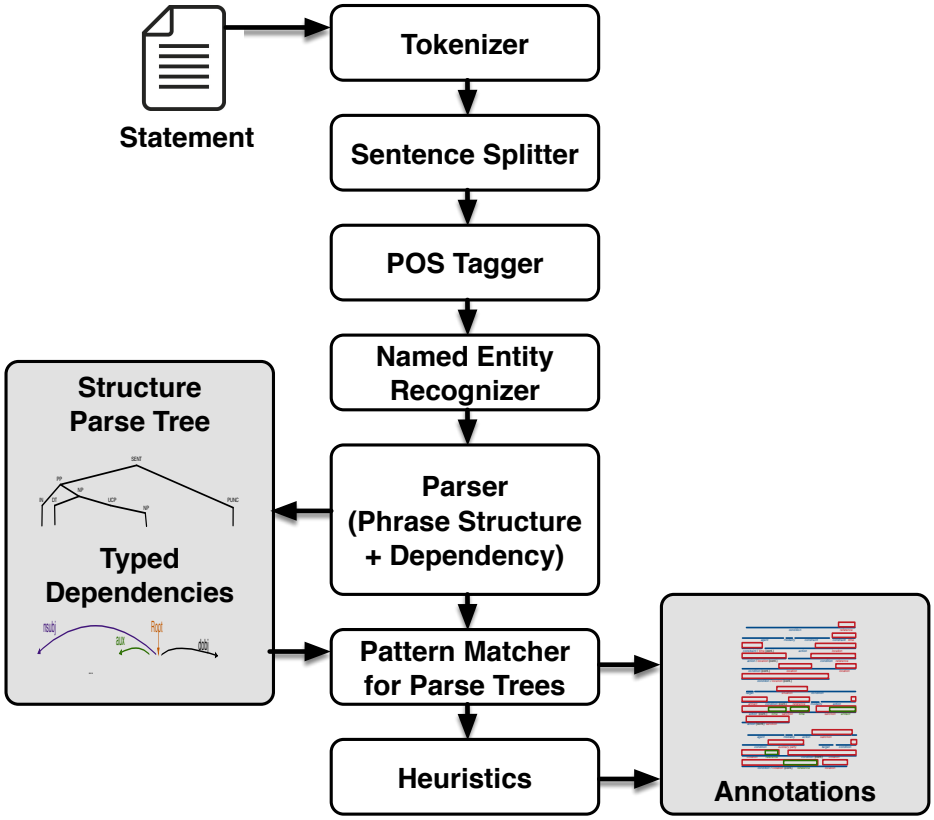

Fig. 4.6 Tool support

Sentence Splitter, POS Tagger, Named-entity Recognizer, and (Constituency and Dependency) Parser.

Alternative implementations exist for each of these modules. We instantiate the pipeline using a combination of module implementations which we found to be the most accurate for the language of the legal texts in our context, namely French. For the lexical analysis modules, we use standard libraries from Python for the Tokenizer and the Sentence Splitter. With regard to POS Tagging, we use a language-specific framework called Lefff [75], while we base our Named-Entity Recognition on our specific sets of markers. For constituency and dependency parsing we use the Berkeley Parser [76] and the Malt Parser [77] respectively.

Design heuristics. The heuristics that we have developed (such as wrapping together annotations of the same type that are next to each other, or prioritizing classifications) are related to strategies for avoiding the over-generation of annotations.

The first heuristic is related to the presence of two overlapping annotations of the same type. For example, consider the annotation "the director of the grand-ducal police". In this example, we have two actor markers, one for "the director" and one

for "the grand-ducal police". Instead of generating two annotations ("the director of the grand-ducal police" and "the grand-ducal police"), we only consider the one with the longest span, since it identifies more accurately the entity that plays a role in the statement.

The second heuristic is similar to the previous one. It is related to the presence of annotations of different types, where one contains the other. Annotations can contain other annotations, (e.g., a *condition* annotation can include an *actor* annotation). However, annotations contained in *references* do not make sense for the statement analysis. Take for example the "Directive 2014/65/EU of the European Commission...". In this case, "the European Commission" should not be annotated as an *actor*, since it is not related to the action expressed in the statement. We discard annotations of type *actor*, *time*, *location*, *artifact* and *condition* when they are contained within a *reference* annotation.

The third heuristic is related to hierarchically-ordered annotations spanning the same content. For instance, the annotation "an imprisonment" is annotated as both a *situation* and a *sanction*. However, following our previously described conceptual model, *sanction* is a kind of *situation*. In these cases, we discard the annotation of the most generic type (in this example, *situation*).

The last heuristic is related to ambiguity. For instance, let us consider the marker "court of justice": it can mean either the authority (*actor*) delivering the judicial decision, or the *location* where such decision is taken. While a human is able to infer the correct meaning from the general context, automating this task is very difficult since in many cases the sentence does not provide any linguistic cue to help with the disambiguation. Consequently, we decided to assign the ambiguous marker to both metadata types. Although this decreases the overall precision, this marker at least restricts the choice to two metadata types instead of leaving all the 18 classification possibilities open for that particular phrase.

4.5 Empirical Evaluation

In this section, we measure the accuracy of our extraction rules through two case studies.

4.5.1 Research Questions

Our evaluation is targeted at answering the following research questions (RQ):

RQ1. How accurately can our approach extract semantic legal metadata when it is employed in the same domain as our qualitative study? This RQ aims at evaluating the completeness of our extraction rules, markers, and classification techniques, for the extraction of both statement-level and phrase-level metadata when applied to statements in our initial analysis domain, i.e., the traffic laws.

RQ2. How accurately can our approach extract semantic legal metadata when it is employed outside the domain where we conducted our qualitative study? This RQ is an attempt toward evaluating the generalizability of our approach by applying it to different domains within the Luxembourgish legislation.

4.5.2 Case Studies Description

To answer our research questions, we have set up two different case studies that we describe below.

Case study 1 (CS1). The objective of this case study is to measure the accuracy of the extraction rules of Table 4.2 against a ground truth. To build the ground truth, we manually annotated 150 randomly selected legal statements from the traffic laws, in addition to the 200 statements previously annotated for our qualitative study of Section 4.3. We followed the same protocol coding process as described in our qualitative study. The construction of the ground truth took place strictly after the conclusion of our qualitative study. Specifically, our extraction rules (including the concept markers) were already finalized and frozen at the time when we selected and analyzed the 150 statements. The ground truth was constructed in two rounds. In the first round, we annotated 100 statements and performed a complete round of evaluation, following the same procedure that we explain below. Our analysis of the results in the first round did not lead to new extraction rules, but prompted marginal improvements to the concept markers for *condition*, *time*, and *location* (see Table 4.3). Following the first evaluation round, we annotated another 50 statements and measured the accuracy of our improved solution over them. We obtained accuracy levels similar to those of the first round. This provides confidence that our extraction rules and markers have saturated. We report the evaluation results for the 100+50=150 statements combined. To avoid biased conclusions, the reported results use the baseline set of concept markers, i.e., the same set with which the first evaluation round was performed.

The first researcher annotated the 150 statements used in the evaluation, and the second researcher independently annotated 10% of these statements to examine reliability. We obtained $\kappa = 0.815$, suggesting "almost perfect agreement" [65]. In total, the ground truth has 1202 annotations covering 1177 phrases (25 phrases have double annotations). A detailed breakdown is provided in the ground truth column of Table 4.6. Similar to the qualitative study, we observed no occurrences of *result* and a very low number of occurrences of *constraint*.

To evaluate our extraction rules, we exclude occurrences of *constraint* and *result* for which we do not provide rules, and occurrences of *reference*, whose detection we leave to existing solutions. Our evaluation is thus based on 1127 ground-truth annotations.

Case study 2 (CS2). The objective of this case study is to explore the generalizability of our approach for semantic metatadata extraction in different domains of the legislation, and to investigate how the rules and markers that we have acquired in the qualitative study and in the first case study can expand to other legislative domains. To do so, we randomly selected 40 statements from five Luxembourgish legislative codes, including the Code of Commerce, the Penal Code, the Code for healthcare, the Labor Code and the Code for the Environment, for a total of 200 statements. Among these codes, each of the first two acts is a single harmonized legal act, whereas each of the last three is a collection of legislative acts of various nature (laws, regulations, etc.). These codes possess their own terminology, thus providing an interesting context in which to investigate how important domain knowledge is in the automatic extraction of semantic metadata.

Regarding the manual construction of the second ground truth, the 200 statements were manually annotated by the second researcher. This is to mitigate the potential bias due to the fact that the implementation of the tool was performed by the first researcher. Similarly to the first case study, the first researcher independently annotated 10% of the statement to assess reliability. We obtained $\kappa = 0.813$, suggesting "almost perfect agreement" [65]. In total, the ground truth for this case study has 2132 annotations, covering 1974 phrases (158 phrases have double annotations). The detailed breakdown is provided in the ground truth column of Table 4.7.

Similarly to the first case study, and for the same reason, we do not report on *constraint*, *result* and *reference*.

4.5.3 Analysis Procedure

Each phrase-level annotation has two parameters: a *type* and a *span*. The former specifies the legal concept, while the latter specifies where the annotation begins and

where it ends in the statement. We evaluate the results of automated phrase-level metadata extraction using the following notions:

- A computed annotation is a *match* if its span has a non-empty intersection with some ground-truth annotation of the same type.
- A computed annotation is *misclassified* if it is not a match.
- A ground-truth annotation for which there is no match is considered as *missed*.

For sentence-level annotations, only the *type* parameter is pertinent; the *span* is implied since the annotation covers the entire statement. We therefore evaluate the results of automated sentence-level metadata extraction using the following notions:

- A computed annotation is a *match* if its type matches that of the ground-truth annotation for the same statement.
- A computed annotation is *misclassified* if it is not a match.
- A ground-truth annotation for which there is no match is considered as *missed*.

Our evaluation results are presented in columns 3 through 8 of Table 4.6. For each legal concept (metadata type), we provide the number of matches, the number of misclassified annotations, the number of missed annotations, and scores for precision and recall. Each match counts as a true prositive (TP); each misclassified annotation counts as a false positive (FP), and each missed annotation counts as a false negative (FN). Precision is computed as $|\text{TP}|/(|\text{TP}| + |\text{FP}|)$ and recall as $|\text{TP}|/(|\text{TP}| + |\text{FN}|)$.

4.5.4 Results for the First Case Study

We first discuss the results for the classification of phrase-level metadata, except actors. This is followed by the results for the classification of actors (into sub-types) and statements. The classification of both actors and statements relies on other phrase-level metadata (see Sections 4.3.2 and 4.3.3). Finally, we perform an error analysis of the misclassifications and the missed metadata types.

Results for phrase-level metadata. In summary, out of the 1100 computed annotations, 1069 annotations were correct matches and 31 (2.8%) were misclassified. There are 58 ground-truth annotations (5.1%) that were missed by the extraction rules, due to either misclassification or the impossibility to classify (i.e., the span of the ground-truth annotation lacks a computed annotation altogether). The impossibility

Table 4.6 Statistics for Automated Semantic Metadata Extraction (CS1)

Legal Concept	Ground Truth	Results of Automatic Metadata Extraction					Accuracy	
		Extracted	Perfect Match (TP)	Partial Match (TP)	Misclassified (FP)	Missed (FN)	Precision	Recall
Definition	13	11	9	N/A	2	4	81,8%	69,2%
Fact	2	--	--	N/A	--	2	N/A	N/A
Obligation	114	122	107	N/A	15	7	87,7%	93,9%
Penalty	30	29	27	N/A	2	3	93,1%	90,0%
Permission	33	28	28	N/A	0	5	100,0%	84,8%
Prohibition	8	10	7	N/A	3	1	70,0%	87,5%
Subtotal	200	200	178	N/A	22	22	89,0%	89,0%
Action	165	169	3	148	18	14	89,3%	91,5%
Actor	526	521	169	328	24	29	95,4%	94,5%
Agent	114	111	99	N/A	12	15	89,2%	86,8%
Aux. Party	239	251	210	N/A	41	29	83,7%	87,9%
Target	173	159	135	N/A	24	38	84,9%	78,0%
Artifact	319	321	89	204	168	26	91,3%	91,8%
Condition	301	321	88	164	69	49	78,5%	83,7%
Constraint	55	--	--	--	--	--	N/A	N/A
Exception	14	12	8	3	1	3	91,7%	78,6%
Location	87	99	17	65	17	5	82,8%	94,3%
Modality	72	102	61	10	31	1	69,6%	98,6%
Reason	18	27	11	3	13	4	51,9%	77,8%
Reference	97	--	--	--	--	--	N/A	N/A
Result	7	--	--	--	--	--	N/A	N/A
Sanction	57	58	16	40	2	1	96,6%	98,2%
Situation	249	272	129	117	26	3	90,4%	98,8%
Time	132	145	36	92	17	4	88,3%	97,0%
Violation	33	26	7	15	4	11	84,6%	66,7%
subtotal	2132*	2073	634	1189	390	150	82,4%	92,4%

*We exclude from our evaluation contraints, references and results as noted in the text: we do not have extraction rules for constraints and results; detecting (cross-)references is outside the scope of this article. This leaves us with 2132-55-97-7 = 1973 relevant annotations.

to classify was mostly due to parser errors, or to missing Tregex patterns (or markers) that are either too rare or inconclusive, i.e., they are not unambiguous enough to be used as patterns (or markers) for classification. We obtain an overall precision of 97.2% and an overall recall of 94.9%.This means that our approach identifies the types of metadata items with very high accuracy. Analysts can thus expect to have a correct type assigned automatically in the large majority of cases.

Results for actor classification. The results from the automated classification of actors are aligned with what we observed during the training of the classifiers and the 10-fold cross validation. The results are far better than in our initial evaluations based on rules, with increases of $\approx 30\%$ for precision and $\approx 40\%$ for recall when detecting the sub-types of actor. Still, our error analysis highlights that the large majority of issues are related to ambiguities between *targets* and *auxiliary parties*. Consider for example the following complex statement related to driving a car rented in a foreign country (simplified): "Any vehicle belonging to *a physical or moral person* having their main residence or office located in another European State [...] and allowed to perform car leasing can, on the basis of the registration document drawn up by *the competent authorities of that State*, be put into circulation on the public roads of Luxembourg by *any person* having its residence or office in Luxembourg if this vehicle was made available to *this person* through a lease [...]." In this statement we have italicized the different *actor* annotations, namely: "a physical or moral person", "the competent authorities of that State", "any person", and "this person". Here the last annotation ("this person") refers to the previous one ("any person"), and both should therefore be classified as *target*. However, our classifier wrongly considers "any person" to refer to the first two annotations, which are *auxiliary parties*, thus generating a wrong classification for the last two actors.

Disambiguating *targets* from *auxiliary parties* in those situations would require a representation of domain semantics, e.g., through a taxonomy, which is left for future work and thus not addressed here.

Results for statement-level metadata. In summary, out of the 150 computed statement-level annotations, only two are incorrect, implying two annotations being misclassified (FP) and an equal number of missing (FN) classifications. In both cases, the tool triggered an obligation whereas a prohibition was the choice of the annotator. This is due to the lack of context: to capture those annotations it is necessary to take into account the preceding statements, which were implicitly taken into account by the annotator, but that the tool is unable to handle. For instance, one of the statements began with "It is the same for ...". In the absence of any other cue, the rule triggered

the default choice, i.e., *obligation*. The preceding statement was instead a prohibition, but the tool lacked this information.

Discussion. To answer RQ1, the results presented above show that our approach is highly accurate when employed in the same domain as our qualitative study. To determine the root causes for the automation inaccuracies observed, we analyzed the misclassified and missed annotations. Of the 31 misclassifications in Table 4.6, 20 are related to polysemous concept markers. For example, the term "seizure" is a marker for *sanction*, since the term may refer to the confiscation of a possession. This term may also refer to a medical symptom, in which case it suggests a *situation*. Both senses of the term are used in the traffic laws. When the term is used in the latter sense, our rules generate a misclassified annotation. Three misclassifications arise from complex legalese and are thus unavoidable. The remaining eight misclassifications are due to constituency parsing errors, discussed later.

Of the 58 missed annotations, 25 are related to double annotations in the ground truth. In all these cases, our rules identify one of the two ground-truth annotations, but we still count one FN for each case, since, compared to a human annotator, the rules lack the ability to take multiple perspectives into account. Among the remaining 33 missed annotations, 26 are due to misclassifications, discussed earlier. Five missed annotations result from a pair of distinct ground-truth annotations that our rules grouped into a single annotation, that intersects with both ground-truth annotations. Each of these five cases leads to one match and one missing annotation. The last two missed annotations are caused by constituency parsing errors, discussed later.

4.5.5 Results for the Second Case Study

Similarly to the first case study, our evaluation results for the second case study are presented in columns 3 through 8 of Table 4.7 and follow the same presentation order. For the second case study, the results of the evaluation are analyzed in comparison with the results of the first case study. By doing so, we attempt to assess how our findings within the legal domain of our qualitative study – which is the same as that of our first case study – would generalize to other legal domains. The error analysis for each part is directly discussed, while a broader discussion is performed at the end of the section.

Results for phrase-level metadata. When tested against the new multi-domain ground truth, the set of general markers and rules that we have developed maintained good results in terms of recall but not in terms of precision. In summary, over the 2213 computed annotations, 1823 annotations were correct matches and 390 (17.6%) were

Table 4.7 Statistics for Automated Semantic Metadata Extraction (CS2)

Legal Concept	Ground Truth	Results of Automatic Metadata Extraction				Accuracy	
		Extracted	Match (TP)	Misclassified (FP)	Missed (FN)	Precision	Recall
Definition	13	11	9	2	4	81,8%	69,2%
Fact	2	--	--	--	2	N/A	N/A
Obligation	114	122	107	15	7	87,7%	93,9%
Penalty	30	29	27	2	3	93,1%	90,0%
Permission	33	28	28	0	5	100,0%	84,8%
Prohibition	8	10	7	3	1	70,0%	87,5%
Subtotal	200	200	178	22	22	89,0%	89,0%
Action	165	169	151	18	14	89,3%	91,5%
Actor	526	521	497	24	29	95,4%	94,5%
Agent	114	111	99	12	15	89,2%	86,8%
Aux. Party	239	251	210	41	29	83,7%	87,9%
Target	173	159	135	24	38	84,9%	78,0%
Artifact	319	461	293	168	26	63,6%	91,8%
Condition	301	321	252	69	49	78,5%	83,7%
Constraint	55	--	--	--	--	N/A	N/A
Exception	14	12	11	1	3	91,7%	78,6%
Location	87	99	82	17	5	82,8%	94,3%
Modality	72	102	71	31	1	69,6%	98,6%
Reason	18	27	14	13	4	51,9%	77,8%
Reference	97	--	--	--	--	N/A	N/A
Result	7	--	--	--	--	N/A	N/A
Sanction	57	58	56	2	1	96,6%	98,2%
Situation	249	272	246	26	3	90,4%	98,8%
Time	132	145	128	17	4	88,3%	97,0%
Violation	33	26	22	4	11	84,6%	66,7%
subtotal	2132*	2213	1823	390	150	82,4%	92,4%

*see foot note in Table 4.6. Here, this leaves us with 2132-55-97-7 = 1973 relevant annotations.

misclassified. Our rules missed 150 ground-truth annotations (7.6%), due to either misclassification or impossibility to classify (due to parser errors, or to missing Tregex patterns or markers). The overall precision and recall for phrase-level annotations reach 82.4% and 92.4%, respectively. The overall recall for this case study (92.4%) is comparable to the first case study (94.9%); nevertheless, we observe a decrease in precision from 97.2% to 82.4%. As the following error analysis and discussion show, this decrease can be attributed to the increased length of the statements (see Table 4.8) and to the domains being different.

With regard to *artifact*, the lower precision score is related to the absence of markers for other elements, leading to *artifact* as a default classification. In fact, since the domains are different from that of our qualitative study, we expected that our list of general markers would not be able to cover all the new terms.

With regard to *condition*, most of the misclassifications are related to our inability to extract *constraints*, which led to considering them as *conditions* instead. These new occurrences of *constraint* can however be the basis for the elaboration of appropriate extraction rules for this concept. The missed annotations of *condition* are related to cues that cannot be considered as specific markers as-is, as they need to be associated with a deeper NL analysis of the statement. For instance, phrases such as "by (him)", "within", "(member) of", "under (the name)" can hardly be considered as viable cues for *condition* on their own, since they can also be used in other contexts.

With regard to *exception*, the number of occurrences is low. From the initial observations, it emerged that their classification mostly depends on the context of the preceding statements. Because our approach treats each statement in isolation, it is unable to correctly identify such annotations.

With regard to *location*, issues are related to the ambiguity between the location and the authority (actor) who has its premises in that location, since both are referenced using the same term (e.g., a court of justice, school, or third party country).

The issues for *modality* are related to the improper handling of modal verbs which are not the main verb of the statement. These modal verbs in fact should not be classified as *modality* since they have no effect on the classification of the statement that contains them. For instance, in the statement "the officer notifies the person that she *can* ...", the permission modality "can" does not affect the main verb ("notify"), which instead bears an implicit obligation. Our markers and rules however do not take this into account, leading to mistakes at the statement level.

Concerning *reason*, we only have few occurrences, and a deeper analysis of the structure and the content of the statement is thus required in the future.

With regard to *violation*, we had few observations in the traffic laws, and could therefore only extract a limited set of markers for this metadata type. In addition, many violations are vague and implicit, and are understood as such only because of the context of the statement. For instance, "entering a (restricted) area" (implicitly, without authorization) does not provide sufficient cues about a possible *violation*.

Results for actor classification. The results for the automated classification of actors in the second case study are consistent with the results in the first case study, although we do observe a slight decrease (2%) in recall. Regarding the classification of actors as *agents*, *targets*, and *auxiliary parties*, we note that the evaluation takes into account the actors that we failed to classify as such in the first extraction phase. They are therefore counted as false negatives. Regarding the classification of actors, results for *target* and *auxiliary party* have improved though those for *agent* have worsened.

Results for statement-level metadata. We do not tackle the *fact* type in our statement-level metadata; as noted in Section 4.3.2, we do not have rules for classifying statements of this type. Our main observation in relation to the remaining statement-level metadata types is that our rules did not maintain precision and recall as high as in the first case study, with both scores decreasing from 98,7% to 89%. When analyzing false positives and false negatives, the following explanations emerge. With regard to *definitions*, we only had few cues from the initial qualitative study and the first case study. We encountered more *definitions* in the second case study, which gave us access to more markers, and this circumstance may improve the results in future iterations. Nevertheless, it often happens that a sentence does not contain unambiguous cues that can lead to its classification as a definition, whereas a human can still interpret it as such. An example is the verb "to be", which is used to express classification (and therefore definition) in sentences such as "Traders are those who buy and sell goods, currency, or shares". In these cases, the classifier fails to identify the statement as a *definition*, but in the presence of more unambiguous verbs such as "consider", which is a viable *definition* marker ("Traders are *considered as* those who buy and sell goods, currency, or shares"), it would have succeeded.

Similarly to the first case study, the classification of *prohibition* was affected by implicit information from a previous statement, which our rules do not account for. With regard to *permission*, the results are affected by a missing marker ("is likely to"), which denotes the possibility of a future action to be performed, as well as by the issue for *modality* when in presence of modal verbs that are not the main verb of the statement (as noted above). Obviously, *penalty* statements are affected by the ability to detect *violation* and *sanction* annotations. Finally, the precision for *obligation*

Table 4.8 Average Statement Length in CS1 and CS2

	Average Statements Length (#Words)	Standard deviation (#Words)
Traffic Code	35.3	21.2
Code of Commerce	56.4	32.3
Environmental Code	66.0	42.3
Penal Code	69.9	35.4
Health Code	49.0	34.3
Labour Code	58.0	38.5

has decreased, which is a direct consequence of the previous misclassifications, since obligation is a default classification in our approach (see Section 4.3.2).

Discussion. Compared to the first case study, the main reason for the drop in precision is related to the length of the statements. Table 4.8 shows the average length in words of the statements for our case studies and the standard deviations. We observe a significant length increase in the second case study, where statements tend to be longer, more nuanced and more complex than in the first case study. Here, longer statements are related to the introduction of long cross-references but also due to the presence of conditions and constraints that apply to specific elements in the statements. These additions also tend to break the flow of the statement.

As indicated previously, constituency and dependency parsers are more prone to errors when the length of the sentence increases. Beyond $\approx$ 30-35 words, parsers' accuracy starts to decrease rapidly [78, 79]. In the first case study, the average length of statements is already reaching the upper limit. In the second case study, the average length is significantly above this limit and therefore parsers' accuracy is likely to drop. This is due to the nature of the training data over which existing parsers have been trained. The training data is traditionally extracted from newspaper articles, with relatively short or medium-sized sentences written in a style that is not as formal as in legal texts. Accurate parsing of complex and lengthy legal statements would require dependency and constituency parsers to be trained specifically over legal texts. Labeling a sufficiently large corpus of legal texts for training parsers would necessitate a substantial amount of manual effort and is beyond the scope of this thesis.

Problems in detecting the markers in the right parse trees may negatively influence the quality of the annotations in the vicinity of the missed marker. This is due to the combination of the second extraction rule for artifact in Table 4.2 and the first two heuristics presented in Section 5.3.1. The extraction rule automatically classifies a phrase-level concept as artifact by default, i.e., if no other classification rule can be applied. The two heuristics prevent the addition of annotations within a span that is

already annotated as either actor or artifact, in order to limit the over-generation of annotations. As a consequence, if a marker is missed in the parse tree, the concept will be misclassified as an artifact and any other annotation that is contained within (or overlaps with) the misclassified annotation will be discarded altogether.

Furthermore, nuanced and complex statements are likely to present partial semantics and implicit information, especially when they are the continuation of previous statements from which they borrow part of the context. An example is the statement "The request must be sent to the competent tribunal". In this sentence, we lack the identity of the agent, the type of the request, and the conditions under which such a prescription applies. All this information is located in the same legal text, but in a previous statement. While a human analyst is able to locate such information, our approach is unable to do so because it treats each statement in isolation. Addressing this issue would enable a better detection of metadata when facing incomplete information, and will be part of our future work.

Regarding the influence of domain knowledge and the accuracy of our rules, since we did not perform any qualitative analysis over the new domains in the second case study, our rules and markers are based only on our initial observations from the traffic laws, and on the general knowledge that we have extracted from Wiktionary and other synonyms dictionaries (see Section 4.3.1). To answer RQ2, the results of the second case study did not prompt the need for modifying or adding new rules (except for *constraint*), and only prompted a limited need for updating the markers for some concepts.

As a final remark, in the absence of explicit markers, classifiers are unable to automatically process legal concepts and legal statements if relying only on the linguistic characteristics of sentences. This is particularly the case for ambiguous or polysemous words, and for statements such as *definitions* and *violations*, when cues may be too generic for use as reliable markers (e.g., "to be" for *definition*) or when a simple situation (e.g., entering a location) has to be interpreted as a violation because of implicit contextual information (e.g., entering a location *without authorization*). For ambiguous or polysemous terms, a specialized domain glossary would be needed. For general or incomplete cues, classification would require a deeper representation of the relationships between the phrase-level metadata, in order to fine-tune the extraction rules for these metadata. This is left for future work.

Usability of the framework.

The main focus of our collaboration with SCL is to provide the necessary toolset and methodology to improve the internal workflows of the Luxembourg public admin-

istration. In any regulated development environment, a team of legal practitioners and IT specialists will collaborate to ensure that (government) IT applications cover all aspects of the regulations. Our approach is not meant to replace this interdisciplinary collaboration but rather to assist these lengthy and complicated workflows by providing the bits and pieces of information to practitioners when they need them, thus enabling a faster and more systematic process. The rest of this section explains the usability of our framework for legal requirements elicitation.

As already noted in Section 4.1, legal statements entail legal requirements for many IT systems. For example, the first statement in Fig. 3.1 is a source of requirements for the IT systems of municipal authorities, the second statement is a source of requirements for the IT systems of vehicle inspectors, and the third statement is a source of requirements for the IT systems managing courts. In this scenario, our framework for legal metadata extraction can help IT departments perform requirements elicitation, not by directly providing legal requirements but rather by providing metadata that can be used within a semi-automated process of legal requirements elicitation. In other words, our framework therefore supports automated semantic legal metadata extraction that supports semi-automated, legal requirements elicitation. IT departments can use the output of our framework to verify the completeness of the requirements documentation and to discover additional requirements. For example, an analysis of the third statement in Fig. 3.1 enables the discovery that "prohibition of driving" is a sanction that must be present in the IT system of the court as a possible consequence of "contraventions of the traffic regulations on any public road". The main advantage over manual requirements elicitation is that the metadata provided by our framework includes implicit knowledge (e.g., the fact that the contravention takes place on a public road) that might be difficult and time-consuming to identify without automated assistance.

Another use case of our framework, within the banking sector, is when banks have to comply with technical regulations. For example, in Luxembourg such regulations are produced by CSSF, the national Luxembourg commission surveying the banking and financial sector. These technical regulations specify the application parameters of the provisions contained in the legislative texts. Our framework enables the banks and financial institutions that are subject to such regulations to have a head start in eliciting their legal requirements from the main legislative act, even before the publication of technical regulations. Furthermore, once this technical regulation is published, our approach (if applied to these different legal documents) would enable the alignment of the requirements elicited from the main legislative act to those introduced by the technical regulation.

4.6 Threats and Limitations

The most pertinent threats to the internal and external validity of our work are discussed below.

Internal validity.

A potential internal validity threat is that the coding in both the qualitative study of Section 4.3 and the case studies of Section 4.5 was done by the researchers. Since traffic laws are intuitive and one of the researchers (i.e., the last researcher) is a legal expert, we found the risk of misinterpretation during coding to be low. Furthermore, to prevent bias in the coding process, we took several mitigating actions: (1) we carefully discussed the difficult cases encountered during coding; (2) we completed the coding component of our qualitative study before defining any extraction rules; (3) we did not apply our implementation to the legal statements in the ground truth until the coding was completed, in order to minimize the influence of the extraction rules on the construction of the ground truth in the first case study; (4) we assessed the reliability of the coding results by measuring interrater agreement over 10% of the coded statements. With regard to the second case study, the ground truth was built by the second researcher in order to limit the influence of the implementation. Again, we evaluated the coding work by having another annotator (the first researcher) code a sample (10%) of the statements, and measuring inter-annotator agreement.

External validity. The nuanced nature of legal texts often requires research on legal requirements to be based upon qualitative results obtained in specific contexts. However, a qualitative study with a scope as limited as ours makes it difficult to address external validity with sufficient rigor. Although we have extended our case studies to new domains with promising results, this extension also showed a difference in the results, which is related to the complexity of the statements. Therefore, further studies that still cover a variety of legal domains and in larger settings remain essential for ascertaining the general applicability of our results.

With this said, the following observations provide a degree of support for the external validity of our qualitative study: First, the rules of Table 4.2 are, in general, simple; there is no particular reason to suspect that these rules may be domain-specific. This helps mitigate the risk of overfitting the rules to our study context. The second case study did not prompt the need for modifying the rules that we already had, and enabled us to add rules and markers for *constraint*, for which we did not have any. This improves our confidence in the meaningfulness of our results across different domains. Second, as we argued while discussing the concept markers of Table 4.3,

most of the marker sets are either systematically extractable from existing lexicons, or expected to saturate quickly due to the limited set of possible linguistic variations. As shown in the second case study, our markers generalized well to other legal domains. However, the study also prompted the need for a more nuanced set of markers, able to handle polysemous and vague terms that can pose a challenge in the interpretation of a statement. We note that the markers are necessarily language-dependent and do not carry over from one language to another.

Data quality and Sufficiency for the ML technique Another threat to the validity of the results of the ML technique used in our work is the quality and sufficiency of the data used as training data for the Random Forest algorithm to predict the category of actors. We used 1000 actor instances for training the ML technique which is sufficient for the domain (i.e., traffic regulations) that we analyzed as we reached saturation in terms of evaluation metrics. In Section 4.5, we reported on the generalizability of the ML model for the second case study for 5 additional legal domains (i.e., commerce, environment, health, labor and penal laws). However, we can not ascertain the generalizability of the ML model to all legal domains as we have only examined the accuracy of the domains that we investigated. We incrementally increased the number of actor instances used to train the ML technique until we reached saturation in terms of standard accuracy metrics. In order to guarantee the quality of the data, we also ensured that the training set for actors had all the desirable characteristics for a training dataset, i.e., accuracy, validity, relevance, and completeness. In particular, we guaranteed completeness by manually verifying that all actors were extracted. We note that we did not feed non-actor entities to the ML technique as negative examples because the classes of actors are already skewed, and adding more non-actor samples would become problematic, as the ML technique would suffer from an imbalanced training set.

4.7 Conclusion

Automatic extraction of metadata about the semantics of legal statements is an important enabler for legal requirements analysis. In this chapter, we derived, through a qualitative study of traffic laws, extraction rules for the reconciled metadata types. Our extraction rules are based on natural language processing, more precisely on constituency and dependency parsing, and are complemented with machine learning for distinguishing subtypes of the metadata type *actor*. We evaluated our extraction rules via two case studies, covering 150 statements from the traffic laws and 200 statements

from five different legislative domains (commerce, environment, health, labour, and penal codes). The results are promising: we obtained a precision of 97.2% and 82.4% and a recall of 94.9% and 92.4% for the first and second case studies respectively. The loss of precision in the second case study is mainly related to the increased length of the statements, which challenges the ability of existing NLP parsers to perform accurately. Still, our results give us confidence in the ability of our rules and markers to achieve good accuracy across different domains.

In the next chapter we will analyze the usefulness and relevance of these semantic legal metadata in a practical use case: a query system to answer the questions that a requirements analyst could ask when elaborating legal requirements.

Chapter 5

Query System for Extracting Requirements-Related Information from Legal Text

Searching legal texts for relevant information is a complex and expensive activity. The search solutions offered by present-day legal portals are targeted primarily at legal professionals. These solutions are not adequate for requirements analysts whose objective is to extract domain knowledge including stakeholders, rights and duties, and business processes that are relevant to legal requirements. Semantic Web technologies now enable smart search capabilities and can be exploited to help requirements analysts in elaborating legal requirements.

In the previous chapter, we developed an automated framework for extracting semantic metadata from legal texts. In this chapter, we investigate the use of our metadata extraction framework as an enabler for smart legal search with a focus on requirements engineering activities. We report on our industrial experience helping the Government of Luxembourg provide an advanced search facility over Luxembourg's Income Tax Law. The experience shows that semantic legal metadata can be successfully exploited for answering requirements engineering-related legal queries. Our results also suggest that our conceptualization of semantic legal metadata can be further improved with new information elements and relations.

5.1 Introduction

Many information systems in domains such as healthcare, finance and taxation have to comply with the various laws and regulations that are pertinent to these domains.

Nowadays, regulations like the General Data Protection Regulation [80] introduce provisions on systems that previously had only sparse regulatory constraints.

As a consequence, when eliciting requirements for such systems, requirements analysts often have to examine the relevant laws in order to identify the software-related concepts and the statements that lead to legal requirements.

Support in searching the law is provided by legal publishers, but only for legal professionals. This kind of support for legal advice is inadequate for requirements analysts, who have different concerns and objectives.

One way to help requirements analysts in their understanding of the law and in the derivation of legal requirements is by enabling them to search the law based on semantic metadata. Examples of such search include looking for (1) the stakeholders of a system, (2) the stakeholders' rights and duties, and (3) the relationships that hold between the stakeholders and the system entities [46].

In chapter 3, we proposed a conceptual model of semantic legal metadata for requirements engineering (RE). Our set of metadata provides information about the statements and the phrases contained in legal provisions. In chapter 4, we further devised an approach to automatically extract our proposed metadata types using natural language processing (NLP).

In this chapter, we organize the semantic legal metadata extracted using our previously developed solution into a knowledge base whose intended purpose is to support a legal query system in the context of requirements elaboration. We provide an implementation of such a query system using Semantic Web technologies. We then utilize our implementation for conducting an industrial feasibility analysis, using Luxembourg's Income Tax Law as a case study. This law is in French; but, throughout this chapter, we use English translations for the excerpts we borrow from the law for exemplification. Finally, we reflect on the lessons learned from our experience.

Our work focuses on the following two research questions:

- **RQ1: Is our existing conceptual model for semantic legal metadata expressive enough to provide an adequate answer to the questions that a requirements analyst may ask when identifying and elaborating legal requirements?**

RQ1 investigates the questions that a requirements analyst may ask, and how she can formulate them in a query system.

- **RQ2: Does our metadata-based query system yield accurate results?** RQ2 is aimed at measuring the accuracy of the results returned by the queries, as well as building insights into how these queries should be posed so as to obtain answers that are as precise and complete as possible.

Contributions. In light of the relevant literature in RE and artificial intelligence and law (AI and Law), we identify five questions related to legal requirements elaboration. We transform these questions into templates of queries for a knowledge base containing semantic legal metadata, and assess the accuracy of the query system based on selected queries.

Our results show that semantic metadata can be successfully leveraged for retrieving high-level information such as definitions, articles and prescriptions. Nevertheless, the results also indicate that there is room for improving the metadata that underlies our query system. In particular, we observe that the metadata should be enhanced with certain additional information in order to enable finer-grained analysis of legal provisions at the level of phrases. We believe that the experience we have gained through our work is a useful stepping stone toward providing computerized assistance in the specification of legal requirements.

Structure. The remainder of the chapter is organized as follows. Section 5.2 discusses background and related work. Section 5.3.1 introduces our legal query toolchain. Sections 5.3.2 and 5.3.3 address RQ1. Section 5.4 addresses RQ2. Section 5.5 discusses threats to validity. Section 5.6 concludes the chapter.

5.2 Background and Related Work

In this section, we review the relevant literature on RE, specially requirements mining, and on AI and Law, specially legal knowledge representation.

5.2.1 Search Systems in RE

Mining requirements. Using NLP and machine learning for identifying and deriving requirements from textual sources of information has received a lot of attention in recent years. Strands of work include requirements gathering from (1) requests for proposals [81], (2) appstore reviews [82], (3) Twitter feeds [83, 84], (4) user manuals [85] and (5) log files [86]. However, being concerned with feature extraction, these

contributions do not target legal analysis or the development of regulated systems, and, more importantly for what concerns this chapter, they are not targeted at querying a knowledge base looking for specific information on a given concern or topic.

Legal requirements analysis. There is considerable research on extracting semantic information from legal provisions with the objective of helping with legal compliance. Breaux and Antón [87] propose an upper-level ontology aimed at classifying statements and their constituents. Maxwell and Antón [46] propose a taxonomy of rights, duties, actors and rules' preconditions for elaborating compliance rules. Massey [47] uses a taxonomy of legal concepts for traceability mapping of requirements to legal texts. Frameworks like Nomos [38], GaiusT [37], NomosT [39] and LegalGRL [42] are aimed at representing legal provisions as goal models. Apart from GaiusT and NomosT, none of these contributions provide tool support for automatically extracting semantic information. In addition, since these threads of work aim at supporting requirements analysts in eliciting legal requirements from specific legal provisions, they do not address the issue of retrieving such provisions in the first place.

Query systems in RE. Query systems in RE are seen as enablers for the analysis of large systems, in particular in the context of traceability management. Mäder and Cleland-Huang [88] propose VTML, a graphical modeling language for visualizing and querying traceability links. Sannier and Baudry [89] propose the INCREMENT tool for the analysis of safety standards and regulations. In this work, standards and regulations can be represented as models, and their content can be searched through a query system based on information retrieval. However, the work only considers structural elements, with a shallow level of provision classification. Pruski et al. [90] propose TiQi, a framework to convert natural languages queries into SQL for querying traceability links. Kanchev et al. [91] propose the Canary approach to query a database of RE-related annotations of online discussions. Canary enables a requester to find discussions related to a given requirement as well as argumentation elements for prioritizing requirements. Again, the granularity level of the metadata is rather shallow as it only considers RE objects (requirement and solution), argumentation objects (support and rebuttal), user information, and the scoring of discussions.

5.2.2 Legal Search and Analysis in AI and Law

Opijnen and Santos [92] identify two types of IT systems in the legal domain: (1) legal expert systems (LES) and (2) systems for legal information retrieval (LIR). While LES rely on Semantic Web technologies (taxonomies, controlled vocabularies,

legal ontologies) to provide a specific answer to a query, LIR is more concerned with retrieving relevant legal documents (or parts thereof) in larger corpora.

Legal expert systems (LES). Examples of LES that rely on semantic metadata are abundant and the large majority of them are based on legal ontologies built using OWL or RDF. For instance, Quaresma and Rodrigues [93] propose a question answering system for the Portuguese criminal law. This approach relies on Prolog and is paired with an ontology supporting the semantic analysis and the pragmatic interpretation of the questions. The approach has nevertheless not been tested on judicial texts but rather on newspaper articles, and the results, although encouraging, are not high quality enough for practical applications. Other examples rely on rule languages such as LegalRuleML [13]: Wyner et al. [94] perform manual annotation on legal texts in order to answer a set of queries concerning the legal semantics of the provisions; Gandon et al. [95] provide an ontological extension of LegalRuleML to support SPARQL queries that go beyond the expressiveness of OWL 2. These systems are nonetheless only presented at the level of proof of concept and are not implemented in a concrete use case.

Legal information retrieval (LIR). In Legal information retrieval, we distinguish between (1) systems based on ontologies [96], and (2) systems using NLP technologies.

Within the first type of systems, the Légilocal system [97] and the Nomothesia platform [98] propose solutions for authorities to manage local regulations implementing national laws in France and Greece, respectively. Their conceptual models cover legal document types as well as structural, geographical and topographical metadata, but do not provide semantic metadata about the content of the provisions.

Within the second type of systems, relevant contributions include Do et al. [99], Adebayo et al. [100] and Collarana et al. [101], all of which aim to retrieve relevant documents. They do not employ a particular conceptual model to formalize the content of the law, and therefore are not able to answer specific queries about content. In Eunomos [102], Boella et al. developed a conceptual model combined with NLP capabilities. Eunomos is a document management system that is mainly focused on vocabulary building; more specifically, it is aimed at reconciling and aligning vocabularies across the European legislation landscape. The conceptual model of Eunomos focuses on domain keywords, domains, and cross-references between articles.

Our approach attempts to bridge the gap between approaches with deep semantic and interpretation capabilities, but almost no tool support, and approaches that provide some support for automatic metadata generation, but lacking means for semantic analysis. In particular, we rely on a domain-independent conceptual model of

semantic legal metadata with automated support for metadata extraction from legal texts.

5.3 Approach

5.3.1 Our Toolchain

In this section, we describe our toolchain for a query system based on legal metadata. This toolchain has been developed in collaboration with Luxembourg's Central Legislative Service (*Service Central de Législation*, hereafter SCL) – the government agency responsible for the publication of all legislative acts in Luxembourg through the online official portal Legilux (http://legilux.public.lu).

The overall workflow of the toolchain is depicted in Fig. 5.1. The first step is to identify the structure of an input legal text and convert the text into a markup document in XML. This step leverages our existing infrastructure for generating structural metadata [103]. The generated markup document includes annotations for provisions at the article level (using Uniform Resource Identifiers - URIs) as well as for cross-references. These structural annotations are essential for providing traceability between the legal text fragments and the legal statements expressed therein. Resources are named using ELI templates [104]. ELI (the European Legislation Identifier) is an EU-endorsed initiative aimed at providing a unified legal referencing mechanism. Its ultimate goal is to facilitate access, exchange and reuse of legal knowledge across the EU member states.

The second step of our approach is semantic metadata extraction. Here, the markup document from the first step is converted into individual statements. Each statement is subsequently processed in order to automatically extract semantic metadata for the statement itself as well as the phrases contained therein. The metadata annotations produced in this step follow the conceptual model developed in chapter 3 and shown in Fig. 5.2. In this chapter, we do not elaborate further on the conceptual model; the reader can find definitions, examples and discussions in our prior work.

The third step is concerned with building a knowledge base that can be queried. Here, for the representation of our metadata, we have chosen RDF (Resource Description Format) – a metadata model and a W3C recommendation since 1999 [105]. Our RDF schema, shown in Fig. 5.3, is a direct implementation of the conceptual model of Fig. 5.2. Fig. 5.4 presents a snippet of the schema and introduces two predicates aimed at building the RDF graph. The first one, *contains* (with its inverse *containedIn*),

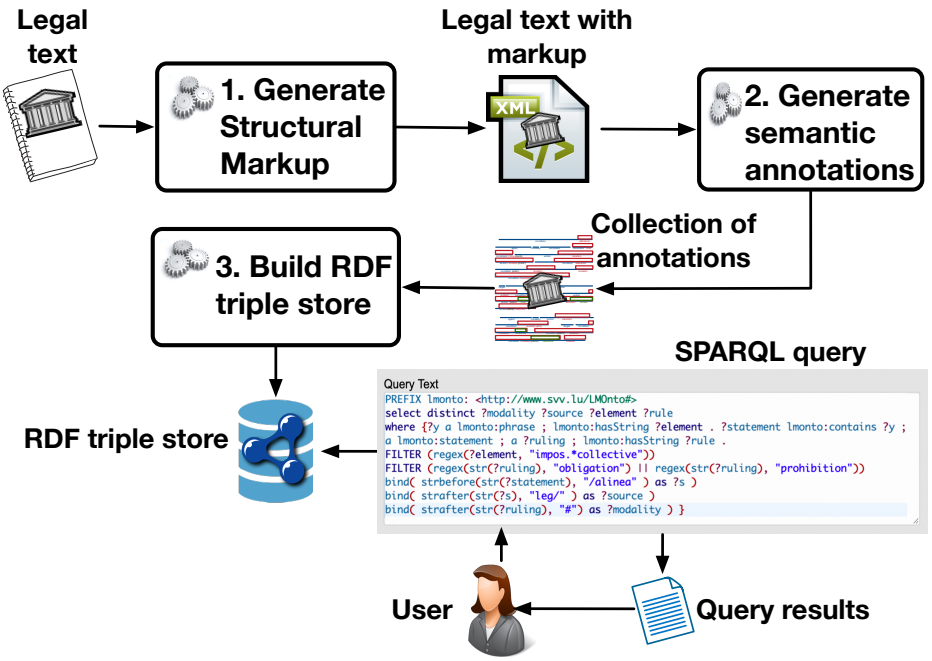

Fig. 5.1 Our Toolchain

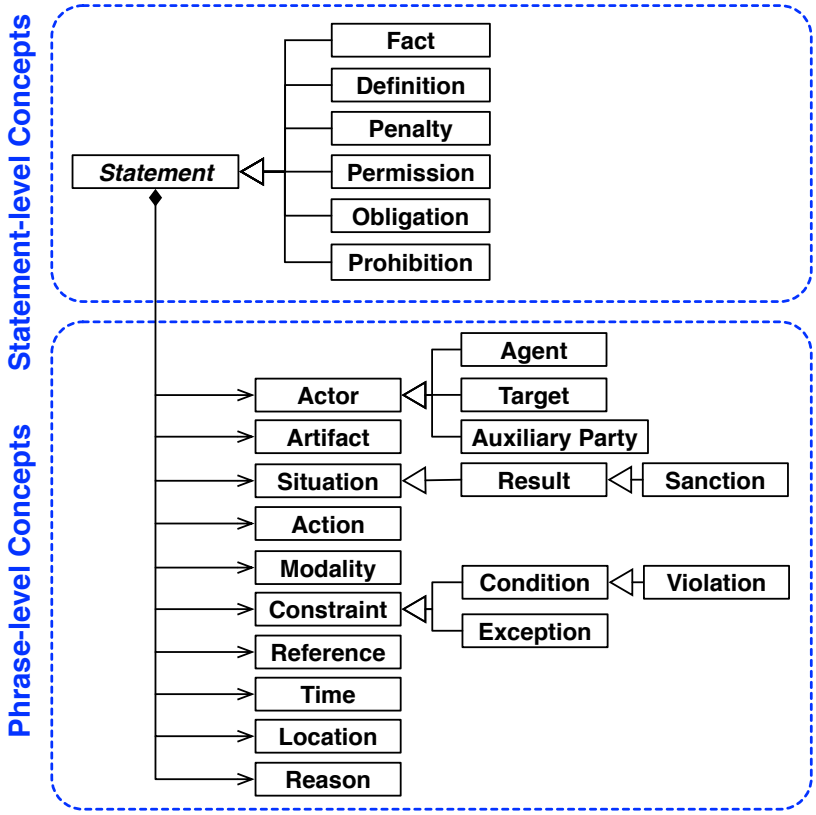

Fig. 5.2 Our Conceptual Model for Semantic Legal Metadata

Fig. 5.3 The RDF Schema in Protégé

links a statement and the phrases enumerated therein, while the second one, *hasSource* (with its inverse *SourceOf*), links a statement and its source, i.e. the article to which the statement belongs.

For querying the RDF triple store, we use SPARQL (SPARQL Protocol and RDF Query Language), the most popular query language for RDF graphs and a W3C recommendation since 2008 [106]. Two factors were decisive in the choice for this technical implementation: (1) it is a convenient and scalable way to handle a large amount of metadata, and (2) SCL has significant experience with these two technologies and is already using them in the Legilux portal.

5.3.2 Most Relevant Questions to Legal RE

Before analyzing the adequacy of our conceptual model for answering RE-related questions, we first need to identify these questions. In this section, we analyze a typical

```
<rdfs:Class rdf:ID="definition">
  <rdfs:subClassOf rdf:resource="#statement"/>
<rdfs:property rdf:ID="contains">
  <rdfs:domain rdf:resource="#statement"/>
  <rdfs:range rdf:resource="#phrase"/>
</rdfs:property>
<rdfs:property rdf:ID="hasSource">
  <rdfs:domain rdf:resource="#statement"/>
  <rdfs:range rdf:resource="#source"/>
</rdfs:property>
```

Fig. 5.4 Excerpt from the RDF Schema

scenario as well as examples from the literature in order to identify a list of high-level questions that could be of interest to a requirements analyst working with legal texts.

RE questions. There is a long history of research regarding the kind of questions that software developers and requirements analysts are likely to ask during RE activities and for software evolution [107–109, 86]. Recently, Malviya et al. [110] have classified relevant questions for RE activities into nine families according to their purposes, among which we deem Business Rule Analysis (family 1), Requirements Elicitation (family 3), Process (family 5), Quality Assessment (family 7), Risk Management (family 8) and Stakeholder Analysis (family 9) to be the most relevant to legal requirements analysis.

We now describe a typical legal requirements elaboration scenario by listing the four essential knowledge extraction activities that an analyst needs to perform when dealing with a domain that is heavily regulated, e.g., taxes, trade, or data protection and privacy.

1. First, the analyst needs to **extract the relevant concepts of the domain** from the underlying legal texts [111].

2. For each relevant domain concept, the analyst then has to extract the applicable authoritative **definition**, in order to align her understanding of the domain with what is envisaged by the law.

3. Once the relevant domain concepts have been identified and defined, the analyst needs to extract the **prescriptions and conditions that apply** to these concepts in order to elaborate legal requirements.

4. Finally, she may be interested in extracting the possible consequences of breaching the law in order to **assess risks** and prioritize requirements [112].

Next, we elaborate each of the activities presented above and identify the practical questions that the analyst may ask in order to extract the required information. We support our choice of questions with examples of similar questions from the literature, with their wording adapted to the tax domain.

Domain concept extraction. **Domain concepts** broadly include the stakeholders of a system, the objects the system handles, and the processes it has to perform or take part in [111].

The elicitation of domain concepts corresponds to multiple questions by Malviya et al. [110], notably those having to do with business rule analysis (family 1, e.g., *"list all business objectives"*) and stakeholder identification (family 9, e.g., *"for a given requirement, who are the stakeholders of interest?"* and *"what kind of users are going to use the system?"*).

- We therefore introduce the first question for our query system (Q1): **What are the relevant concepts of the domain?**

Domain concept definition. Quaresma et al. [93] and Gandon et al. [95] propose questions such as *"what is a taxpayer?"*. These questions are aimed at retrieving **definitions** and indicative statements [113] from which the analyst can derive dictionaries or taxonomies of concepts [36, 87]. Jackson's questions [111], *"what do we mean by 'y is a company'?"*, *"what do we mean by 'z is a kind of commercial profit'?"*, and *"what do we mean by 'y realizes z'?"* go even further in that they effectively attempt to build a domain model.

In addition, identifying the terms that lack an authoritative definition allows answering the questions of Malviya et al. [110] concerning stakeholder identification (see above) and project glossary extraction (family 7, e.g.,*"find all ambiguous words in the requirements"* and *"are there weak words in the document?"*).

- We introduce our second question for our query system (Q2): **What are the definitions for a given domain concept?**

Prescriptions and conditions that apply. After retrieving and classifying all relevant domain concepts, the analyst needs to find in the law all the restrictions and constraints related to these concepts. This means identifying the obligations, permissions and prohibitions (from now on collectively referred to as **prescriptions**) that involve these concepts.

Concrete examples are provided by Collarana et al. [101] (*"what shall a company do with regard to tax obligations?"*) and Wyner et al. [94] (*"what prohibitions apply to*

foreign companies?" and *"what obligations have been placed on which entities, e.g., resident taxpayer?"*). Extracting prescriptions related to specific concepts corresponds to answering the questions of Malviya et al. [110] aimed at requirements elicitation (family 3, e.g., *"which requirements are related to requirement x?"* and *"need to know the regulatory compliance requirements pertinent to process x"*), and at reviewing requirements to uncover errors or inconsistencies (family 7, e.g., *"did I miss any requirements from stakeholders?"*).

- We introduce our third question (Q3): **What are the prescriptions for a given domain concept?**

Interestingly, restrictions and constraints on a domain concept do not only entail prescriptions but also **conditions** and **exceptions** which determine whether that concept is included or excluded from the area of application of a given prescription. For example, the statement *"if the transfer profit [...] includes a capital gain realized on an immovable property, the capital gain may, upon request, be immunized [...]"* includes a clause for "capital gain" ("realized on an immovable property"). This clause does not express the prescription introduced by the statement *"the capital gain may [...] be immunized"*, but rather identifies the subset of capital gains to which the prescription applies.

Capturing the legal conditions and exceptions is linked to the following question in Malviya et al. [110]: *"what type of constraints are embedded in this rule?"* (family 1). Capturing these conditions and exceptions is important not only in order to know the conditions of validity of a prescription in a given context, but also to understand which constraints apply to the business processes that would need to be implemented in the system-to-be. From the analysis of conditions and exceptions, the analyst will be able to extract, among other things, the time or duration constraints related to activities, the input conditions that trigger an activity, and information about the sequencing of different activities. This is exemplified by Robertson & Robertson [114] according to which *"time constraints can be imposed to enable the product to meet a window of opportunity [...] or to satisfy many other scheduling demands"*.

- Observing the important role that conditions and exceptions play during requirements elaboration, we introduce a fourth question (Q4): **Which conditions and exceptions apply to a given domain concept?**

Risk assessment. Malviya et al. [110] identify risk management (family 8) and more particularly compliance analysis as important RE purposes. However, the question

shown as an example of compliance analysis (*"what are the regulations to comply with?"*) is too abstract and does not capture the essence of legal risk [115, 116], which is not merely the identification of the laws to comply with, but also the risk of losses from non-compliance. In the law (and in our conceptual model), **sanctions** identify the concrete consequences of breaching a legal requirement: as such, *sanctions* are the source of legal risks. Wyner et al. [94] ask the general question *"what are all the offenses and associated penalties?"*.

- Following a similar line of reasoning, we introduce the fifth question (Q5): **What are the sanctions for a given breach?**

Our question is more specific than the one in Wyner et al. [94] and Malviya et al. [110] in that it retrieves only the sanctions related to a given offense.

The five questions introduced in this section may not be sufficient to gain a complete understanding of the law from the perspective of a legal expert. Nonetheless and from an RE standpoint, being able to answer these questions is a critical step towards having a systematic and reliable process for the elaboration of legal requirements.

5.3.3 Adequacy of Semantic Metadata for Extracting Requirements-related Information

In this section, we first map onto our conceptual model of Fig. 5.2 the key notions that underlie the questions identified in Section 5.3.2. We then convert the questions into SPARQL queries for automation purposes. By doing so, we assess whether our conceptual model is sufficiently expressive to support the extraction of requirements-related information from legal texts.

5.3.4 Mapping the Questions onto the Existing Metadata Types

The relationship between the questions and the metadata types in our conceptual model is explained below and summarized in Table 5.1.

Q1 (*What are the relevant concepts of the domain?*) aims at characterizing all the relevant domain concepts in a legal text. Jackson [111] identifies domain concepts as stakeholders, objects and processes. In our conceptual model, stakeholders correspond to the phrase-level metadata type **actor** and its subconcepts. Objects correspond to **artifact**. Processes correspond to **situation**. In some circumstances, **location** and **time** may also represent relevant information to elicit. In order to limit the results of

Table 5.1 Mapping between Questions and Metadata Types in our Conceptual Model

Question	Related Metadata Types in our Conceptual Model
Q1. What are the relevant concepts of the domain?	actor, agent, target, auxiliary party, artifact, situation, location, time
Q2. What are the definitions for a given domain concept?	definition
Q3. What are the prescriptions for a given domain concept?	obligation, prohibition, permission
Q4. Which conditions and exceptions apply to a given domain concept?	constraint, condition, exception
Q5. What are the sanctions for a given breach?	penalty, violation, sanction

the query, one can ask more specific questions related to specific metadata types, e.g., "*what are the relevant stakeholders in the domain?*".

Q2 (*What are the definitions for a given domain concept?*) aims at retrieving definition(s) of a domain concept in a given legal text, e.g., the definition of "special expense" in Income Tax Law. In our conceptual model, answering Q2 means retrieving all statements annotated as **definition** and containing the domain concept of interest (e.g., "special expense").

Q3 (*What are the prescriptions for a given domain concept?*) aims at retrieving all the statements that express a legally enforceable order involving (but not necessarily targeting) a domain concept, e.g., a "resident taxpayer". These statements often provide important information for deriving legal requirements. In our conceptual model, answering Q3 means retrieving **obligation**, **permission** and **prohibition** statements containing the domain concept of interest.

Q4 (*Which conditions and exceptions apply to a given domain concept?*) aims at retrieving any text segment that makes a certain domain concept (ir)relevant to the law, thus making the law (in)applicable to that concept. In our conceptual model, answering Q4 means retrieving phrase-level concepts which are typed as **constraint**, **condition** or **exception** and which are related to the domain concept of interest, e.g., "a resident taxpayer who is not married".

Q5 (*What are the sanctions for a given breach?*) retrieves penalties (e.g., "to pay a EUR 12.500 fine") that are associated with a specified breach, e.g., "to forge a certificate". In our conceptual model, answering Q5 means retrieving the **sanctions**

```
1  PREFIX lmonto: <http://www.svv.lu/LMOnto#>
2  SELECT DISTINCT ?concept ?modality ?verbatim ?source
3  WHERE {?x a lmonto:phrase ; lmonto:hasString ?concept .
4  ?req lmonto:contains ?x ; a lmonto:statement ; a ?metadata_type ;
5  lmonto:hasString ?verbatim ; lmonto:hasSource ?s .
6  FILTER (regex(?concept, 'joint taxation'))
7  FILTER (regex(str(?metadata_type), "obligation")
8  || regex(str(?metadata_type), "permission")
9  || regex(str(?metadata_type), "prohibition"))
10 BIND ( strafter(str(?metadata_type), "#") as ?modality )
11 }
```

Fig. 5.5 The SPARQL Query for Q3 on "Joint Taxation"

that are related to a specific **violation** within a **penalty** statement, e.g., "*The one who has forged a certificate [...] shall pay a fine ranging from EUR 251 to EUR 12.500*".

5.3.5 Translating the Questions into SPARQL Queries

We translate the questions identified in Section 5.3.2 into SPARQL query templates. The questions conform to the RDF schema presented in Section 5.3.1. Due to limited space, we do not present all our query templates here. Instead, we discuss only one of the templates, namely that of Q3 from Section 5.3.2. The other templates are similar.

Fig. 5.5 shows the template for Q3, instantiated for the concept of "joint taxation". Our template covers the SELECT, WHERE and BIND parts of the query, while the FILTER part must be specified manually.

Regarding the SELECT part (line 2), we are interested in retrieving the concept to be queried (?concept), the type of prescription that contains it (?modality), the verbatim (i.e. the original text) of the prescription (?verbatim) and its source (?source, i.e., the ELI resource). Regarding the WHERE part and the conditions for triggering a result (lines 3 to 5), we look for phrases that contain the queried concept and for statements of certain type(s) that contain these phrases. The FILTERs (lines 6 and 7) contain the parameters of the query (e.g., the concept of "joint taxation") as well as the metadata types of interest (e.g., *obligation* and/or *permission* and/or *prohibition*), specified manually. The query can be fine-tuned by specifying a selection of metadata types instead of all possible ones, e.g., by selecting only *obligations* in the query filters. The last line (BIND) is a simple post-processing directive aimed at displaying the string of the metadata type (e.g., "obligation") instead of its verbose description in RDF.

Answer to RQ1. We have been able to map onto the conceptual model of Fig. 5.2 all the elements that we have identified to be of interest for posing requirements-related questions over legal texts. By doing so, we provide confidence that our conceptual model is a suitable basis for developing an RE-oriented query system for legal texts.

5.4 Accuracy of the Query System

In this section, we first report on the evaluation of our query system over a case study. We then reflect on our observations and lessons learned.

5.4.1 Case Study Description

Description. The main goal of our case study is to investigate the accuracy of our query system. The study was performed in collaboration with Luxembourg's Central Legislative Service (SCL). SCL already employs a range of Semantic Web technologies for legal text processing, and has considerable prior experience with legal metadata for coordinating and consolidating legal texts. SCL has shown interest in investigating the use of semantic legal metadata for the interactive querying of the law by various interested parties including lay individuals, legal experts, and software and business process analysts.

Data collection procedure. For our case study, SCL proposed to focus on the modified "Law of 4 December 1967 on Income Tax", in short the *Income Tax Law* (ITL). This law is the basis for Luxembourg's taxation system and has implications for the IT systems of the country's national tax administration bodies. The text is 210 pages long and has 241 articles in its 2018 version. On its own, the law does not cover the entire income tax domain as several secondary legislative texts further elaborate on specific aspects of the law. The law nevertheless already provides good coverage of the tax calculation policies that need to be implemented in eGovernment applications [117]. This characteristic makes ITL particularly relevant to requirements analysts. ITL is also reasonably contained in size: we can thus rely on human experts for high-quality manual analysis with reasonable effort.

To process the text of the law, we followed the metadata extraction process described in Fig. 5.1. Overall, we extracted ≈1770 statements, including ≈19000 semantic metadata items. In the process, some phrases were cloned in an attempt to provide self-contained sentences when lists are present, this being standard pre-processing in NLP tasks [62]. In chapter 4, we assessed the accuracy of our semantic metadata

extraction rules, which showed high – yet not perfect – accuracy (overall precision of 87.4% and overall recall of 85.5%). While we do keep track of the NLP errors in our evaluation of the results in Section 5.4.2, we do not reflect on the exact nature and root causes of these errors, noting that our conclusions about NLP are the same as those reached and presented in chapter 4.

For the purposes of our case study, we focus on two topics: (1) taxes related to commercial activities, more precisely, the concept of "commercial profit", and (2) taxation of households, more precisely, the concepts of "indigenous income" and "joint taxation", where the latter is a phenomenon strictly related to the former. These are important topics that affect a large portion of Luxembourg's tax system. They are also addressed in various parts of the law, meaning that an analyst cannot, in normal conditions, scope the search by focusing on a small portion of the law.

Analysis procedure. We perform with respect to our concepts of interest a detailed examination of three questions, Q2, Q3 and Q4, of the five posed in Section 5.3.2. Specifically, when instantiated for the concepts of interest, these three questions will address the following: the definitions of "commercial profit" and "indigenous income" (Q2), the prescriptions that apply to "commercial profit" and "joint taxation" (Q3), and the conditions (and exceptions) that apply to "commercial profit" and "joint taxation" (Q4). Q3 is evaluated at two different levels of granularity: at the article level (Q3.1) and at the metadata level (Q3.2). Q3.1 enables us to examine whether our query system is able to identify the articles containing relevant information, whereas Q3.2 allows us to measure the ability of the query system to provide detailed information by returning relevant verbatim statements from the law. In total, we evaluate eight queries, that is, four queries (Q2, Q3.1, Q3.2 and Q4) for each topic.

Each question was independently analyzed by a different pair of researchers among the first three researchers. All researchers have prior experience in legal informatics, with the second and last researchers being legal experts. For each question, the first three researchers manually investigated the text in order to retrieve the relevant elements together with the location where these elements appear. The retrieved elements were then compared, and discrepancies in the results were discussed among the three researchers to form a ground truth for each question.

In order to build the SPARQL queries, we instantiated with the chosen concepts the templates discussed in Section 5.3.3. We then evaluated the accuracy of these SPARQL queries by comparing their results against the ground truth.

In this study, we are also interested in measuring the effort required for an analyst to manually retrieve relevant information. To this end, we kept track of the time taken

for the construction of the ground truth for each query. This enables us to provide a preliminary indication of the effort that could be saved by using our query system as opposed to a fully manual approach.

Our accuracy analysis is based on the following notions:

- A returned result is *relevant* if it is present in the ground truth. Relevant results count as true positives (TP).
- A returned result is *irrelevant* if it does not appear in the ground truth. Irrelevant results count as false positives (FP).
- A result is *missed* if it is not returned by the query but appears in the ground truth. Missed results count as false negatives (FN).

We measure the accuracy of our query system using the standard precision and recall metrics. Precision is computed as $|\text{TP}|/(|\text{TP}|+|\text{FP}|)$ and recall as $|\text{TP}|/(|\text{TP}|+|\text{FN}|)$.

Finally, we perform an error analysis over the FPs and FNs to identify potential areas for improvements. Specifically, we manually investigate the results in order to assess whether the errors could possibly have arisen from (1) NLP-related issues, (2) our set of extraction rules, (3) shortcomings in our conceptual model, or (4) the query system. In this chapter, we discuss the errors related to only the last three points; as for the NLP-related issues, we refer the reader to chapter 4 where we provide detailed discussions.

We make the following remarks about the two questions, Q1 and Q5, which we do not evaluate in depth here:

- Q1 retrieves a total of 4306 concepts when executed over ITL. A thorough vetting of all these concepts was not possible due to their broad scope. Nevertheless, to ensure the overall quality of the results, the first three researchers collaboratively reviewed a random subset of 430 concepts from the output of Q1 (i.e., 10% of all the retrieved concepts). They deemed 376 of the concepts as being TPs and 54 as being FPs, thus giving a precision of ≈87,4%.

  Naturally, since we did not examine the Q1 results in their entirety, we cannot analyze recall. Nevertheless, there is no reason to suspect issues with recall for Q1, given the promising results from chapter 4 for all the individual metadata types that Q1 retrieves.

- Q5 yielded no result for ITL. Our manual analysis of the law confirmed that the law is not concerned with stating penalties; this function is fulfilled by secondary legal acts.

Table 5.2 Statistics for the Queries in Our Case Study

Query	Ground Truth			Query Results	Relevant (TP)	Irrelevant (FP)	Missed (FN)	Precision	Recall
	Results	Effort							
		R1	R2						
Q2 - What are the definitions for commercial profit?	4	25'	35'	4	3	1	1	75,0%	75,0%
Q2 - What are the definitions for indigenous income?	3	60'	40'	4	3	1	0	75,0%	100,0%
Q3.1 - What articles contain prescriptions for commercial profit?	9	75'	105'	15	8	7	1	53,3%	88,9%
Q3.1 - What articles contain prescriptions for joint taxation?	19	60'	120'	21	19	2	0	90,5%	100,0%
Q3.2 - What are the prescriptions for commercial profit?	11	75'	105'	22	10	12	1	45,5%	90,9%
Q3.2 - What are the prescriptions for joint taxation?	30	60'	120'	33	26	7	4	78,8%	86,7%
Q4 - Which conditions/ exceptions apply to commercial profit?	13	65'	120'	26	4	22	9	15,4%	30,8%
Q4 - Which conditions/ exceptions apply to joint taxation?	23	75'	55'	50	17	33	6	34,0%	73,9%

5.4.2 Results

The results of the evaluation over the eight queries are presented in Table 5.2. Columns 2 to 4 report the size of the ground truth for each query and the approximate evaluation time (rounded up or down to the nearest five minutes) spent by the pair of analysts who manually answered that query. Columns 5 through 8 report the results from the query system and their evaluation. Columns 9 and 10 report the accuracy measures for each query. Although we provide percentages for the precision and recall scores of all the queries, we note that where the results in the ground truth are few, these scores are not good indicators. Below, we discuss the accuracy of each query based on Table 5.2.

Observations on the ground truth. Overall, the analysts needed on average ≈74.7 minutes to analyze the law for each query. While building the ground truth, it emerged that manually identifying the precise text spans in the law took the most time and effort, whereas identifying the information at article level was easier. Another interesting observation was that legal drafting practices can complicate the precise identification of relevant text segments in the provisions. This explains the gap observed between

the two analysts in Q3.1 on "joint taxation" and in Q3.2 on "commercial profit", where one analyst had more difficulty precisely identifying relevant information in the law.

Results from Q2 queries. Regarding the search for *definitions*, our query has only one FN, where the concept of "commercial profit" is conveyed through the general notion of "profit", and this was not accounted for by our query. We elaborate this point in lesson learned L1 in Section 5.4.3.

We also have two FPs, which are due to the presence of the concept in a definition statement that defines another concept. To illustrate, consider the statement *"The following are considered to be indigenous incomes of non-resident taxpayers:[...] commercial profit within the meaning of Articles 14 and 15"*. This statement is a definition of "indigenous income", but not a valid definition of "commercial profit". This raises the issue of identifying the right *subject* of a statement. We elaborate this issue further in Section 5.4.3 (see L4).

Overall, the results show that our query is adequate for retrieving definitions from which the requirements analyst can later derive a dictionary or taxonomy of domain concepts.

Results from Q3 queries. Regarding the retrieval of *prescriptions*, our query system shows good recall scores at the level of both articles (Q3.1, ≈94.5% on average) and statements (Q3.2, 90% on average). The only FN in Q3.1 and Q3.2 for the query on "commercial profit" is, similarly to Q2, related to a prescription for the more general domain concept of "profit".

Regarding the four prescriptions for "joint taxation" that are not retrieved in Q3.2, the error analysis shows that they are due to NLP errors during metadata extraction.

As indicated by Table 5.2, precision varies from 45.5% to 90.5%, depending on the query and its granularity. In particular, precision decreases when we search for information at the statement level, the retrieved results being finer-grained. Regarding the nine FPs in Q3.1, two are related to NLP errors. Five FPs are concerned with other domain concepts as discussed above for Q2. Another two FPs are related to the retrieval of delegation statements, which we elaborate momentarily.

Regarding the 19 FPs in Q3.2, two of them are due to NLP errors. Ten FPs are concerned with other domain concepts. The remaining seven FPs are related to the retrieval of the following statements:

- **Delegation statements**, which give powers to a secondary (legislative or administrative) legal instrument to specify or implement a given provision. An example such statement from the ITL is: *"A Grand-Ducal Regulation may extend*

to the partners taxed jointly the regulatory provisions [...] applicable to the spouses taxable jointly".

- **Party-to-the-law statements**, which express a legal requirement through the extension (or restriction) of the area of application of another legal provision [118]. An example from the ITL is: *"The provisions of Title I of this law are applicable for the determination of the taxable income and the net income of which it is composed [...]."*

Although it is important from a legal perspective, the information contained in delegation and party-to-the-law statements is often only marginally relevant to a requirements analyst since such information does not provide the details of the concrete prescriptions for the domain concept. We discuss the implications of delegation and party-to-the-law statements in Section 5.4.3 (see L3).

Overall, the results show that our query is adequate for retrieving prescriptions related to domain concepts, but in order to increase precision we need finer-grained information in the conceptual model.

Results from Q4 queries. Q4 differs from the previous questions in that it is not aimed at retrieving entire statements or articles but precise phrases (*conditions* or *exceptions*).

The accuracy for Q4 is low: we have a total of 55 (33+22) FPs and 15 (9+6) FNs. The FNs shown in Table 5.2 are explained as follows: (1) four conditions are contained in statements that were already FNs in Q3; (2) another four are due to an erroneous NLP extraction given the complexity of the statements in question; finally, (3) seven conditions use common linguistic patterns for which no extraction rules exist due to a design decision (the resulting extraction rules would have been too generic and we wanted to avoid generating too many FPs).

As for the 55 FPs, 26 FPs come from statements that were already FPs in Q3. In these cases, solving the issues observed for Q3 (i.e., fixing the NLP errors and adding new statement types) would increase the precision for Q4 as well (on average, precision would increase from 21/76 ≈27.6% to 21/50 = 42%). A further 26 FPs are due to the fact that the retrieved conditions are valid conditions but concern a different concept from the one specified in the query. For instance, consider the following statement, which is an FP for Q4 on joint taxation: *"Remuneration paid to a relative other than a spouse who is taxed jointly with the operator is deductible as an operating expense if it is due under a service contract that meets the conditions to be specified by a Grand-Ducal Regulation."* Here, the conditions "if it is due under a service contract" and "the

conditions to be specified" are not related to the concept of joint taxation but to the remuneration and to the service contract, respectively. The remaining three cases are due to NLP errors, resulting in *actions* being incorrectly tagged as *conditions* or *exceptions*.

Overall, the results show that our conceptual model needs to better handle the relationships between metadata in order to answer detailed questions such Q4. We further discuss this point in Section 5.4.3 (see L5).

Answer to RQ2. Our query system can provide accurate results when searching for statement-level and article-level information (Q2 and Q3). Nevertheless, further work needs to be done for successfully answering queries that are aimed at retrieving phrase-level information (Q4).

5.4.3 Observations and Lessons Learned

In this section, we present the observations and lessons learned from our case study.

Observations concerning a domain taxonomy. Q2 and Q3 on "commercial profit" showed the importance of having a domain taxonomy for managing the existing hierarchy of terms and concepts that affect the queries. In the law, the concept of *commercial profit* is defined as a kind of *profit* alongside various other types of profit including *profit from agriculture*, *profit from forestry*, and *profit from independent activity*. These concepts also share subconcepts, such as *divestment profit*. Not knowing these relationships entails the risk of (1) missing general prescriptions on *profit* that span all the subconcepts, (2) missing prescriptions for *divestment profit* related to *commercial profit*, or (3) erroneously accounting for *divestment profits* that are not related to *commercial profit* but to other types of profit, e.g., agricultural profit.

Lesson learned 1 (L1): Having a domain taxonomy or an ontology available would enable easier exploration of the law and make the querying of the RDF graph easier. Building such a taxonomy (or ontology) can be facilitated by Q2 queries.

Observations concerning cross-references. In our queries, some of the returned results contained cross-references. In certain cases, the full content of a definition or prescription could only be retrieved by following those cross-references. Cross-references may contain information that has a direct impact on legal requirements [66, 59]. It is thus important that requirements analysts carefully consider and inspect cross-references during requirements elaboration. To help with this, our structural markup generator (Fig. 5.1) already detects and resolves cross-references.

Automatically navigating and analyzing cross-references can improve the quality of legal query results. However, doing so also raises the question of how far to extend the

analysis: indeed, the targeted provision might in turn contain more cross-references, which should also be resolved and analyzed, with the risk of drifting too far from the initial scope of the analysis. Maxwell et al. [66] and Sannier et al. [67], among others, have taken steps in the direction of (automatically) interpreting cross-references. Despite these interesting contributions, more work is required before cross-references can be handled automatically and sufficiently accurately for questions-answering purposes.

Lesson learned 2 (L2): At this stage, from a practical perspective, it seems preferable to provide the cross-references as additional information and let the analyst decide how to handle them.

During the analysis of our results, we encountered two particular types of cross-references: (1) cross-references that delegate the implementation of a prescription to another legal text, and (2) cross-references that modify the application area of another provision. The presence of such cross-references affects the classification of the statements that contain them, as we elaborate next.

Observations concerning statement types. Statement may delegate the specification or implementation of a prescription to a future legal document, or modify the area of application of a statement. Although such a statement can be understood as an obligation, a permission or a prohibition, it should be considered as a delegation statement or as a party-to-the-law statement. From a legal standpoint, party-to-the-law statements have the effect of a prescription, but the sentence itself does not include the information that would enable the precise identification of the prescription, since this information is located elsewhere (i.e., in the referenced legal provision). Ideally, useful information would come from resolving the corresponding cross-reference [59]. However, performing this analysis and providing the information through the query system would require rethinking both our conceptual model of semantic legal metadata and the extraction rules we have developed for metadata extraction. We therefore leave this to future work.

Nevertheless, it would be useful to identify these statements in order to filter them out when they are deemed irrelevant by the analyst. This identification can be achieved by adding two boolean attributes (*isDelegation* and *isPartyToTheLaw*) to all statement-level metadata types in our conceptual model.

Lesson learned 3 (L3): Adding the notions of *delegation statement* and *party-to-the-law statement* in our conceptual model would offer easier exploration of the law, and provide a filtering mechanism to the analyst. Those statements can be detected

by looking for cross-references within the *subject* of the statement. We elaborate on *subject* next.

Observations concerning the subject of a statement. The notion of *subject* in the literature identifies the addressee or main target of a legal provision [87, 38]. Linguistically, it corresponds most of the time to the semantic subject of the main clause. In our current conceptual model, this notion is addressed through the *agent* metadata type, which, however, can only specify actors. This notion could instead encompass all possible phrase-level concepts that can appear as addressees of the law, i.e., *actors*, *situations*, and *artifacts*. This way, when the actual human addressee is not explicitly mentioned in the statement, labeling as *subject* the addressed artifact or a situation would provide a first clue toward the identification of the real addressee. For example, consider the statement *"Compensation paid to a close relative other than the spouse taxable jointly with the operator is deductible as an operating expense [...]"*. Here, the *subject* is "compensation". However, one correct interpretation of the statement would be *"the taxpayer can deduct compensations paid to a close relative other than the spouse taxable jointly with the operator [...]"*, where the addressee is "the taxpayer paying the compensation".

The addressee may also correspond to a different element than the subject of the main clause, e.g., a *target*, and less commonly, an *auxiliary party*. Consider for instance the following statement: *"It is allowed for operators with regular accounts to include in the net assets invested goods [...]"*. Here, the addressee, namely, "operator", is not the linguistic subject of the sentence. This happens not only with impersonal verbs (i.e., verbs with no determinate subject), as in the example, but also with party-to-the-law statements, discussed above.

Lesson learned 4 (L4): Capturing the *subject* of a statement requires enhancing the conceptual model with a boolean attribute (*isSubject*) added to *actors*, *situations* and *artifacts*, as well as defining and implementing new extraction rules aimed at identifying the correct addressee of the legal provision.

Observations concerning fine-grained analysis. Looking at the results of Q4, we learned that, in order to successfully retrieve all the conditions related to a given domain concept while discarding those that are not, it is necessary to improve the conceptual model with relationships between metadata types. In practice, we need to account for the relationships between *actors*, *artifacts* and *situations* on one side and *constraints* on the other side.

Lesson learned 5 (L5): It seems useful to link *constraints* and their subconcepts, namely *conditions* and *exceptions*, to their related phrase-level concepts in our con-

ceptual model. This requires an extension of the conceptual model as well as new extraction rules.

5.5 Threats to Validity

The validity considerations most pertinent to our work are internal and external validity, as we discuss below.

Internal validity. The first threat to internal validity is related to the risk of misinterpreting (or having changing interpretations of) the provisions in the law when elaborating the ground truth for each query. This risk is minimized by the analysts having background in legal analysis and compliance. Second, while elaborating our queries and criteria for evaluation, we avoided as much as possible restricting alternative legal interpretations, in order to leave the final decision on the interpretation to the analyst. Third, each question was analyzed by a pair of analysts and the results were discussed and reconciled among all the analysts.

Another threat to internal validity is related to the alignment between the questions that we identified in Section 5.3.2 and the SPARQL queries that we built. We note that the questions are simple, and thus there is a limited risk of misinterpretation. There remains, though, the risk that the query does not fully cover the initial question as observed in the results for Q4, due to potential limitations in the conceptual model. If present, such limitations would however also apply to a manual search, which, in the case of Q4, would leave the identification of conditions and exceptions totally in the hands of the analyst.

External validity. The main threat to external validity has to do with the generalizability of our results. Due to the effort-intensive nature of the tasks in our study (e.g., building the ground truth), we evaluated our queries on two topics only ("commercial profit" and "joint taxation"), among the many different topics that would need to be covered in relation to the Income Tax Law. There is a risk that our observations and suggestions for improvements would not readily generalize to other topics. Further studies that cover other legal domains and a more comprehensive list of topics therefore remain essential for validating the general applicability of our results.

A second threat to external validity is related to the size of the corpus. The law over which we posed our queries in this chapter does not cover the entirety of its underlying domain (taxation) as there exists considerable secondary legislation providing implementation and enforcement details. Going for a larger corpus could have an impact, since the number of elements to retrieve and the ones that would actually be

retrieved by a query system will inevitably increase. This gives rise to the risk that the analyst may be overwhelmed by large result sets. This risk is, however, only relevant for very broad queries such as Q1. In such situations, the analyst would still be able to scope the search to a specific context or document and thus obtain result sets of manageable sizes. To the best of our knowledge, such query answering datasets do not yet exist for legal RE. In future work, we plan to build such datasets and investigate to which extent we can leverage the automatically extracted semantic metadata for a real-world question answering system for other legal domains. A possible direction is to use transfer learning techniques (e.g., few-shot learning) with supervised attention mechanism to ensure the generalizability and accuracy of the query system.

5.6 Conclusion

In this chapter, we described an industrial experience aimed at helping requirements analysts to query legal texts. The work is a follow-up to our previous research on automated legal metadata extraction [119]. To build a query system, we convert the extracted metadata into RDF triples and populate a knowledge base using the resulting triples. We identified five important questions that requirements analysts are likely to ask when elaborating legal requirements. We proposed SPARQL query templates corresponding to each question and evaluated the accuracy of the templates through a case study on Luxembourg's Income Tax Law. Finally, we drew several lessons learned to guide future work.

Our analysis suggests that our conceptualization of legal metadata is a useful basis for smart legal search in the context of RE. Further, our empirical results show that we can accurately query for relevant information at the article and sentence level. At the same time, the results pinpoint areas for further improvement. First, we observe that certain drafting practices in legal texts pose challenges for our query system. Second, we identify possible enhancements to our legal metadata information such as an attribute identifying the concrete subject of a statement and additional relationships between metadata types.

In the next chapter, we will investigate whether our existing query system can be augmented with techniques that can automatically derive templates for legal requirements from legal texts.

Chapter 6

Automated Recommendation of Templates for legal requirements

In legal requirements elicitation, requirements analysts need to extract obligations from legal texts. However, legal texts often express obligations only indirectly, for example, by attributing a right to the counterpart. This phenomenon has already been described in the Requirements Engineering (RE) literature [10]. We investigate the use of requirements templates for the systematic elicitation of legal requirements. Our work is motivated by two observations: (1) The existing literature does not provide a harmonized view on the requirements templates that are useful for legal RE; (2) Despite the promising recent advancements in natural language processing (NLP), automated support for legal RE through the suggestion of requirements templates has not been achieved yet. Our objective is to take steps toward addressing these limitations. We review and reconcile the legal requirement templates proposed in RE. Subsequently, we conduct a qualitative study to define NLP rules for template recommendation.

Our contributions consist of (a) a harmonized list of requirements templates pertinent to legal RE, and (b) rules for the automatic recommendation of such templates. We evaluate our rules through a case study on 400 statements from two legal domains. The results indicate a recall and precision of 82,3% and 79,8%, respectively. We show that introducing some limited interaction with the analyst considerably improves accuracy. Specifically, our human-feedback strategy increases recall by 12% and precision by 10,8%, thus yielding an overall recall of 94,3% and overall precision of 90,6%.

6.1 Introduction

The elicitation of requirements for IT systems in regulated domains such as labor and healthcare necessarily includes (a) the identification of the laws and regulations that are applicable to the domain in question, and (b) the extraction, by means of legal interpretation, of the legal requirements entailed by the applicable laws and regulations. Since requirements analysts typically do not have the legal expertise to handle these activities, they usually rely on the advice of legal professionals. This type of collaboration, if done without any automated assistance, is costly and time-consuming. Besides, the communication gap that exists between requirements analysts and legal professionals may result in missed legal requirements or legal requirements that are inaccurate or impossible to implement in IT systems. Providing automated support for directly extracting legal requirements from legal texts is an important step toward addressing these challenges.

Legal provisions often state criteria and rules that lead to legal requirements; however, an individual legal statement may express more than one rule or criterion. In many cases, legal statements affect more than one stakeholder (addressee) and in different ways: attributing to a person an obligation that benefits a counterpart has the automatic effect of attributing to that counterpart a right. So, for example, an obligation for a bank in terms of confidentiality of financial information (bank's viewpoint) entails a corresponding right for the customer of the bank for secure authentication (customer's viewpoint).

Our investigation of multiple legal texts suggests that around one out of every six legal statements expresses multiple legal requirements, one for each applicable stakeholder's viewpoint. However, from a linguistic point of view, a legal provision is normally drafted taking into account one and only one of such viewpoints, in order to avoid redundancy. For example, the obligation of a stakeholder (e.g., a bank as in the previous example) is often expressed only by attributing a right to their counterpart (e.g., the bank's customers). The presence of multiple angles to a legal statement introduces a *viewpoint issue* [120] for legal requirements extraction, namely, the issue of detecting obligations (and legal requirements) even though they are only indirectly expressed by a legal statement.

Requirements analysts are interested in writing *good requirements*, i.e., requirements that are unambiguous, testable, clear, correct, understandable, feasible, independent, atomic, necessary, and implementation-free [121–123]. In good requirements, the required action is expressed in the active voice, and from the viewpoint of the addressee (i.e., the IT system or one of its stakeholders). For IT systems that operate in

regulated domains, solving the viewpoint issue is necessary in order to write good legal requirements. This need has been corroborated by Breaux et al. [10], who deem it necessary to "increase requirements coverage, since obligations derived from rights [...] may be operationalized as requirements."

To help requirements analysts with this viewpoint issue, it is necessary to (a) identify the presence of multiple viewpoints in a legal statement, and (b) suggest a different legal requirement for each of these viewpoints. The best way to represent a plurality of requirements is by using templates [11, 9]. Requirements templates assist the requirements analyst in writing requirements that follow best practices in the RE community.

In chapter 4, we devised an approach for extracting statement- and phrase-level legal metadata at a linguistic level. In this chapter, we utilize the extracted metadata for automatically recommending legal requirements templates, thus assisting requirements analysts with legal requirements elicitation. We rely on the phrase-level metadata types *action*, *target* and *violation*, and the statement-level metadata types *obligation*, *prohibition*, *permission*, and *penalty*.

This chapter is motivated by two observed limitations in the literature on legal requirements elicitation:

1) Lack of a harmonized view of templates for legal requirements. While the RE community acknowledges the importance of requirements templates and systematic legal requirements elicitation, there is no consensus on the templates for legal requirements. Different strands of work propose different templates, but none provide sufficiently complete coverage of legal requirements templates.

2) Lack of automated support for the recommendation of templates for legal requirements. As our previous research suggests [119], NLP techniques have considerably improved in recent years. This raises the prospect that modern NLP techniques may be accurate enough for automated requirements extraction from legal texts. However, to the best of our knowledge, a fully fledged application of NLP has not yet been attempted in legal requirements elicitation.

Research questions. Throughout the chapter, we investigate three Research Questions (RQs). RQ1 tackles the first limitation above, while RQ2 and RQ3 tackle the second.

RQ1: What are the adequate and sufficient templates for legal requirements? RQ1 aims at developing a harmonized set of templates for legal requirements with a sufficient level of expressiveness. To this end, we review and reconcile several

existing proposals of legal requirements templates. Our answer to RQ1 is the first contribution of the chapter: a set of legal requirements templates.

RQ2: Can one define template-recommendation rules over linguistic cues from legal texts? RQ2 investigates the possibility to define rules for template recommendation that rely on linguistic cues from legal texts. We designate as linguistic cues the output of NLP technologies such as constituency parsing, dependency parsing and verb lexicons (e.g., VerbNet), as well as the semantic medatata extracted following our existing approach [119]. In order to define template recommendation rules, we conduct a qualitative study over 1000 randomly selected legal statements from the labor and health domains. Specifically, we annotate the statements with the appropriate templates from the ones identified in RQ1. We use the results of this study for defining rules for automatic template recommendation. The answer to RQ2 is the second contribution of the chapter: a set of NLP-based rules for the recommendation of legal requirements templates identified in RQ1.

RQ3: How accurate is our approach at recommending legal requirements templates? RQ3 aims at evaluating the accuracy of our approach for template recommendation. Our evaluation is based on 400 legal statements randomly selected from both health and labor laws. Our empirical results suggest that our approach has a recall of 82,3% and precision of 79,8%. A follow-on analysis of the recommendation errors reveals that most of the errors can be easily avoided with limited interactive guidance from the analyst. We show that by incorporating into our approach a lightweight human-feedback component, recall and precision increase by 12% and 10,8%, respectively, thus resulting in an overall recall of 94,3% and an overall precision of 90,6%.

Overview and structure. Section 6.2 reviews the related work. Section 6.4 answers RQ1 by describing our harmonization of existing legal requirements templates. Section 6.5 presents our qualitative study and the recommendation rules resulting from it, thus answering RQ2. Section 6.6 answers RQ3 through a case study that evaluates the accuracy of our approach. Section 6.7 discusses threats to validity. Section 6.8 concludes the chapter.

6.2 Background and Related Work

In this section, we review the relevant literature from RE, specially concerning requirements elicitation, and from AI and Law, specially concering legal knowledge representation.

Foundations from legal theory. A systematic account of the relationship between legal positions was first investigated by J. Bentham [124] and further formalized by W. N. Hohfeld. The Hohfeldian system [33] distinguishes eight terms for legal positions: right, privilege, power, immunity, duty, no-right, liability, and disability. Each term in the Hohfeldian system is paired with one opposite and one correlative term. In this work we are interested in correlatives, i.e., legal positions that entail each other. For example, the right of a party entails a correlative duty for the counterparty: an employee has the right to obtain a copy of the payslip, which entails a correlative duty for the employer to provide the employee with such payslip.

Balancing rights and obligations in RE. The RE community has already highlighted the viewpoint issue in legal texts. Darke & Shanks [120] provide a conceptual framework to "increase requirements coverage by integrating 'viewpoints' representing particular perspectives or set of perceptions of the problem domain". Breaux et al. [10] "identify implied rights and obligations [...] to ensure requirements coverage and consider multiple viewpoints". The authors show three ways to balance rights and obligations, dealing with delegations, direct provisions, and indirect provisions. Kiyavitskaya et al. [125] highlight how EU Directives may contain "two-level provisions that impose an obligation on member states and at the same time guarantee a right for an individual person".

Requirements Templates. Considerable work has been devoted to structuring requirements through template suggestion. Palomares et al. [5] report on the use of patterns in RE in a comprehensive survey. First attempts include Robertson's study [6] on "how event/use case modelling can be used to identify, define and access requirements patterns" and Dwyer et al.'s set of templates [7] for the specification of verifiable requirements through state machines. The latter involves manual mapping and transformation of requirements into logical expressions. More recently, Mavin et al. [8] present the Easy Approach to Requirements Syntax (EARS). EARS templates have a high-level perspective, and do not adequately account for actors and stakeholders other than the IT system.

Other contributions are specifically aimed at capturing legal requirements. Breaux & Gordon [9] present a list of generic templates to highlight information within legal

provisions in the Legal Requirement Specification Language (LRSL). LRSL is aimed at encoding legal provisions for developers and policy makers. It accounts for conditions, actions, the syntactic structure of the legal provision, and the different stakeholders of the IT system. In the previously mentioned work, Breaux et al. [10] present a methodology for extracting rights and obligations from regulations using a semantic model. They define a list of patterns for such rights and obligations. Young & Anton [11] present a list of templates for translating provisions into legal requirements. These templates have IT systems as their main focus and take into account the different stakeholders' viewpoints. Yoshida et al. [12] update the templates proposed by Young & Anton by adding templates for definitions and processing data objects as first-class components. Although they present a method for automatically suggesting templates, the implementation has important limitations, the most notable being its exclusive focus on functional requirements, thus not accounting for non-functional and quality requirements.

Contributions from AI and Law focus on representing legal requirements with logical rules rather than templates. LegalRuleML [13] is a rule language that classifies statements into facts and norms, further specialized into constitutive, prescriptive, and penalty statements. LegalRuleML provides a solution to accurately express complex legal rules, but it is not supported by automatic extraction of concepts.

NLP technologies. As noted before, the potential of NLP technologies has increased with recent advancements. *Semantic Role Labeling* [23, 24] is the activity of assigning semantic roles to each of the predicate's arguments in a sentence. These roles usually capture the semantic commonality between instantiations of actors or artifacts across the language. The most notable contribution in the field is FrameNet [25], rooted in the theory of frame semantics. *Deep language analysis* [26] consists of using knowledge of linguistics to extract knowledge from text. It is a type of analysis that takes into account the nuances and complexities of linguistic constructs such as negation and conditionality. A *verb lexicon* is a lexical database of the different variations of syntactic representations of verbs in a sentence. VerbNet [27] is a verb lexicon that incorporates both semantic and syntactic information about verb types following Levin's classification of verbs [28].

Use of Semantic Legal Metadata. In chapter 4, we proposed a conceptual model of semantic legal metadata for RE. The proposed metadata types provide information about the statements and phrases contained in legal provisions. We further developed an approach to automatically extract their proposed metadata types using NLP techniques. We rely on [119] for automatically extracting from legal texts the metadata that form

the cues for our recommendation rules. We do not elaborate further on our harmonized conceptual model and instead refer the reader to [119], where we also discuss the state of the art on legal requirements extraction [126].

6.3 Approach

6.4 Legal Requirements Templates

In this section, we present a synthesis of the different approaches to requirements templates outlined in the previous section, in order to devise a harmonized set of templates to express legal requirements from multiple viewpoints.

Required features. We begin by presenting the features that we need for our legal requirements templates:

- In order to represent multiple viewpoints [10], our templates will express pairs of corresponding statements, each formulated from the viewpoint of a different stakeholder. The first required feature for our templates is therefore **to be able to handle different stakeholders as subject**.
- Legal drafting practices often implicitly refer to a stakeholder by referring to the data objects they are related to. For example, the obligation "A person must write a report that contains [...]" is often expressed in the form "Report must contain [...]". This raises the need for our templates to **handle different data objects as first-class components**.
- Finally, we want to **present the templates in a textual form**, due to the ubiquitous and universal use of natural lanuage in RE, especially in the elicitation and specification of legal requirements which typically involve different stakeholders with different expertise [127, 128].

Table I compares the approaches presented in the previous section against our required features. We note that LRSL supports different stakeholders, and so does Breaux et al.'s patterns. The latter also balances rights and obligations, which is paramount for handling multiple viewpoints. However, LRSL has a graphical representation and Breaux et al.'s patterns have an itemized representation. This does not fit with our required feature of templates being in textual format. Requirements that are not in a textual representation bring with them the need for additional training; for

Table 6.1 Mapping of Approaches to Requirements Templates

Related Work	Support for Different Stakeholders	Support for Different Data Objects	Textual Template	Balancing Viewpoints
Easy Approach To Requirements Syntax [15]	✘	✘	✔	✘
Yoshida et al.'s Functional Requirements Templates [16]	✘	✔	✔	✘
Young and Anton's Templates for Legal Requirements [6]	✘	✔	✔	✘
Legal Requirement Specification Language [7]	✔	✘	✘	✘
Breaux et al.'s patterns [1]	✔	✔	✘	✔

legal requirements, this would include training legal professional who may not be keen to use formal languages.

EARS proposes widely known and used textual templates for requirements. However, it is not suited for our objectives as it does not handle different stakeholders and data objects, depending on viewpoints. To cover these aspects, we adopt Young & Anton's and Yoshida et al.'s templates as a starting point and enhance them with multiple viewpoints.

Statement types. Having compared the approaches from the literature and identified the features of our templates, we proceed to define the type of rules that we want to represent. The reference model covers the Hohfeldian concepts that are relevant to RE, namely duty and right [129]. To do this, we focus on four statement types in our conceptual model [119] that are sources of legal requirements: obligation, prohibition, permission, and penalty.

Although *obligation* and *prohibition* are presented as distinct statement types in our reference conceptual model, we note that a prohibition is just a linguistic construct to express a negative obligation. Simply put, a prohibition requires that a specified action does not take place in the system. Since we are focusing on the semantic content of legal statements, we group those two statement types into a single one: **duty**. Duties are the main source of legal requirements, and the easiest to transform into requirements

when expressed directly by a legal statement. Detecting indirectly expressed duties in statements with multiple viewpoints is the main focus of our study.

Permissions can express two types of legal rules, either a right or a power. *Rights* are a secondary source of legal requirements. From rights are derived obligations, that can subsequently be transformed into legals requirements. We note, however, that not all rights entail obligations. *Powers* attribute to one or more public servants a legal competence or duty. From the point of view of requirements elicitation, they have the same effect as rights, in that they often (but not always) entail an obligation for the liable stakeholders.

With regard to **penalties**, we note that the requirement engineer should extract duties from instances of the phrase-level metadata type *violation*. For example, in the sentence "Anyone inciting acts of hatred against a person is punished by an imprisonment of eight days to two years" the violation item is "inciting acts of hatred against a person" and the corresponding duty (prohibition) is "Individuals are forbidden from inciting acts of hatred against a person".

The remaining statement types in the reference conceptual model, i.e. *facts* and *definitions*, are outside of the scope of this study. These statement types have constitutive effects and do not prescribe behaviors, as explained by Ceci et al. [130]. They can, however, interact with rules that express requirements and therefore affect those requirements. An example is the statement "Article 13 applies to public health workers". Rules with such interactions are called *metarules* in formalizations such as LegalRuleML [55]. The present research does not deal with metarules, since they pose challenges that are far from our main focus here, i.e., detecting multiple viewpoints. Also, our reference conceptual model does not cover metarules extraction. This limitation implies that our requirements might be missing additional stakeholders and conditions that are introduced by the metarules. A possible approach to handle metarules is to follow Breaux [131], which uses state-event tables and transition tables to link "events generated from rights and obligations [...] to pre-conditions of other rights and obligations". Until a solution for the automatic handling of metarules is achieved, it is possible to circumvent this limitation by asking a legal expert to analyze the metarules and manually amend the affected requirements accordingly. Automatic identification of metarules involves detecting and analyzing cross-references [59].

In addition to the three statement types described above (duty, permission, penalty), we classify our templates into two categories depending on whether the action supports a *target*: **intransitive requirements** are those where the action does not support a target, and have the structure "Actor <modality> Action"; **transitive require-**

Table 6.2 Excerpt of Legal Requirements Templates

Template Category	Proposed Template	Example
Statement with no counterpart	Actor <modality> Action	A covered entity shall document a restriction in accordance with §160.530(j) of this subchapter.
	The system <modality> Action	The system shall document a restriction in accordance with §160.530(j) of this subchapter.
Statement with correlative statement	Actor_1 <modality_1> Action_1 to Actor_2	The data subject shall be able to obtain confirmation as to whether or not personal data concerning him or her are being processed from the controller.
	Actor_2 <modality_2> Action_2 to Actor_1	The controller shall provide confirmation as to whether or not personal data concerning him or her are being processed to the data subject.
Statement with implied statement	Actor_1 <modality> Delegation_Action Actor_2	A consumer shall have the right to request to a business that they disclose to that consumer the sale of personal information.
	Actor_2 <modality_2> Action_2 to Actor_1 if R1	A business shall disclose the sale of personal information to the consumer if requested by the consumer.

ments are those where the action supports a target, and have the structure "Actor1 <modality> Action to Actor2". The concept of target of a legal requirement is defined by Young & Anton [11] as "the intended recipient of the actor's action" and is a phrase-level concept in our reference conceptual model.

Legal requirements templates. Based on the above classification, we derive six legal requirement templates. We classify these templates into three categories:

(1) Legal statements with no counterpart. These statements express only one legal requirement, i.e., they carry a single viewpoint. The action of the requirement is directly expressed by the main verb, which does not have a beneficiary – hence the denomination of "legal statements with no counterpart". The templates in this category translate into *intransitive requirements*. This category includes two templates, depending on the classification of the legal statement itself:

a - *Duty with no counterpart*, e.g., "The bank must undergo a standardized accounting exercise each end of year." The legal requirement is "Bank shall undergo a standardized accounting exercise each end of year."

b - *Permission with no counterpart*, e.g., "Personal property that was deposited at the time of bankruptcy may be claimed." The legal requirement is "Depositor

shall be capable of claiming personal property if it was deposited at the time of bankruptcy."

(2) Legal statements with correlative statements. These statements express two correlative legal requirements. Of these two legal requirements, the main one is directly expressed by the verb, and the other is indirectly expressed as the correlative of the main one. An example is the sentence "The user shall obtain a copy of his personal data from the website." This statement reads as a right for the user to obtain a copy of his personal data from the website, and as an obligation for the website to provide a copy of the personal data to the user. The general template for this category corresponds to *direct provision* in Breaux et al.'s work [10], and translates into two instances of *transitive requirement*.

This category includes two templates, depending on the classification of the main legal statement:

c - *Duty with correlative permission*, e.g., "The creditors of the bankrupt are required to file at the district court the declaration of their claims." The corresponding legal requirements are "Creditor shall file the declaration of claims at the district court" and "The district court shall be able to obtain the declaration of claims."

d - *Permission with correlative duty*, e.g., "The user shall obtain a copy of her personal data from the website." The corresponding legal requirements are "The user shall be able to obtain a copy of her personal data from the website" and "The website shall provide a copy of the personal data to the user."

(3) Legal statements with implied statements. These statements express two requirements: a legal requirement directly expressed by the sentence, and another implied legal requirement. An example is the sentence "The Minister delegates to the Police Administration the notification to the driver." This statement reads as a power statement for the authoritative entity (i.e., the Minister) to delegate the notification; this means that each exercise of the power by the Minister implies an obligation for the Police Administration to perform the notification. The general template for this category corresponds to the template for *delegation* in Breaux et al.'s work [10], and translates into two templates: a *transitive requirement* for the requirement directly expressed by the text, and another that is either *transitive* or *intransitive* depending on the implied action. It is also important to notice that the implied requirement is pre-conditioned on the invocation of the original delegation: in the words of Breaux

[131], "a stakeholder must first be delegated a right before they can invoke that right." Considering that, in legal requirements elicitation, we focus on obligations, we can rephrase that into "a stakeholder must first be delegated an obligation before they are subject to that obligation."

In this third category, we have two templates depending on the classification of the main legal statement:

e - *Permission with implied duty*, e.g., "The bankrupt may have the circumstances reported by the district court." The corresponding legal requirements are "The bankrupt shall be able to request that the district court report the circumstances" and "The district court shall report the circumstances if requested by the bankrupt."

f - *Penalty with implied duty*, e.g., "Anyone who incites acts of hatred against a person is punished by an imprisonment of eight days to two years." The corresponding legal requirements are "The court shall punish with an imprisonment of eight days to two years anyone who incites acts of hatred against a person" and "Individuals are forbidden from inciting acts of hatred against a person."

An excerpt of our set of legal requirements templates is presented in Table 6.2, and the complete set is available in an online appendix[1]. The set of templates presented in this section provides an answer to RQ1. Using these templates, it is possible to capture legal requirements expressed both directly and indirectly in legal statements, with the exclusion of metarules for which further research is necessary.

6.5 Recommending Templates for Legal Requirements

In this section, we report on a qualitative study aimed at deriving rules for the automatic recommendation of the legal requirements templates presented in the previous section.

6.5.1 Study context and data selection

Our qualitative study is based on 1000 statements randomly selected from the labor and health laws of Luxembourg (500 statements from each law).

[1] http://shorturl.at/hxzKL

Table 6.3 Rules for Requirements Template Recommendation

Recommendation Rule	Example
IF ("correlative verb" == False AND statement type = type1) Then type1 with no counterpart	A covered entity shall document a restriction in accordance with §160.530(j) of this subchapter.
IF ("correlative verb" == TRUE AND "implied trigger" == False AND statement type = type2) Then type2 with correlative	The data subject shall be able to obtain confirmation as to whether or not personal data concerning him or her are being processed from the controller.
IF ("correlative verb" == TRUE AND "implied trigger" == TRUE AND statement type = type2) Then type2 with implied	A consumer shall have the right to request to a business that they disclose to that consumer the sale of personal information.

The choice of the labor and health laws was motivated by three factors. First, due to these domains being widely known, legal experts found them to be good showcases for automated legal requirements recommendation. Second, the provisions in the labor and health laws are interesting from an RE perspective, due to their broad implications for the IT systems used by employers, courts and public offices such as the tax department, healthcare institutions and insurance companies. Third, our preliminary study on 200 statements from five different legal domains highlighted labor and health laws as the domains where the viewpoint issue is more common (about 20% of legal statements in the labor domain and 15% in health domain carry multiple viewpoints).

As it is the case with most legal texts, the source texts in our study contain statements with enumerations and lists embedded in them. To treat these statements properly, we took the common legal text pre-processing measure of merging the beginning of a statement with its individual list items to form complete, independent sentences.

A legal expert (second researcher) annotated the 1000 statements with the applicable template category from the ones presented in Section 6.4. We note that the legal expert decided on matters of legal interpretation within the scope of the annotation guidelines detailed above. We discuss the foreseen validity threats in Section 3.4.

6.5.2 Rules for Legal Requirements Templates Recommendation

Table 6.3 presents the template recommendation rules that we derived by analyzing the 1000 statements in our study. To maximize accuracy over these templates, we did five iterations over the 1000 statements, progressively refining the rules. For the first iteration, we built rules for a batch of 200 statements. From the second iteration, we evaluated against a new batch of statements and refined the rules until saturation of the evaluation metrics over all the batches.

The element highlighted in blue in each rule of Table 6.3 is the marker, i.e., the target of that rule.

The first step in our approach for recommending legal requirements templates is to use the statement-level semantic metadata we developed in Sleimi et al. [119] to classify the legal provision as expressed in the legal text. As noted in Section 6.4, our conceptual model has six different statement types: fact, definition, penalty, permission, obligation, prohibition. As noted before, we discard statements classified as *fact* or *definition*, as they do not express requirements. Each template recommendation has two parameters: a statement type and a template category classification. The statement type assigned to the legal statement by the metadata extraction module is used to restrict the choice of templates. For example, if the legal statement as expressed by the text is classified as *obligation*, the possible templates will be restricted to *duty with no counterpart* and *duty with correlative right*.

The second step consists of extracting and processing the main verb. We extract the main verb using the following Tregex patterns:[2]

- SENT <(VN=mark)
- SENT <(VPinf <(VN <(VPP=mark)))
- SENT <(PP <(VPinf <(VN <(VPP=mark))))
- SENT <(VPinf <(VN <(VINF=mark)))
- SENT <(PP <(VPinf <(VN <(VINF=mark))))

Note that the keyword *mark* in the rules leads to extracting the verb that forms the linguistic root of the action. We illustrate some of our rules in Table 6.3, in order

[2]VPinf means an infinitive clause. VPP means a nonfinite clause. VN means a verbal nucleus. Vinf means an infinitve verb. PP means a prepositional phrase. For details about Tregex, we refer the reader to [132].

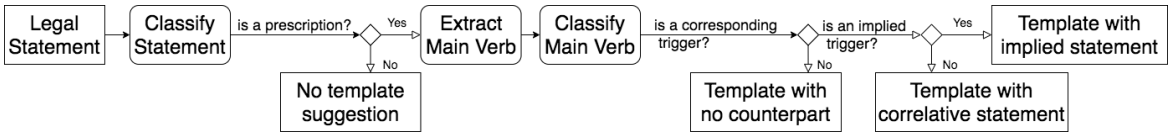

Fig. 6.1 Overview of Our Approach for Requirements Template Recommendation

to facilitate understanding and to discuss some important technicalities of the rules. Having identified the main verb, the next step consists of fetching the possible frames for that verb, using VerbNet. Our qualitative study indicates in fact that the main verb is the most reliable element from which we can infer the presence of multiple viewpoints applicable to the statement (also roles can signal such viewpoints, but they are often left implicit in legal statements). More specifically, the presence of multiple viewpoints is signaled by the main verb of the sentence being a *corresponding verb*. Corresponding verbs are two verbs that express two viewpoints on the same situation. They share the same roles, but with their orders switched. Let us consider, for example, the corresponding verbs *to send* and *to receive*: the subject of *send* is the indirect object of *receive*, and vice versa. Our definition of corresponding verbs is in line with the definition provided by Breaux et al. [10] for *direct provisions*, i.e., "provisions which have a binary opposite where the subject of the activity assumes the value of the co-requisite attribute". We classify the main verb as a *corresponding verb* if VerbNet includes, within the possible frames for said verb, one of the following roles: recipient, patient, experiencer, or theme (as an animate object). The design heuristics for this choice are described below. If a corresponding verb is detected, we know that the statement is not a *statement with no counterpart* (the first group of templates presented in Section 6.4). We then proceed to verify whether the main verb, already marked as corresponding verb, is also an *implied trigger*. We define an *implied trigger* as a verb expressing an interaction where an agent assigns a right or obligation to another person, in line with the definition of *delegation* provided by Breaux et al. [10]. In order to detect the presence of an implied trigger, we look for the verb in our curated list of delegation verbs[3].

Fig. 6.1 presents a summary of the approach. To illustrate the rules, we now describe how they apply to the example statements in the second column of the table.

Statement 1 reads as: "A covered entity shall document a restriction in accordance with §160.530(j) of this subchapter." The statement is classified as obligation, and the main verb (the *mark* keyword in the tregex pattern previously described) is "to

[3]http://shorturl.at/wEI57

document", that is not a correlative verb. As a consequence, the template *duty with no counterpart* is recommended.

Statement 2 reads as: "The data subject shall have the right to obtain confirmation as to whether or not personal data concerning him or her are being processed from the controller." The statement is classified as permission, and the main verb, "to obtain", is a correlative verb but not an implied trigger. The template *permission with correlative duty* is recommended.

Statement 3 reads as: "A consumer shall have the right to request that a business disclose to that consumer the sale of personal information." The statement is classified as permission, and the main verb, "to request", is both a correlative verb and an implied trigger. Therefore the template *permission with implied duty* is recommended. The recommented template for the implied statement is that of *transitive requirement*.

Design heuristics. During our study, we reviewed the list of corresponding verbs for our dataset in light of legislative drafting practices. We note that the roles *experiencer* and *patient* on their own do not constitute cues of a corresponding verb, especially in the absence of the role *agent*. In other cases, although the frame includes the role *theme as an animate object*, the verb does not express corresponding statements. As a result, we excluded 83 verbs and two phrasal verbs which met the aforementioned criteria. In addition, we implemented two heuristics resolving two issues related to legal drafting practices that we encountered during the qualitative study:

- Several statements in the qualitative study have main verbs that on their own do not have a correlative (e.g., "to keep"). However, in these statements, these verbs are part of compound expressions that prompt a correlative statement (e.g., "to keep confidential"). We also made the same observation with nominalizations; our first heuristic is related to the presence of these compound expressions or nominalizations: After we extract the main verb, we validate whether one such expression is present in the statement. If that is the case, we consider the complete compound expression instead of the main verb only. This allows us to correctly recommend a correlative template based on the expression that contains the verb, rather than on the verb alone.

- In some cases, the statements have a main verb that would under normal circumstances prompt a correlative statement. Take, for example, the verb "to take" which has a correlative "to give". However, this verb can be part of a phrasal verb that would indicate a statement with no counterpart, for example, "to take effect". Here, we cannot elicit any legal requirement about a correlative action.

The second heuristic is related to the presence of correlative verbs that can be part of phrasal verbs. After we extract these correlative verbs, we do not immediately recommend a template "with correlative statement". Instead, we consider the presence of these phrasal verbs to prevent incorrect recommendations.

The rules and design heuristics presented in this section enable the automatic recommendation of legal requirement templates, thus providing an answer to RQ2.

6.6 Empirical Evaluation

In this section, we describe our implementation and measure the accuracy of our approach through a case study.

6.6.1 Implementation

Our template recommendation rules are implemented using Tregex and Java. The rules utilize the outputs of the classic NLP pipeline for syntactic analysis. We also use our framework [119] for semantic metadata extraction.

6.6.2 Accuracy of the Template Recommendation

Case study description. The objective of our case study is to measure the accuracy of the template recommendation rules of Table 6.3 against a ground truth. To build the ground truth, a legal expert manually classified 400 randomly selected legal statements from the labor and health laws, in addition to the 1000 statements of our qualitative study (see Section 6.5). The construction of the ground truth took place after the conclusion of our qualitative study. As explained in Section 6.4, we excluded legal statements that express metarules. In order to have a cohesive dataset for the ground truth, we also excluded statements that have contractual effects, i.e., statements for which the interpretation of contract law is necessary in order to be able to identify correlative statements. An example is the statement "The employer may terminate the contract after thirty days". Here, the action "to terminate" by itself would have no consequence for the employer. However, depending on the contract that is terminated, this could lead for example to (a) a severance package (i.e., a correlative right for the employee) or (b) a temporary prohibition for the former employee to approach the clients of the employer if there is a non-compete clause.

Table 6.4 Statistics for Template Recommendations

Corpus	Template Category	Ground Truth Results	Correct (TP)	Misclassified (FP)	Missed (FN)	Precision	Recall
Labor	Statement with no counterpart	96	80	11	16	87,90%	83,30%
	Statement with correlative statement	47	41	15	6	73,20%	87,20%
	Statement with implied statement	13	9	1	4	90,00%	69,20%
	Subtotal	156	130	27	26	82,80%	83,30%
Health	Statement with no counterpart	107	84	14	23	85,70%	78,50%
	Statement with correlative statement	37	32	22	5	59,30%	86,50%
	Statement with implied statement	16	14	3	2	82,40%	87,50%
	Subtotal	160	130	39	30	76,90%	81,30%
Total		316	260	66	56	79,80%	82,30%

Our analysis of the results did not lead to new template recommendation rules.

Analysis procedure. Each template recommendation has two parameters: a statement type and a template-category classification. The first is assigned following the statement-level metadata in our conceptual model [119]; because it is not a contribution of this work, we do not evaluate it. The second parameter is assigned by the approach described in Section 6.5. We evaluate the second parameter, i.e., the automated template recommendation rules, using the following notions:

- A recommended template is a *match* if it has the same template category as the ground-truth classification.
- A recommended template is *misclassified* if it has a different template category than the ground-truth classification.
- A ground-truth statement for which the methodology did not provide a match is considered as *missed*.

Our evaluation results are presented in columns 4 through 8 of Table 6.4. For each category of templates, we provide the number of correct matches, misclassified and missed template-rule recommendations, and scores for precision and recall.

Table 6.5 Error Analysis for Template Recommendations

Template Category	Result Type	Nominalization	Phrasal Verb	Legal Terminology	Error in Metadata	Design Heuristics
Statement with no counterpart	Misclassified	11	8	0	1	5
	Missed	7	11	8	8	5
Statement with correlative statement	Misclassified	5	15	6	7	4
	Missed	6	2	0	0	3
Statement with implied statement	Misclassified	1	0	2	0	1
	Missed	1	0	4	1	0

Each match counts as a true positive (TP). Each misclassified recommendation counts as a false positive (FP), and each missed recommendation counts as a false negative (FN).

Precision is computed as $|\text{TP}|/(|\text{TP}| + |\text{FP}|)$ and recall as $|\text{TP}|/(|\text{TP}| + |\text{FN}|)$. The final row in the table shows the overall results. Note that the overall precision and recall scores are computed over all the recommended templates across both domains, and are not the averages of the precision and recall scores for the individual template categories.

Results. We first discuss the results for the recommended templates for *statements with no counterpart*. Second, we present the results for the *statements with correlative statement* and the *statements with implied statement*. Third, we discuss the discrepancies between the results in the two legal domains considered in our case study. Finally, we perform an error analysis on the misclassifications and missed template recommendations.

Results for statements with no counterpart. Out of 189 recommended templates annotated as *with no counterpart*, 164 were correct matches and 25 were misclassifications. 39 statements with no counterpart in the ground truth were missed. The error analysis is presented in Table 6.5. Our error analysis (summary) results are presented in columns 3 through 7 of Table 6.5. For each category of templates, we provide the number of inaccuracies leading to errors, for each situation. These situations are formally introduced and discussed at the end of this subsection. We obtain an overall precision of 86,7% and an overall recall of 80,7%.

Results for statements with correlative statement. Out of 110 recommended templates annotated as *with correlative statement*, 73 were correct matches and 37 were misclassifications. 11 statements with correlative statement in the ground truth were missed. The error analysis is presented in Table 6.5.

We obtain an overall precision of 66,3% and an overall recall of 86,9%.

Results for statements with implied statement. Out of 27 recommended templates annotated as *with implied statement*, 23 were correct matches and 4 were misclassifications. 6 statements with implied statement in the ground truth were missed. The error analysis is presented in Table 6.5.

We obtain an overall precision of 85,1% and an overall recall of 79,3%. We note that we did not have enough statements from this template category in the dataset (7,25%) to draw meaningful conclusions on the accuracy of our rules.

Legal Domains. Regarding the two legal domains, our recommendation rules performed well in both cases, but slightly better over the labor law: we obtained a precision of 82,8% and a recall of 83,3%, while for the health law the precision was 76,9% and the recall 81,3%. The difference is due to the fact that, while for the labor law we could easily classify and exclude contractual obligations, for the health law there was no clear way of excluding rules that require external knowledge. The error analysis, however, confirmed that the types of errors are equally distributed across the two domains of our study.

Answering RQ3. We can now provide an initial answer to RQ3 based on our quantitative results: our recommendation rules achieve good accuracy with a recall of 82,3% and a precision of 79,8% over the two examined domains in our case study. Despite being good, these accuracy results are still far from perfect. Therefore, based on the quantitative results alone, our study suggests that analysts will need to carefully validate the recommended templates and discard the incorrect ones. Nonetheless, as we are going to argue next, these quantitative results per se are not reflective of the true usefulness of our approach, as they can be considerably improved by introducing a human-feedback component.

Error analysis. To identify the root causes for automation inaccuracies, we analyzed the misclassified and missed template recommendations. As indicated in Table 6.5, the inaccuracies stem from five different situations: (a) Nominalizations, (b) Phrasal verbs, (c) Legal terminology, (d) Errors in automated metadata extraction and (e) Design decisions and heuristics.

In *nominalizations* (deverbal nouns [133]), as noted by Breaux et al. [10], the main action (e.g., to investigate) is nominalized in a form (e.g., investigation) that we did not encounter during our design heuristics process (see Section 6.5).

In the case of *phrasal verbs* such as “to take effect”, we populated a list to be used by the heuristics; however, this list turned out to be incomplete during our case study, though we would expect it to become increasingly more complete as we cover more domains. Besides, in some cases, we could not devise heuristics because the errors

stemming from the new design decision outweighed the correct recommendations. A possible approach to solve the issue would be to paraphrase these constructs, but automating such paraphrasing is outside the scope of this work.

In the case of verbs that are part of *legal terminology*, the issue is that the semantics of the verb when used in a legal statement can be different from the semantics of the verb in its general lexicon. For example, the verb "to suspend" supports a corresponding statement in its general meaning ("the employer cannot suspend the payment of the salary in ordinary circumstances"). However, when used in its legal meaning ("This regulation is suspended until December 31st, 2010") it does not express any legal requirement. We made an effort to build a list for this terminology during our design heuristics process, but it turned out to be incomplete.

Regarding the *errors in automated metadata extraction*, which originate from our metadata extraction framework [119], a validation of the necessary metadata by a human annotator might reveal these errors. Nevertheless, and until a more accurate solution for automated metadata extraction is available, we consider this semi-automatic approach the best trade-off between the human effort required and the accuracy of the results. Finally, we note that the errors stemming from our own *design decisions and heuristics* are outweighed by the correct recommendations, which was the reason for our adoption of these heuristics in the first place during the qualitative study.

6.6.3 Observations and Lessons Learned

In this section, we present the observations and lessons learned from our case study.

Integrating Human feedback. Based on the error analysis presented above, our approach can be modified to enable smart and minimal interactions with the analyst to prevent most of the errors in template recommendations. Such a semi-automated process would prompt the intervention of the analyst for tasks that are relatively simple, such as reformulating a nominalized verb or disambiguating between the general and legal use of a verb. In an interactive mode of use, the analyst's intervention would occur as a pre-processing step in order to assist our approach (Fig. 6.1) with correctly extracting actions from statements. Take for example the statement "The investigation may be performed". Here, the template recommendation approach would prompt the analyst to answer the following simple yes/no question: "The nominalization *investigation* is the main action of the statement. Is this correct?" With this human-in-the-loop component, we can identify the majority of the legal requirements (38 out of 56) currently missed by our fully automated approach.

The observations presented above clearly show how the presented semi-automated approach would provide much higher recall and precision. Specifically, adding this human-feedback component would increase recall by 12% and precision by 10,8%, thus yielding an overall recall of 94,3% and an overall precision of 90,6%. To conclude, our quantitative and qualitative analyses clearly suggest that a semi-automated, interactive approach is a better option for legal template recommendation.

Action in legal statements. We highlighted in Section 6.5 how identifying the main action expressed by a legal statement is key to detecting the presence of multiple viewpoints. The main action is in fact more important than the statement type for identifying the presence of multiple viewpoints. We also note that detecting the type of action from the main verb contained in the legal statement is especially difficult in four situations: (a) nominalizations, (b) phrasal verbs, (c) legal terminology, and (d) implicit roles (subjects or counterparties).

The first three issues were described in the previous subsection. The issues of nominalizations and phrasal verbs are not specific to the legal domain. A possible solution could be (a) the identification of these linguistic constructs, and (b) their resolution through lemmatization (for nominalizations) and the use of a locution thesaurus[4] (for phrasal verbs). Regarding verbs that employ legal terminology, semantic-role labeling should in theory be able to tackle different linguistic forms to express actions. Unfortunately, the most robust tools such as FrameNet are not trained for the legal language, and thus not very effective in the context of our work.

Roles that are left implicit in the text are either referenced by anaphora or totally omitted. A common drafting technique is the conjugation of the main verb in the passive form, omitting the agent. For this reason, during the elaboration of our rules for template recommendation (see Section 6.5), we could not rely on the presence of roles within the legal sentence. For the same reason, we cannot rely on the presence of roles for deriving heuristics that address the first three issues.

Solving the issue of implicit roles therefore seems to be a priority, because of the potential that roles carry for creating new recommendation rules. Solving this issue requires anaphora resolution for implicit roles and a domain model for omitted roles.

The importance of a domain model. As noted in the previous subsection, some misclassifications were caused by the fact that the actions contained in the legal text only expressed multiple perspectives when seen in the light of the applicable legal framework. For this reason, we had to exclude contractual obligations from our study.

[4]A locution is a sequence of words (a phrase) in the sentence that has the same grammatical (semantic) value of a single word. A locution thesaurus is therefore a resource that groups locutions with their corresponding words according to semantic similarity.

In order to overcome this limitation, it is necessary to extract knowledge from additional sources, e.g., contract law for contractual obligations. As noted also in the previous observation on omitted roles, this could be achieved by relying on domain models.

The importance of corresponding statements in extracting requirements from legislation. Our preliminary study on five different legal domains highlighted that the relevance of the issue is domain dependent: while about one out of four statements in health and labor laws express multiple viewpoints, other domains have a much lower ratio. This is only marginally due to drafting techniques, and rather depends on the type of legal relations that are predominant in the domain. For example, health and labor are domains that are interested by many constitutional guarantees and therefore the laws in these domains often attribute rights to subjects. On the hand, the commerce and environment domains are more focused on ensuring due diligence by the operators and the requirements are therefore often expressed directly as duties.

We notice nevertheless that the criteria to detect multiple viewpoints, that we embedded in our recommendation rules, work well across domains.

6.7 Threats to Validity

Internal validity. A potential threat to internal validity is related to the subjectivity of legal analysis and how it affects the elicitation of legal requirements. Oftentimes, complex requirements specifications are organized hierarchically with the requirements expressed at multiple levels of granularity. The same principle applies to legal requirements, but we are not aware of any systematic means for defining granularity levels for legal requirements. To mitigate subjectivity about granularity levels, we restrict our work to the granularity level at which the underlying legal text(s) have been articulated. Furthermore, we note that the coding in both the qualitative study of Section 6.5 and the ground truth of Section 4.5 was done by the second researcher. To mitigate against potential subjectivity caused by the involvement of a researcher in coding, we set clear, upfront criteria for the analysis of the legal statements, and therefore for the ground-truth construction. First, annotations were limited to legal statements that explicitly led to legal requirements for IT systems and/or their stakeholders. Second, we completed the coding component of our qualitative study before defining any recommendation rules. Third, we left out statements that involved the interpretation of external sources, e.g., metarules and contractual obligations. Finally, we did not apply

our implementation to the legal statements in the ground truth until the coding was completed.

External validity. The first threat to external validity is related to the generalizability of our results; for this, we refer the reader to our observations on legal domains in Section 4.5.

The second threat to external validity is related to the differences between languages. Our corpus of legal texts is in French. Our current tool support would thus not readily work for other languages and needs to be adapted in terms of both linguistic cues and heuristics. In the particular case of the English language, one has access to highly developed NLP frameworks that have been trained on very large corpora; this is likely to increase the accuracy of requirements template recommendations, but verifying this claim requires separate empirical investigations.

The third threat to external validity is related to the other ways in which a legal statement can express more than one legal rule. Our goal in this chapter was to extract from a given legal statement all the legal requirements that are the result of multiple viewpoints. However, this is not the only case of one-to-many relations between legal statements and legal rules.

There can in fact be multiple possible interpretations for a legal statement. A multitude of interpretations is different from a multitude of viewpoints in that interpretations are alternatives, i.e., they cannot be valid in the same legal context at the same time. Detecting, extracting, and comparing alternative legal interpretations are topics for research in AI and Law and outside the scope of our current work.

It is also possible that multiple legal statements (e.g., a duty and a definition) are contained within a single legal sentence. An example is "The controller is forbidden from storing sensitive data, which means data that holds sensitive information". These statements are expressed in different parts of the sentence, and the sentence could be split into two different sentences, one per rule, without altering its meaning and without redundancy except for the phrase "sensitive data". This process is, however, not always straightforward, and it can be argued that establishing the granularity of rules in a legal provision is a matter of legal interpretation. Extracting multiple rules merged into a single legal statement remains outside the scope of the present research.

6.8 Conclusion

In this chapter, we presented an approach to automatically recommend templates for legal requirements based on legal statements, thus assisting requirements analysts with

legal requirements elicitation. To do so, we first defined a set of templates that account for multiple viewpoints. These templates are grouped into three categories: statements with no counterpart, statements with a correlative statement, and statements with an implied statement. We then devised, using Natural Language Processing, automated rules for recommending suitable requirements templates. We evaluated our approach on 400 statements from labor and health laws in Luxembourg. Our results show good accuracy with a recall of 82,3% and a precision of 79,8%.

We further collected and synthesized knowledge about the verb constructs that were the cause of incorrect recommendations. We outlined how such knowledge can be leveraged for developing a semi-automated, human-in-the-loop approach that can much more accurately identify suitable requirements templates based on minimal input from legal experts.

Chapter 7

Conclusion

This chapter summarizes the research contributions of this dissertation and discusses the potential areas for future work.

7.1 Summary

In this dissertation, we investigated the feasibility of automated legal text processing. Our solutions build a streamline approach around legal requirements and semantic legal metadata. Such legal requirements are paramount to the compliance of IT systems. We anticipate that our contributions would be largely applicable to different legal domains and jurisdictions. We have empirically evaluated all our solutions using selected case studies in collaboration with a government entity partner. In addition, we identified several lessons learned through our experience and where possible we proposed mitigation workarounds and alternative techniques. In short this dissertation made the following contributions:

Chapter 3 described our proposed conceptual model for the abstract building blocks of legal text. We described an attempt at reconciling the different types of semantic legal metadata proposed in the RE literature. Multiple conceptualizations of legal metadata have been developed. While the research community acknowledges the importance of semantic legal metadata, there is no consensus on the metadata types that are beneficial for legal compliance analysis. Indeed, these conceptualizations are at different levels of abstraction, depending on the targeted analysis as well as on the desired degree of interpretation. By looking at the literature, we have identified these conceptualizations and performed a mapping that reconciles these works into a general, high-level conceptual model that we deem general enough to be domain-independent, along with a precise definition for each of its elements.

Chapter 4 presented our approach for the automated extraction of semantic legal metadata. We derived, through a qualitative study extraction rules for the reconciled metadata types of the conceptual model presented in Chapter3. Given the established conceptual model, we devised extraction rules for the elements of the conceptual model through several qualitative studies and case studies performed over six legislative domains, including: traffic law, commerce law, environmental law, health law, penal law, and labor law. The extraction of semantic metadata is realized through subjecting individual legal statements to automated analysis, leveraging Natural Language Processing (especially constituency parsing and dependency parsing) and Machine Learning.

Chapter 5 presented our query system for extracting requirements related information from legal text. We described an industrial experience aimed at helping requirements analysts to query legal texts. We built a query system to streamline the validation of the automatically extracted semantic legal metadata. This is an advanced search facility over regulations. We showcase that semantic legal metadata can be successfully leveraged to answer requirements engineering-related questions. Hence, this query system enables resolving the relevance challenge. At the same time, the experience pinpoints for further improvements to the conceptual framework of semantic legal metadata.

Chapter 6 presented our solution for automated recommendation of templates for legal requirements. We propose an approach to automatically recommend templates for legal requirements based on legal statements, thus assisting requirements analysts with legal requirements elicitation. We then devised, using Natural Language Processing, automated rules for recommending suitable requirements templates. We investigate the use of requirements templates for the systematic elicitation of legal requirements. Subsequently, we conduct a qualitative study to evaluate our approach for template recommendation.

7.2 Limitations

As observed multiple times in this thesis, this research has a restricted scope and focus. In this section we discuss the validity limitations of the different case studies presented in earlier chapters, and the observed gaps between the semantic legal metadata pertinent to legal requirements elicitation and the legal concepts relevant to legal interpretation.

Construct validity: the investigation by the researchers of the relevant work for existing taxonomies of legal concepts, questions relevant to legal requirements analysts

and templates for legal requirements introduces the risk of subjective interpretation. To mitigate this threat, we reported to the best of our knowledge all the relevant elements of study identified in the literature and we tabulated alignments between the said elements and our proposed set of legal concepts, questions relevant to legal RE and templates for legal requirements. This examination of relevant work in the requirements engineering and AI and law communities was conducted over several passes by different researchers including a legal expert and a requirements analyst. By doing so, we ensured that no elements of study were overlooked and that the mappings were rooted in the definitions from the source work. While we cannot completely rule out subjectivity, our reporting of these alignment procedures was precise and thus open to scrutiny by the scientific community.

Internal validity: the evaluation of the different parts of the framework was performed against a ground truth constructed specifically for the respective use cases. Due to the scarcity of labeled datasets for legal requirements elicitation, the coding for the qualitative studies and the case studies presented herein was performed by the researchers. Given that one of the researchers is a legal expert, we found the risk of misinterpretation to be low. Add to that, we reported the inter-annotator agreement scores for all the annotation procedures where non-legal expert annotators took part. Finally, to ensure the reliability of the ground truth we devised our annotation procedures in terms of scope in accordance with the legal expert directions prior to any implementation of approaches. This point highlights the need for standardized legal datasets for future research involving multiple legal experts. We foresee that this protocol of annotation will introduce risks to validity given the different facets of legal interpretation.

External validity: we devised approaches that make use of linguistic cues as provided by the different NLP toolsets that we deemed acceptable for the French legal language. This poses a threat in terms of correctness of automatically retrieved linguistic cues and linguistic constructs such as parse trees, dependency graphs, and verb frames. This threat is bound to occur as NLP tools are not domain agnostic. We took steps to mitigate this threat by conducting preliminary validation passes over the source material as present in our datasets. Also, given the nuanced nature of legal texts, we devised heuristics to recover such errors. Following the evaluation phase, we analyzed the errors to assess whether design heuristics introduced errors of their own.

As discussed in the introduction and reported in each of the respective chapters, we identified obstacles to fully automate the elicitation of legal requirements at the different steps of the said process. We also reported the different lessons learned

concerning the use of NLP techniques in the legal domain within the scope of our work. One major lesson is the need for domain knowledge for the development but also the validation of the automation of legal text processing by a domain expert.

7.3 Future Work

In this dissertation, we focused on the parts of legal text that lead explicitly to legal requirements for IT systems.

In the future, it is important to process all of the legal text to uncover the implicit legal requirements. We already reported the specific types of legal statements that need to be properly handled to constitute the the missing bits and pieces of these legal requirements. Considering that such information might bring further improvements to the overall approach particularly for building a real-world domain model, further case studies and user studies need to be conducted.

We would further like to perform user studies in realistic settings to determine the practical utility of automation for legal metadata extraction. Another interesting field would be to adapt our approach to compliance rules for e-government applications. In addition, a foreseeable perspective is to dynamically adjust the extracted metadata and the proposed templates of legal requirements following amendments and changes in the regulations.

We also plan to include a domain modeling element in our approach. This would support the elicitation of legal requirements from inter-connected legal statements, thus resolving the challenges posed by metarules and cross-references. We would like to further expand our approach so that it not only recommends suitable templates, but also fills (populates) the templates by pulling in relevant information from the underlying legal statements.

Finally, we plan to explore the possibilities of combining all the proposed solutions into a commercial tool suite.

References

[1] "How much data do we create every day? the mind-blowing stats everyone should read," [https://bit.ly/2R0cN72.

[2] R. Wacks, *Law a very short introduction.* Oxford University Press, 2015.

[3] N. Uphoff, "Distinguishing power, authority & legitimacy: Taking max weber at his word by using resources-exchange analysis," *Polity*, vol. 22, p. 295, 01 1989.

[4] J. Searle, *The Construction of Social Reality.* Free Press, 1995.

[5] C. Palomares, C. Quer, and X. Franch, "Requirements reuse and requirement patterns: a state of the practice survey," *Empirical Software Engineering*, vol. 22, no. 6, pp. 2719–2762, 2017.

[6] S. Robertson, "Requirements patterns via events/use cases," in *Proceedings of the Third Conference on the Pattern Languages of Programs (PLoP'96)*, 1996.

[7] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "Patterns in property specifications for finite-state verification," in *Proceedings of the 21st International Conference on Software Engineering(ICSE'99)*, 1999, pp. 411—-420.

[8] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (ears)," in *Proceedings of the 17th IEEE International Requirements Engineering Conference (RE'09)*, 2009, pp. 317 – 322.

[9] T. D. Breaux and D. G. Gordon, "Regulatory requirements traceability and analysis using semi-formal specifications," in *Requirements Engineering: Foundation for Software Quality.* Springer, 2013, pp. 141–157.

[10] T. D. Breaux, M. W. Vail, and A. I. Antón, "Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations," in *Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06)*, 2006, pp. 46–55.

[11] J. Young and A. Antón, "A method for identifying software requirements based on policy commitments," in *Proceedings of the 18th IEEE International Requirements Engineering Conference (RE'10)*, 2010, pp. 47–56.

[12] Y. Yoshida, K. Honda, Y. Sei, H. Nakagawa, Y. Tahara, and A. Ohsuga, "Towards semi-automatic identification of functional requirements in legal texts for public administration," in *Proceedings of the 26th Annual Conferenceon Legal Knowledge and Information Systems (JURIX'13)*, 2013, pp. 175–184.

[13] M. Palmirani, G. Governatori, A. Rotolo, S. Tabet, H. Boley, and A. Paschke, "Legalruleml: Xml-based rules and norms," in *Proceedings of RuleML'11*, 2011, pp. 298–312.

[14] C. Arora, M. Sabetzadeh, L. C. Briand, and F. Zimmer, "Automated checking of conformance to requirements templates using natural language processing," *IEEE Transactions on Software Engineering*, vol. 41, no. 10, pp. 944–968, 2015.

[15] ——, "Extracting domain models from natural-language requirements: approach and industrial evaluation," in *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS'16)*, 2016, pp. 250–260.

[16] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with visual narrator," *Requirements Engineering*, vol. 22, no. 3, pp. 339–358, 2017.

[17] T. Quirchmayr, B. Paech, R. Kohl, H. Karey, and G. Kasdepke, "Semi-automatic rule-based domain terminology and software feature-relevant information extraction from natural language user manuals," *Empirical Software Engineering*, 2018.

[18] Y. Elrakaiby, A. Ferrari, P. Spoletini, S. Gnesi, and B. Nuseibeh, "Using argumentation to explain ambiguity in requirements elicitation interviews," in *Proceedings of the 25th IEEE International Requirements Engineering Conference (RE'17)*, 2017, pp. 51–60.

[19] B. Rosadini, A. Ferrari, G. Gori, A. Fantechi, S. Gnesi, I. Trotta, and S. Bacherini, "Using NLP to detect requirements defects: An industrial experience in the railway domain," in *Proceedings of the 23rd International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'17)*, 2017, pp. 344–360.

[20] J. Bhatia, T. D. Breaux, and F. Schaub, "Mining privacy goals from privacy policies using hybridized task recomposition," *ACM Transactions on Software Engineering and Methodology*, vol. 25, no. 3, pp. 22:1–22:24, 2016.

[21] J. Bhatia, M. C. Evans, S. Wadkar, and T. D. Breaux, "Automated extraction of regulated information types using hyponymy relations," in *Proceedings of the 3rd International Workshop on Artificial Intelligence for Requirements Engineering (AIRE'16)*, 2016, pp. 19–25.

[22] M. C. Evans, J. Bhatia, S. Wadkar, and T. D. Breaux, "An evaluation of constituency-based hyponymy extraction from privacy policies," in *Proceedings of the 25th IEEE International Requirements Engineering Conference (RE'17)*, 2017, pp. 312–321.

[23] L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson, "Semantic role labeling: An introduction to the special issue," *Computational Linguistics*, vol. 34, no. 2, pp. 145–159, 2008.

[24] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational Linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[25] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, 1998, pp. 86–90.

[26] Y. Miyao and J. Tsujii, "Deep linguistic analysis for the accurate identification of predicate-argument relations," in *Proceedings of the 20th International Conference on Computational Linguistics (COLING'04)*, 2004, pp. 1392–1398.

[27] K. Kipper, A. Korhonen, N. Ryant, and M. Palmer, "A large-scale classification of english verbs," *Language Resources and Evaluation*, vol. 42, no. 1, pp. 21–40, 2008.

[28] B. Levin, *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, 1993.

[29] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[30] R. Levy and G. Andrew, "Tregex and tsurgeon: tools for querying and manipulating tree data structures," in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, 2006, pp. 2231–2234.

[31] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*, 2014.

[32] J. F. Horty, *Agency and Deontic Logic*, ser. Oxford scholarship online. Oxford University Press, USA, 2001.

[33] W. N. Hohfeld, "Fundamental legal conceptions as applied in judicial reasoning," *The Yale Law Journal*, vol. 26, no. 8, pp. 710–770, 1917.

[34] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Modeling security requirements through ownership, permission and delegation," in *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*, 2005, pp. 167–176.

[35] N. Kiyavitskaya, N. Zeni, L. Mich, J. R. Cordy, and J. Mylopoulos, "Text mining through semi automatic semantic annotation," in *Proceedings of the 6th International Conference on Practical Aspects of Knowledge Management (PAKM'06)*, 2006, pp. 143–154.

[36] N. Kiyavitskaya, N. Zeni, T. D. Breaux, A. I. Antón, J. R. Cordy, L. Mich, and J. Mylopoulos, "Automating the extraction of rights and obligations for regulatory compliance," in *Proceedings of ER'08*, 2008, pp. 154–168.

[37] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "GaiusT: supporting the extraction of rights and obligations for regulatory compliance," *Requirements Engineering*, vol. 20, no. 1, pp. 1–22, 2015.

[38] A. Siena, J. Mylopoulos, A. Perini, and A. Susi, "Designing law-compliant software requirements," in *Proceedings of ER'09*, 2009, pp. 472–486.

[39] N. Zeni, E. A. Seid, P. Engiel, S. Ingolfo, and J. Mylopoulos, "Building large models of law with NómosT," in *Proceedings of ER'16*, 2016, pp. 233–247.

[40] A. Siena, I. Jureta, S. Ingolfo, A. Susi, A. Perini, and J. Mylopoulos, "Capturing variability of law with nómos 2," in *Proceedings of the 31st International Conference on Conceptual Modeling (ER'12)*, 2012, pp. 383–396.

[41] S. Ingolfo, I. Jureta, A. Siena, A. Perini, and A. Susi, "Nòmos 3: Legal compliance of roles and requirements," in *Proceedings of the 33rd International Conference on Conceptual Modeling (ER'14)*, 2014, pp. 275–288.

[42] S. Ghanavati, D. Amyot, and A. Rifaut, "Legal goal-oriented requirement language (legal GRL) for modeling regulations," in *Proceedings of MISE'14*, 2014, pp. 1–6.

[43] S. Ghanavati, "Legal-urn framework for legal compliance of business processes," Ph.D. dissertation, University of Ottawa, Ottawa, Ontario, Canada, 2013.

[44] T. Breaux, "Legal requirements acquisition for the specification of legally compliant information systems," Ph.D. dissertation, North Carolina State University, Raleigh, North Carolina, USA, 2009.

[45] T. D. Breaux and A. I. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 5–20, 2008.

[46] J. C. Maxwell and A. I. Antón, "The production rule framework: developing a canonical set of software requirements for compliance with law," in *Proceedings of IHI'10*, 2010, pp. 629–636.

[47] A. Massey, "Legal requirements metrics for compliance analysis," Ph.D. dissertation, North Carolina State University, Raleigh, North Carolina, USA, 2012.

[48] A. K. Massey, P. N. Otto, L. J. Hayward, and A. I. Antón, "Evaluating existing security and privacy requirements for legal compliance," *Requirements Engineering*, vol. 15, no. 1, pp. 119–137, 2010.

[49] G. Boella, L. D. Caro, L. Humphreys, L. Robaldo, P. Rossi, and L. van der Torre, "Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law," *Artificial Intelligence and Law*, vol. 24, no. 3, pp. 245–283, 2016.

[50] W. Peters, M. Sagri, and D. Tiscornia, "The structuring of legal knowledge in LOIS," *Artificial Intelligence and Law*, vol. 15, no. 2, pp. 117–135, 2007.

[51] G. Sartor, P. Casanovas, M. Biasiotti, and M. Fernndez-Barrera, *Approaches to Legal Ontologies: Theories, Domains, Methodologies.* Springer, 2013.

[52] R. Hoekstra, J. Breuker, M. D. Bello, and A. Boer, "The LKIF core ontology of basic legal concepts," in *Proceedings of the 2nd Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT'07)*, 2007, pp. 43–63.

[53] J. Breuker, A. Boer, R. Hoekstra, and K. van den Berg, "Developing content for LKIF: ontologies and frameworks for legal reasoning," in *Proceedings of the 19th Annual Conference on Legal Knowledge and Information Systems (JURIX'06)*, 2006, pp. 169–174.

[54] A. Boer, R. Winkels, and F. Vitali, "Proposed XML standards for law: Metalex and LKIF," in *Proceedings of the 20th Annual Conference on Legal Knowledge and Information Systems (JURIX'07)*, 2007, pp. 19–28.

[55] T. Athan, H. Boley, G. Governatori, M. Palmirani, A. Paschke, and A. Z. Wyner, "OASIS LegalRuleML," in *Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL'13)*, 2013, pp. 3–12.

[56] H. Lam, M. Hashmi, and B. Scofield, "Enabling reasoning with LegalRuleML," in *Proceedings of the 10th International Symposium on Rule Technologies. Research, Tools, and Applications (RuleML'16)*, 2016, pp. 241–257.

[57] "Specification of RuleML 1.02," http://wiki.ruleml.org/index.php/Specification_of_RuleML_1.02.

[58] D. G. Gordon and T. D. Breaux, "Reconciling multi-jurisdictional legal requirements: A case study in requirements water marking," in *Proceedings of the 20th IEEE International Requirements Engineering Conference (RE'12)*, 2012, pp. 91–100.

[59] N. Sannier, M. Adedjouma, M. Sabetzadeh, and L. C. Briand, "An automated framework for detection and resolution of cross references in legal texts," *Requirements Engineering*, vol. 22, no. 2, pp. 215–237, 2017.

[60] "Online Annex," https://sites.google.com/view/metax-re2018/.

[61] D. Grossi, J.-J. C. Meyer, and F. Dignum, "The many faces of counts-as: A formal analysis of constitutive rules," *Journal of Applied Logic*, vol. 6, no. 2, pp. 192 – 217, 2008, selected papers from the 8th International Workshop on Deontic Logic in Computer Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570868307000559

[62] F. Dell'Orletta, S. Marchi, S. Montemagni, B. Plank, and G. Venturi, "The SPLeT–2012 shared task on dependency parsing of legal texts," in *Proceedings of SPLeT'12*, 2012, pp. 42–51.

[63] J. Saldaña, *The Coding Manual for Qualitative Researchers*. Sage, 2015.

[64] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, 1960.

[65] J. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[66] J. C. Maxwell, A. I. Antón, P. P. Swire, M. Riaz, and C. M. McCraw, "A legal cross-references taxonomy for reasoning about compliance requirements," *Requirements Engineering*, vol. 17, no. 2, pp. 99–115, 2012.

[67] N. Sannier, M. Adedjouma, M. Sabetzadeh, and L. C. Briand, "Automated classification of legal cross references based on semantic intent," in *Proceedings of REFSQ'16*, 2016, pp. 119–134.

[68] "Wiktionary," https://fr.wiktionary.org/.

[69] Princeton University, "About WordNet," http://wordnet.princeton.edu, 2010.

[70] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," in *the 38th Annual Conference of the Association for Computational Linguistics (ACL-00)*, 2000, pp. 512–520.

[71] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[72] Q. Pradet, L. Danlos, and G. de Chalendar, "Adapting verbnet to french using existing resources," in *the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, May 2014, pp. 1122–1126.

[73] E. Frank, M. A. Hall, , and I. H. Witten, "The WEKA workbench. online appendix for "data mining: Practical machine learning tools and techniques"," 2016.

[74] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-weka: Combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD*, 2013, pp. 847–855.

[75] B. Sagot, "The Lefff, a Freely Available and Large-coverage Morphological and Syntactic Lexicon for French," in *Proceedings of the International Conference on Language Resources and Evaluation(LREC'10)*, 2010, pp. 2745–2751.

[76] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL'06)*, 2006.

[77] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi, "Maltparser: A language-independent system for data-driven dependency parsing," *Natural Language Engineering*, vol. 13, no. 2, pp. 95–135, 2007.

[78] R. T. McDonald and J. Nivre, "Characterizing the errors of data-driven dependency parsing models," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'07)*, 2007, pp. 122–131.

[79] J. K. Kummerfeld, D. L. W. Hall, J. R. Curran, and D. Klein, "Parser showdown at the wall street corral: An empirical investigation of error types in parser output," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'12)*, 2012, pp. 1048–1059.

[80] The European Parliament and the Council of the European Union, "General data protection regulation," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679.

[81] A. Falkner, C. Palomares, X. Franch, G. Schenner, P. Aznar, and A. Schoerghuber, "Identifying requirements in requests for proposal," in *Proceedings of REFSQ'19*, 2019, pp. 176–182.

[82] N. Jha and A. Mahmoud, "Mining user requirements from application store reviews using frame semantics," in *Proceedings of REFSQ'17*, 2017, pp. 273–287.

[83] E. Guzman, M. Ibrahim, and M. Glinz, "A little bird told me: Mining tweets for requirements and software evolution," in *Proceedings of RE'17*, 2017, pp. 11–20.

[84] G. Williams and A. Mahmoud, "Mining twitter feeds for software user requirements," in *Proceedings of RE'17*, 2017, pp. 1–10.

[85] T. Quirchmayr, B. Paech, R. Kohl, H. Karey, and G. Kasdepke, "Semi-automatic rule-based domain terminology and software feature-relevant information extraction from natural language user manuals," *Empirical Software Engineering*, 2018.

[86] I. T. Koitz and M. Glinz, "A fuzzy galois lattices approach to requirements elicitation for cloud services," *IEEE Transactions on Services Computing*, vol. 11, no. 5, pp. 768–781, 2018.

[87] T. D. Breaux and A. I. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 5–20, 2008.

[88] P. Mäder and J. Cleland-Huang, "A visual language for modeling and executing traceability queries," *Software and System Modeling*, vol. 12, no. 3, pp. 537–553, 2013.

[89] N. Sannier and B. Baudry, "INCREMENT: A mixed MDE-IR approach for regulatory requirements modeling and analysis," in *Proceedings of REFSQ'14*, 2014, pp. 135–151.

[90] P. Pruski, S. Lohar, W. Goss, A. Rasin, and J. Cleland-Huang, "Tiqi: answering unstructured natural language trace queries," *Requirements Engineering*, vol. 20, no. 3, pp. 215–232, 2015.

[91] G. M. Kanchev, P. K. Murukannaiah, A. K. Chopra, and P. Sawyer, "Canary: Extracting requirements-related information from online discussions," in *Proceedings of RE'17*, 2017, pp. 31–40.

[92] M. van Opijnen and C. Santos, "On the concept of relevance in legal information retrieval," *Artificial Intelligence and Law*, vol. 25, no. 1, pp. 65–87, 2017.

[93] P. Quaresma and I. Pimenta Rodrigues, "A question answer system for legal information retrieval," in *Proceedings of JURIX'05*, 2005, pp. 91–100.

[94] A. Wyner, F. Gough, F. Levy, M. Lynch, and A. Nazarenko, "On annotation of the textual contents of scottish legal instruments," in *Proceedings of JURIX'17*, 2017, pp. 101–106.

[95] F. Gandon, G. Governatori, and S. Villata, "Normative requirements as linked data," in *Proceedings of JURIX'17*, 2017, pp. 1–10.

[96] K. T. Maxwell and B. Schafer, "Concept and context in legal information retrieval," in *Proceedings of JURIX'08*, 2008, pp. 63–72.

[97] N. Mimouni, A. Nazarenko, and S. Salotti, "Answering complex queries on legal networks: A direct and a structured IR approaches," in *Proceedings of AICOL'17*, 2017, pp. 451–464.

[98] I. Chalkidis, C. Nikolaou, P. Soursos, and M. Koubarakis, "Modeling and querying greek legislation using semantic web technologies," in *Proceedings of ESWC'17*, 2017, pp. 591–606.

[99] P. Do, H. Nguyen, C. Tran, M. Nguyen, and M. Nguyen, "Legal question answering using ranking SVM and deep convolutional neural network," *CoRR*, vol. abs/1703.05320, 2017.

[100] K. J. Adebayo, L. D. Caro, G. Boella, and C. Bartolini, "An approach to information retrieval and question answering in the legal domain," in *Proceedings of JURISIN'16*, 2016.

[101] D. Collarana, T. Heuss, J. Lehmann, I. Lytra, G. Maheshwari, R. Nedelchev, T. Schmidt, and P. Trivedi, "A question answering system on regulatory documents," in *Proceedings of JURIX'18*, 2018, pp. 41–50.

[102] G. Boella, L. D. Caro, L. Humphreys, L. Robaldo, P. Rossi, and L. van der Torre, "Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law," *Artificial Intelligence and Law*, vol. 24, no. 3, pp. 245–283, 2016.

[103] N. Sannier, M. Adedjouma, M. Sabetzadeh, L. C. Briand, J. Dann, M. Hisette, and P. Thill, "Legal markup generation in the large: An experience report," in *Proceedings of RE'17*, 2017, pp. 302–311.

[104] "ELI - the European Legislation Identifier," http://eur-lex.europa.eu/eli-register/about.html.

[105] World Wide Web Consortium, "Resource Description Framework (RDF)," https://www.w3.org/RDF/.

[106] ——, "SPARQL Query Language for RDF," https://www.w3.org/TR/rdf-sparql-query/.

[107] B. Nuseibeh and S. M. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of ICSE-FOSE'2000*, 2000, pp. 35–46.

[108] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Eds. Springer, 2005, ch. 2, pp. 19–46.

[109] T. Fritz and G. C. Murphy, "Using information fragments to answer the questions developers ask," in *Proceedings of ICSE'10*, 2010, pp. 175–184.

[110] S. Malviya, M. Vierhauser, J. Cleland-Huang, and S. Ghaisas, "What questions do requirements engineers ask?" in *Proceedings of RE'17*, 2017, pp. 100–109.

[111] M. Jackson, "The world and the machine," in *Proceedings of ICSE'95*, 1995, pp. 283–292.

[112] A. K. Massey, P. N. Otto, and A. I. Antón, "Prioritizing legal requirements," in *Proceedings of RELAW'09*, 2009, pp. 27–32.

[113] P. Zave and M. Jackson, "Four dark corners of requirements engineering," *ACM Transanctions on Software Engineering and Methodololgy*, vol. 6, no. 1, pp. 1–30, Jan. 1997.

[114] S. Robertson and J. Robertson, *Mastering the Requirements Process*. Addison-Wesley Professional, 2006.

[115] B. of International Settlements, "Basel II: International convergence of capital measurement and capital standards: a revised framework," 2004.

[116] M. Laycock, *Operational Risk Reporting Standards (ORRS) Edition 2011*, ORX Association, 2012.

[117] G. Soltana, N. Sannier, M. Sabetzadeh, and L. C. Briand, "Model-based simulation of legal policies: Framework, tool support, and validation," *Software & Systems Modeling*, vol. 17, no. 3, pp. 851–883, 2018.

[118] M. Ceci, T. Butler, L. O'Brien, and F. Al Khalil, "Legal patterns for different constitutive rules," in *Proceedings of AICOL'18*, 2018, pp. 105–123.

[119] A. Sleimi, N. Sannier, M. Sabetzadeh, L. C. Briand, and J. Dann, "Automated extraction of semantic legal metadata using natural language processing," in *Proceedings of RE'18*, 2018, pp. 302–311.

[120] P. Darke and G. Shanks, "Stakeholder viewpoints in requirements definition: A framework for understanding viewpoint development approaches," *Requirements Engineering*, vol. 1, no. 2, pp. 88–105, 1996.

[121] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering*. Springer London, 2010.

[122] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Use Case Approach*. Addison-Wesley, 2003.

[123] R. Young, *Effective Requirements Practices*. Addison-Wesley, 2001.

[124] J. Bentham and H. Hart, *Of laws in general*. University of London, Athlone Press, 1945.

[125] N. Kiyavitskaya, A. Krausová, and N. Zannone, "Why eliciting and managing legal requirements is hard," in *2008 Requirements Engineering and Law*, 2008, pp. 26–30.

[126] N. Zeni, L. Mich, and J. Mylopoulos, "Annotating legal documents with gaiust 2.0," *Int. J. Metadata Semant. Ontologies*, vol. 12, no. 1, pp. 47–58, 2017.

[127] A. Ottensooser, A. Fekete, H. A. Reijers, J. Mendling, and C. Menictas, "Making sense of business process descriptions: An experimental comparison of graphical and textual notations," *Journal of Systems and Software*, vol. 85, no. 3, pp. 596–606, 2012.

[128] Z. Sharafi, A. Marchetto, A. Susi, G. Antoniol, and Y. Guéhéneuc, "An empirical study on the efficiency of graphical vs. textual representations in requirements comprehension," in *Proceedings of the 21st IEEE International Conference on Program Comprehension (ICPC'13)*, 2013, pp. 33–42.

[129] S. Ghanavati, D. Amyot, and A. Rifaut, "Legal goal-oriented requirement language (legal GRL) for modeling regulations," in *6th International Workshop on Modeling in Software Engineering, MiSE*. ACM, 2014, pp. 1–6.

[130] M. Ceci, F. A. Khalil, and L. O'Brien, "Making sense of regulations with sbvr," in *RuleML*, 2016.

[131] T. D. Breaux, "A method to acquire compliance monitors from regulations," in *Proceedings of the 3rd International Workshop on Requirements Engineering and Law (RELAW'10)*, 2010, pp. 17–26.

[132] R. Levy and G. Andrew, "Tregex and tsurgeon: tools for querying and manipulating tree data structures," in *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*, 2006, pp. 2231–2234.

[133] O. Gurevich, R. Crouch, T. King, and V. De Paiva, "Deverbal nouns in knowledge representation," *Journal of Logic and Computation*, vol. 18, 01 2006.

Appendix A

List of modal verbs

- sont soumis
- seront soumis
- sera soumis
- est soumis
- est soumise
- sont soumises
- il doit
- elle doit
- ils doivent
- elles doivent
- il devra
- elle devra
- ils devront
- elles devront
- il oblige
- elle oblige
- ils obligent
- elles obligent
- il obligera
- elle obligera
- ils obligeront
- elles obligeront
- obligent
- soumis
- soumise
- doit
- devra
- doivent
- devront
- obligation
- obligé
- devoir

- est tenue
- est tenu
- il est obligé
- elle est obligeé
- est obligée
- obligé
- il est nécessaire
- nécessaire
- toujours
- exigence
- astreint
- astreinte
- astreint
- vassuré
- assurée
- oblige
- requis
- requise
- Permission
- susceptible
- droit
- facultative
- ont
- elles peuvent
- ils pourront
- il peut
- elle peut
- pourront
- permission
- peut
- vpeuvent
- pouvoir
- pourront
- pourra
- pouvant
- permis
- permise
- est facultative
- autorise
- autorisé
- autorisée
- puisse
- autorisation
- possible
- ont droit
- a droit
- aura droit
- sont en droit

- est en droit
- sera en droit
- seront en droit
- sont susceptibles
- est susceptible
- sera susceptible
- seront susceptibles
- Prohibition
- interdit
- ne doit
- n' est pas en droit
- est interdite
- il est interdit
- est interdit
- ne peut
- ne pourra
- ne peuvent
- ne sont pas autorisé
- est prohibé
- est prohibée
- prohibé
- interdiction
- est illégal
- est réprouvé
- est réprouvée
- proscrit
- proscrite
- est illicite