Automated Extraction of Semantic Legal Metadata Using Natural Language Processing

Amin Sleimi*, Nicolas Sannier*, Mehrdad Sabetzadeh*, Lionel C. Briand*, John Dann§
*SnT Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg
§Central Legislative Service (SCL), Ministry of State, Luxembourg
Email: {sleimi, sannier, sabetzadeh, briand}@svv.lu, john.dann@scl.etat.lu

***Abstract*—*[Context]* 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. *[Objectives]* Our work is motivated by two observations: (1) The existing requirements engineering (RE) literature does not provide a harmonized view on the semantic metadata types that are useful for legal requirements analysis. (2) Automated support for the extraction of semantic legal metadata is scarce, and further does not exploit the full potential of natural language processing (NLP). Our objective is to take steps toward addressing these limitations. *[Methods]* We review and reconcile the semantic legal metadata types proposed in RE. Subsequently, we conduct a qualitative study aimed at investigating how the identified metadata types can be extracted automatically. *[Results and Conclusions]* We propose (1) a harmonized conceptual model for the semantic metadata types pertinent to legal requirements analysis, and (2) automated extraction rules for these metadata types based on NLP. We evaluate the extraction rules through a case study. Our results indicate that the rules generate metadata annotations with high accuracy.**

***Index Terms*—Legal Requirements, Semantic Legal Metadata, Natural Language Processing (NLP).**

I. Introduction

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 [1], [2], [3]. 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.

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 [4], [5] and prioritization [2], [6]. Semantic metadata is a prerequisite for the systematic derivation of compliance requirements [1], [7], [8], [9], and transitioning from legal texts to formal specifications [10] or models [3], [8], [11].

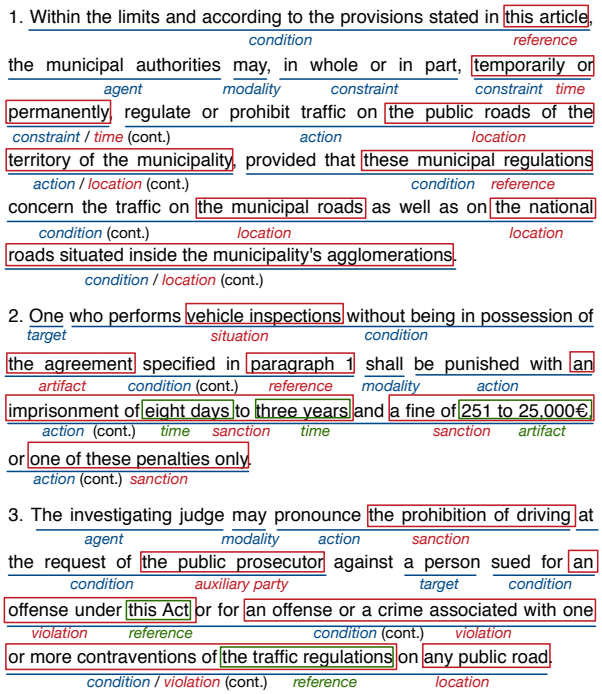

Fig. 1. Examples of Semantic Legal Metadata Annotations

In this paper, we concern ourselves with *semantic legal metadata*. In Fig. 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 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 III, we will further explain the metadata types illustrated in Fig. 1.

The example statements in Fig. 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. 1 provide useful information to requirements analysts. Indeed, and as we argue more precisely in Section II, 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 elicitation (agents and auxiliary parties), as well as the way these stakeholders should interact, potentially using computer systems.

Our work in this paper is motivated by two observed limitations in the state-of-the-art on semantic legal metadata: ***1) 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 [7], [3], [10], [12], [13], but no strand completely covers the others. ***2) NLP's under-exploited potential for metadata extraction.*** If done manually, enhancing a large corpus of legal texts with semantic metadata is extremely laborious. Recently, increasing attention has been paid to automating this task using natural language processing (NLP). Notable initiatives aimed at providing automation for metadata extraction are GaiusT [3] and NomosT [11]. These initiatives do not handle the broader set of metadata types proposed in the RE literature, e.g., locations proposed by Breaux [7], objects by Massey [2], and situation by Siena et al. [13]. Further, they rely primarily on simple NLP techniques, e.g., tokenization, named-entity recognition, and part-of-speech (POS) tagging. Simple NLP techniques have the advantage that they are less likely to make mistakes. Nevertheless, such techniques 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 [14]. 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. 1, the roles of the (sued) person, the investigating judge and the public prosecutor can be derived from such dependencies. Constituency parsing is important for delineating the right span for metadata annotations. For instance, in Statement 1 of Fig. 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, and thus cannot identify the right span for the *location* annotation. To the best of our knowledge, a full-fledged application of NLP, including constituency and dependency parsing, has not yet been attempted over legal texts for a broad scope of metadata types.

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

RQ1: What are the semantic legal metadata types used in RE? RQ1 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 RQ1 is the first contribution of the paper: *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 [15].*

RQ2: Can one define semantic legal metadata extraction rules over constituency and dependency parsing results? RQ2 investigates whether one can readily define rules for extracting semantic legal metadata using constituency and dependency parsing. To answer RQ2, we perform a qualitative study over 200 legal statements from the traffic laws for Luxembourg. Specifically, we annotate the legal statements in question with the legal metadata types established in RQ1. We use the results of this study for defining rules that can automatically detect the annotations. The answer to RQ2 is the second contribution of the paper: *a set of NLP-based rules for automated extraction of semantic legal metadata. Our rules, which leverage constituency and dependency parsing, cover the majority of the phrase-level metadata types identified in RQ1.*

RQ3: How accurate is semantic legal metadata extraction using constituency and dependency parsing? RQ3, posed in light of a positive answer to RQ2, aims at evaluating the accuracy of our extraction rules. Our evaluation is based on 150 new legal statements from the traffic laws. For our evaluation, we adapt precision and recall so that they account not only for the correct assignment of metadata types by the extraction rules, but also for the correct delineation of text spans to which the metadata annotations are applied. Both factors are important, since mistakes in either the type or the span lead to manual effort. Specifically, our adapted notions of precision and recall levy a penalty over annotations whose type is correct but whose span is only partially correct. Overall, our approach has an (adapted) precision of 87.4% and an (adapted) recall of 85.5%. When only type assignment is considered, precision stands at 97.2% and recall at 94.9%.

Overview and Structure. Section II reviews background and related work. Section III describes our conceptual model for semantic legal metadata. Section IV presents our qualitative study and the extraction rules resulting from it. Section V evaluates the accuracy of our extraction rules. Section VI discusses threats to validity. Section VII concludes the paper.

II. Background and Related Work

We begin with background information on deontic logic and the Hohfeldian system. These serve as the foundations for most work in the area of legal analysis. Next, we discuss the related work on semantic legal metadata. Finally, we position our technical approach by explaining how constituency and dependency parsing have been used previously in RE.

A. Preliminaries

When trying to interpret and analyze the semantics of the law, most existing research takes its root in either deontic logic [16] or the Hohfeldian system of legal concepts [17]. Deontic logic distinguishes "what is permissible" (permission or right) from "what ought to be" (obligation) and their negations: what is "impermissible" ("prohibition") and what "not ought to be" ("omissible" or non-obligatory), respectively.

The Hohfeldian system [17] 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.

B. 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. 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 III.

Early foundations. Two of the earliest research strands in RE on extracting information from legal texts are by Giorgini et al. [18] and Breaux et al. [19]. 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 [20], Kiyavitskaya et al. [12] 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** [3]. 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** [8], [11], [13], [21] 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** [11] 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** [22], [23] 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 [19] and motivated by deriving compliance requirements, Breaux et al. [7], [1] 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 [10] 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. [6], [2] 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*.

C. Semantic Metadata in Legal Knowledge Representation

There is considerable research in the legal knowledge representation community on formalizing legal knowledge [24]. Several ontologies have been developed for different dimensions of legal concepts [25], [26]. Our goal here is not to give a thorough exposition of these ontologies, because our focus is on the metadata types (discussed in Section II-B) for which clear use cases exist in the RE community.

The above said, 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 [27], [28], [29] and LegalRuleML [30], [31], 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 [30], [31] – a successor of LKIF – tailors the generic RuleML language [32] 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 II-B. In Section III, we reconcile all the RE-related legal concepts identified in an attempt to provide a unified model of legal metadata for RE.

D. Constituency and Dependency Parsing in RE

As mentioned already, the main enabling techniques we employ from NLP 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 [33], model extraction [34], [35], feature extraction [36], and ambiguity and defect detection [37], [38].

In relation to legal requirements specifically, Bhatia et al. [9], [39] and Evans et al. [40] 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.

III. A Model of Semantic Legal Metadata (RQ1)

Our conceptual model for semantic legal metadata is presented in Fig. 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. [1], Maxwell and Antón [10], Siena et al. [13], Massey et al. [2], Ghanavati et al. [22] and Zeni et al [3]. The model derives the majority – 83.3% (20/24), to be precise – of its concepts from the work of Breaux et al.'s [1] and Zeni et al.'s [3]. 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 [15]. 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 [27] and LegalRuleML [30], outlined in Section II-C. 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. [1]. 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. 1. *Agent* is an actor performing an action, whereas *target* is an actor affected by the enforcement of a provision. A third

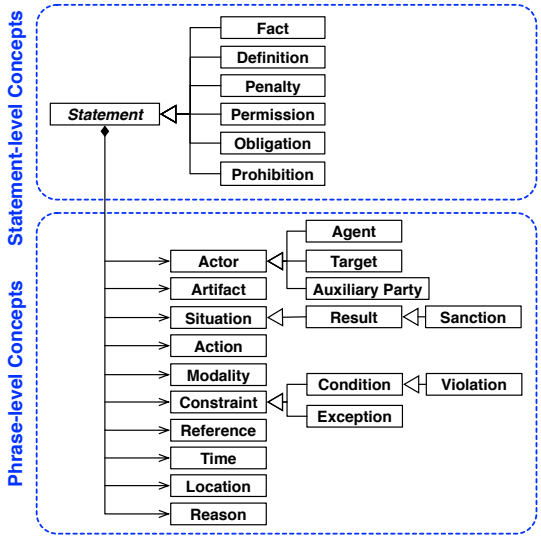

Fig. 2. Conceptual Model for Semantic Legal Metadata Relevant to RE

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 [13]. 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 [13], an *action* in GaiusT [3], an *act* in Breaux et al.'s upper ontology [1] and a *clause* in LegalGRL [22]. In our model, we follow GaiusT's terminology. As illustrated by our statements in Fig. 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. 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. 1.

As a final remark, we note that not all the concepts discussed in Section II 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 directly 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.

IV. Extracting Semantic Legal Metadata (RQ2)

In this section, we report on a qualitative study aimed at defining extraction rules for semantic legal metadata. Our study focuses exclusively on phrase-level metadata. Ascribing (statement-level) metadata to whole statements requires knowledge of metadata at the level of phrases. We leave statement-level metadata extraction to future studies.

Study context and data selection. We conducted our study in collaboration with Luxembourg's Central Legislative Service (in French, 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 with identifying legal provisions that are likely to imply software requirements; in Section I, we illustrated some possible use cases of semantic legal metadata for requirements analysts, and (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 the maximum speed limit of 30 km/h?" Our work is motivated by the former use case for legal metadata.

Our study concentrates on the traffic laws for Luxembourg. The traffic laws are made up of 74 separate legal texts, including legislation, regulations, orders and jurisprudence. Collectively the texts are 1075 pages long and contain ≈12000 statements. The oldest text is from 1955 and the most recent one is 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 to be a good showcase for demonstrating 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, courts, and public infrastructure management departments.

Our study is based on 200 randomly selected statements from the traffic laws. As 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 preprocessing measures, notably merging the beginning of a statement with its individual list items to form complete, independent sentences [41].

Analysis procedure. Our analysis procedure follows *protocol coding* [42], which is a method for collecting qualitative data according to a pre-established theory, i.e., set of codes. In our study, the codes are the phrase-level concepts of the model of Fig. 2. The first author, 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

TABLE I
METADATA ANNOTATIONS RESULTING FROM QUALITATIVE STUDY

Concept	Unique Classification	Multiple Classifications
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
Total	**1299**	**40**

discussed between the authors (including a legal expert) and decisions were made based on consensus.

To assess the overall reliability of the coding, the second author – a native French speaker with background in NLP and regulatory compliance – independently annotated 10% of the selected statements, prior to any discussion among the authors. Interrater agreement was then computed using Cohen's κ [43]. An agreement was counted 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" [44].

Coding results. The coding process did not prompt the use of any concepts beyond what was already present in the conceptual model of Fig. 2. In other words, we found the phrase-level concepts of the model to be adequately expressive.

Table I presents overall statistics about the number of occurrences of each phrase-level concept in the studied statements. In the majority of cases, we could assign a unique annotation to a given phrase. However, we did encounter cases where different annotations would result from different interpretations of the same phrase. The last column of the table provides information about phrases with multiple annotations. 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. 1. What we show in the last column of Table I is exclusive of nesting, and covers only phrases with more than one annotation attached to exactly the same span. An example such phrase is "temporarily or permanently" in Statement 1 of Fig. 1. Here, two annotations, *constraint* and *time*, have been attached to the same span.

In total, we annotated 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 (i.e., no cases observed with more than two annotations).

With regard to the coverage of the concepts, we have more than 20 occurrences of each concept, with two exceptions: *result* is not represented, and *constraint* has only five occurrences. Despite our study not having identified any occurrences of *result*, the concept is conceptually important. In particular, feedback from legal experts indicated that there is a gap between *situation* and *sanction*. To illustrate, consider the following statement (from outside our qualitative study): "If the defect is 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 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. 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 restriction qualifies neither as a *condition* nor an *exception*.

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

Metadata extraction rules. Table II 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. 2. The concepts that are not covered are: *result* and *constraint* (due to the lack of enough observations, noted above), the three specializations of *actor*, and *(cross-)reference*.

With regard to the specializations of *actor*, namely *agent*, *target* and *auxiliary party*, we observed that distinguishing them is highly context dependent. We thus deemed the risk of overfitting to be high if rules were to be defined for these specializations. Our rules instead directly address *actor*.

With regard to *references*, we made a conscious choice not to cover them in our extraction rules. Legal cross-references are well-studied in RE, with detailed semantic classifications already available [45], [46]. As for automated extraction of cross-reference metadata, one can for example use the extraction rules of Sannier et al.'s [5], [46].

The element highlighted blue in each rule of Table II is the phrase that is the target of annotation by that rule. The rules for *actor* use both constituency and dependency parsing, whereas the remaining rules use only constituency parsing. Aside from the rules for *action* and *actor*, all the rules are expressed entirely in Tregex [47], a widely used pattern matching language for (constituency) parse trees. The single rule for *action* annotates every verb phrase (VP) encountered, 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 exposition of Tregex which is already well-documented [47]. Below, we illustrate some

TABLE II
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

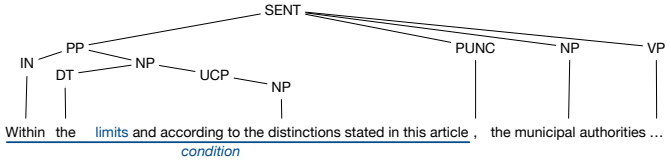

Fig. 3. (Simplified) Parse Tree for an Excerpt of Statement 1 (from Fig. 1)

of our rules to facilitate understanding, and further to discuss some important technicalities of the rules in general.

Consider Statement 1 in Fig. 1. A (simplified) parse tree for an excerpt of this statement is shown in Fig. 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" is such a condition marker. Initial sets of markers for all the concepts in Table II, including for conditions, were gleaned from our analysis of the 200 annotated statements in our study. 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 III illustrates the markers for different concepts. We note that the original markers are in French; the terms in Table III are translations. We further note that, for simplicity, the table provides one set of markers per concept. In practice, different rules for extracting the same concept use different marker subsets. For instance, "who" and "whose" are treated as concept markers by the first *condition* rule in Table II (Srel << (condition marker)), but not by 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 based on a dictionary. Specifically, we scraped all

TABLE III
MARKERS FOR DIFFERENT METADATA TYPES

Concept	Examples of Markers (Non-exhaustive)
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).

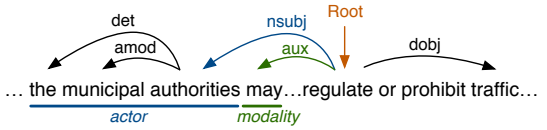

Fig. 4. (Simplified) Dependency Graph for an Excerpt of Statement 1

the entries in Wiktionary [48]. Any entry classified as a noun and with a definition containing "act" or "action" (or variations thereof) is considered a marker for *situation*. For instance, consider the term "inspection", defined by Wiktionary as "*The act of* examining something, often closely." With "inspection" included in the situation markers, the rule for *situation* in Table II, NP < (situation marker) would mark the noun phrase "vehicle inspections" in Statement 2 of Fig. 1 as a *situation*.

In a similar vein, any Wiktionary entry classified as a noun and with a definition containing "person", "organization", "body" (or variations thereof) is considered a marker for *actor*. For example, "authority" is an actor marker since Wiktionary defines it as "The *bodies* that enforce law and order [...]." As shown by the rules in Table II, the mere presence of an actor marker does not necessarily induce an *actor* annotation: an *actor* further has to appear in a subject or object dependency as defined by the rules. To illustrate, let us again consider Statement 1 of Fig. 1. A (simplified) dependency graph for an excerpt of this statement is given in Fig. 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 further has a subject dependency (*nsubj*) to the main (root) verb within the statement.

For *artifacts*, we need the ability to identify human-made objects. One can develop generalizable automation for this purpose in the English language, where one has at their disposal ontologies, notably the WordNet ontology [49], providing a classification of objects. In lieu of such an ontology for French, we derived an initial set of markers from the 200 statements studied. We then enhanced these markers by inspecting their synonyms in a thesaurus and retaining what we found relevant. In addition, we implement a heuristic (second rule under *artifact* in Table II), classifying 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 III, noting the nature of the markers for these seven concepts, the number of possibilities is limited. While, in all likelihood, our qualitative study did not capture all the possibilities, we anticipate that the markers for these concepts will saturate quickly with use.

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 a specific legal domain.

V. Empirical Evaluation (RQ3)

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

A. Implementation

Our metadata extraction rules are implemented using Tregex [47] and Java. These rules utilize the outputs of the classic NLP pipeline for syntactic analysis. The pipeline has the following modules: Tokenizer, Sentence Splitter, POS Tagger, Named-entity Recognizer, and Parser (Constituency and Dependency). Alternative implementations exist for each of these modules. We instantiate the pipeline using a specific combination of module implementations which we found to be most accurate for the language of the legal texts in our context. For the lexical analysis modules (Tokenizer, Sentence Splitter, POS Tagger and Named-entity Recognizer), we use a language-specific framework called Lefff [50]. For constituency and dependency parsing, we use the Berkeley Parser [51] and the Malt Parser [52], respectively.

B. Evaluation

Case study description. The objective of our case study is to measure the accuracy of the extraction rules of Table II against a ground truth. To build a 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 IV. 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 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 III). 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 in the first round. This provides confidence that our extraction rules and markers have saturated. Due to space, we report the evaluation results for the 100+50=150 statements combined. To avoid biased conclusions, the results we report use the baseline set of concept markers, i.e., the same set with which the first evaluation round was performed.

The first author annotated the 150 statements used in the evaluation; the second author independently annotated 10% of these statements to examine reliability. We obtained $\kappa = 0.815$, suggesting "almost perfect agreement" [44]. 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 IV. 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* 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.

Analysis procedure. Each annotation has two parameters: a *type* and a *span*. The latter specifies where an annotation begins and where it ends in a statement. We evaluate the results of automated metadata extraction using the following notions:

- A computed annotation is a *perfect match* if it has the same type and span as some ground-truth annotation.
- A computed annotation is a *partial match* if its span has a non-empty intersection with some ground-truth annotation of the same type, but the spans are not identical.
- A computed annotation is *misclassified* if it is neither a perfect nor a partial match.
- A ground-truth annotation for which there is no perfect or partial match is considered as *missed*.

If the computed annotations are used as-is for analysis, the practical impact of partial matches, misclassifications and missed annotations would be as follows: A partial match is only an approximation of what is desired. The quality of the analysis depends on how good the approximation is, i.e., how well-aligned the span of the partial match is with the intersecting ground-truth annotation. Misclassifications can lead to unnecessary or unsound analysis. Missed annotations can lead to outcomes that are incomplete or even incorrect.

Existing evaluations of automated legal metadata extraction consider only the type parameter of the computed annotations. Our evaluation procedure presents an enhancement by considering annotation spans as well. Specifically, we define notions of precision and recall that penalize span inaccuracies in partial matches. The rationale is that analysts either need to rectify such inaccuracies before using the computed annotations, or take some corrective action during their analysis. In either case, additional manual effort will be incurred.

We use the Jaccard index for assessing the quality of partial matches. Let g be an annotation from the ground truth, and let a be a computed annotation that is a partial match for g. Instead of counting a as a full (perfect) match, we count it as a fraction determined by the Jaccard index: $J(a, g) = [S(a) \cap S(g)]/[S(a) \cup S(g)]$. In this formula, S denotes the span function and $[\,]$ the length (in characters) of a text segment. To illustrate, consider Statement 3 of Fig. 1. Suppose an automated solution annotates "for an offense" as a *violation*. If we take the annotations in Fig. 1 as the ground truth, the Jaccard index for the computed annotation is: ["an offense"]/["for an offense under this Act"] = 10/29 = 0.34. The computed annotation thus counts as 0.34 of a perfect match.

Penalizing partial matches using the Jaccard index is likely to be pessimistic. Span inaccuracies may, in practice, have less impact on manual effort than suggested by the Jaccard index. Further empirical studies would have to be carried out to measure with certainty the level of effort that analysts incur over dealing with span inaccuracies. In the meantime, we believe that the Jaccard index serves as a useful, albeit conservative, measure for the quality of annotation spans.

Results. Our evaluation results are presented in columns 3 through 9 of Table IV. For each legal concept (metadata type), we provide the number of perfect matches, partial matches, misclassified annotations, missed annotations, and precision and recall. Each perfect match counts as one true positive (TP). Each partial match counts as a fraction of a TP, calculated by the Jaccard index (explained above). Each misclassified annotation counts as a false positive (FP) for the metadata type it appears in front of. Each missed annotation counts as one false negative (FN).

Precision is computed as $|\text{TP}|_w/(|\text{TP}| + |\text{FP}|)$ and recall as $|\text{TP}|_w/(|\text{TP}| + |\text{FN}|)$. $|\text{TP}|_w$, which is $\leq |\text{TP}|$, is the total number of TPs, with each partial match individually weighted by the Jaccard index. The Jaccard index for a perfect match is one. The final row in Table IV shows the overall results. Note that the overall precision and recall scores are computed over all the annotations; these are not the averages of the precision and recall scores for the individual metadata types.

In summary, out of the 1100 computed annotations, 873 (79.4%) are perfect matches, 196 (17.8%) are partial matches, and 31 (2.8%) are misclassifications. There are 58 ground-truth annotations (5.1%) that the extraction rules miss, due to either misclassification or being unable to classify. We obtain an overall weighted precision of 87.4% and an overall weighted recall of 85.5% for the concepts covered by our extraction rules. Without penalizing for partial coverage of annotation spans, we obtain an overall precision of 97.2% and overall recall of 94.9% (not shown in Table IV). 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.

Given the complexity of correctly delineating annotation spans through automation, our weighted precision and recall scores are promising. The fact that the automatically-identified spans are fully correct in 79.4% of the cases provides confidence about the accuracy of constituency and dependency parsing. While 17.8% of the matches are partial, the penalties for partial span coverage decrease precision by only 9.8% and recall by only 9.4%. Indeed, the average Jaccard index for the partial matches is 0.46 (SD=0.29). This indicates that the partial matches have considerable overlap with the desired annotations. We thus do not anticipate the amount of manual effort required for adjusting the annotation spans to be too high.

TABLE IV
STATISTICS FOR AUTOMATED SEMANTIC METADATA EXTRACTION

Legal Concept	Ground Truth	Results of Automatic Metadata Extraction					Accuracy	
		Extracted	Perfect Match (TP)	Partial Match (TP)	Misclassified (FP)	Missed (FN)	Precision (%)	Recall (%)
Action	157	157	91	64	2	2	84.1	84.1
Actor	138	133	110	19	4	9	87.9	84.7
Artifact	252	241	182	56	3	14	82.3	78.9
Condition	172	179	148	18	13	6	88.1	91.7
Constraint	3	--	--	--	--	--	N/A	N/A
Exception	12	12	9	1	2	2	77.1	77.1
Location	35	34	27	7	0	1	84.3	82.3
Modality	80	81	74	4	3	2	93.5	94.7
Reason	23	22	21	1	0	1	98.0	93.7
Reference	72	--	--	--	--	--	N/A	N/A
Sanction	29	25	23	2	0	4	95.9	82.7
Situation	150	140	121	16	3	13	90.1	85.0
Time	67	63	56	7	0	4	94.3	88.7
Violation	12	13	11	1	1	0	87.3	94.6
Total	**1202***	**1100**	**873**	**196**	**31**	**58**	**87.4**	**85.5**

*We exclude from our evaluation *contraints* and *references* as noted in the text: we do not have extraction rules for *constraints*; detecting *(cross-)references* is outside the scope of this paper.

To determine the root causes for the automation inaccuracies observed, we analyzed all the misclassified annotations, missed annotations, and partial matches. Of the 31 misclassifications in Table IV, 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 an illness, 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 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 detect all the possibilities. If we do not count these 25 cases as FNs, our overall recall increases to 87.2% (precision remains unaffected). Among the remaining 33 missed annotations, 26 are due to misclassifications, discussed earlier. Five missed annotations result from distinct ground-truth annotations for which our rules produce only a single annotation (intersecting with both ground-truth annotations). Each of these five cases leads to one partial match and one missing annotation. The last two missed annotations are caused by constituency parsing errors discussed later.

As for the 196 partial matches, 21 (10.7%) are due to granularity differences between the computed and the ground-truth annotations. In other words, the computed annotations either fully cover or are fully covered by the ground-truth annotations. Another 21 (10.7%) partial matches are due to limitations in our rules. Among these, 15 are due to missing concept markers, and six due to our rule for the *situation* concept being too restrictive. The remaining 154 partial matches (78.6%) are due to constituency or dependency parsing errors.

As indicated by the discussions above, several of the automation inaccuracies, notably the partial matches, are caused by NLP errors. We observed that these errors stem primarily from subordination, coordination, and prepositional phrase attachments. Constituency parsers do not always connect such attachments to the correct node in the parse tree. Similarly, such attachments can mislead dependency parsers into inferring incorrect types for the dependency links. The limitations of constituency and dependency parsing in dealing with subordination, coordination, and prepositional phrase attachments are well-known [53], [54]. Despite these limitations, our good overall accuracy results help increase confidence that these advanced NLP techniques have matured enough to be applicable to legal texts. Further studies are nevertheless essential to more conclusively assess this claim.

As a final remark, we note that some recent strands of RE research, e.g., Quirchmayr et al. [36], offer useful domain-specific heuristics for working around NLP errors. Developing such heuristics for legal texts requires further investigation.

VI. Threats to Validity

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

Internal validity. A potential threat to internal validity is that the authors 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 [15]. This is thus open to scrutiny.

Another potential internal validity threat is that the coding in both the qualitative study of Section IV and the case study of Section V was done by the authors. Since traffic laws are intuitive and one of the authors (last author) is a legal expert, we found the risk of misinterpretation during coding to be low. 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) to minimize the influence of the extraction rules on the construction of the ground truth in our case study, we did not apply our implementation to the legal statements in the ground truth until coding was completed; (4) we assessed the reliability of the coding results by measuring interrater agreement over 10% of the coded statements.

External validity. The nuanced nature of legal texts often necessitates that research on legal requirements be based upon qualitative results obtained in specific contexts. A qualitative study with a scope as limited as ours makes it difficult to address external validity with sufficient rigor. Further studies that cover a variety of legal domains thus remain essential for ascertaining the completeness and 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 II 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. Second, as we argued while discussing the concept markers of Table III, most of the marker sets are either systematically extractable from existing lexicons, or expected to saturate quickly due to the limited linguistic variations possible. As a result, we anticipate that our markers should be reasonably easy to adapt to other legal domains, noting that the markers are necessarily language-dependent and do not carry over from one language to another.

Another aspect of external validity concerns our evaluation of automation accuracy in Section V, and more specifically, whether the accuracy levels observed would generalize. To this end, we note that traffic laws are very versatile and cover a large variety of topics. The sampling frame for both our qualitative study and our evaluation is the entire set of traffic laws in effect (comprised of $\approx$12000 statements, as noted in Section IV). The 150 statements in our evaluation ground truth are thus unlikely to be too similar to the 200 statements in our qualitative study, given the sampling frame being so large. Although not a replacement for additional case studies, the large sampling frame helps mitigate external validity threats.

VII. Conclusion

Metadata about the semantics of legal statements is an important enabler for legal requirements analysis. In this paper, we first described an attempt at reconciling the different types of semantic legal metadata proposed in the RE literature. We then derived, through a qualitative study of traffic laws, extraction rules for the reconciled metadata types. Our rules are based on natural language processing, and more specifically, constituency and dependency parsing. Finally, we evaluated our extraction rules via a case study. The results are promising. Depending on whether a penalty is levied on annotation span inaccuracies or not, we obtain a precision between 87.4% and 97.2%, and a recall between 85.5% and 94.9%.

In the future, we plan to more thoroughly examine the completeness and generalizability of our extraction rules by conducting additional studies. We would further like to perform user studies in realistic settings to determine the practical utility of automation for legal metadata extraction.

Acknowledgments. Supported by the Luxembourg National Research Fund (FNR) under grants PUBLIC2-17/IS/11801776 and PoC16/11554296.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] A. Siena, J. Mylopoulos, A. Perini, and A. Susi, "Designing law-compliant software requirements," in *Proceedings of the 28th International Conference on Conceptual Modeling (ER'09)*, 2009, pp. 472–486.

[9] 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.

[10] 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 the ACM International Health Informatics Symposium (IHI'10)*, 2010, pp. 629–636.

[11] N. Zeni, E. A. Seid, P. Engiel, S. Ingolfo, and J. Mylopoulos, "Building large models of law with NómosT," in *Proceedings of the 35th International Conference on Conceptual Modeling (ER'16)*, 2016, pp. 233–247.

[12] 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 the 27th International Conference on Conceptual Modeling (ER'08)*, 2008, pp. 154–168.

[13] 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.

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

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

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

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] S. Ghanavati, D. Amyot, and A. Rifaut, "Legal goal-oriented requirement language (legal GRL) for modeling regulations," in *Proceedings of the 6th International Workshop on Modeling in Software Engineering (MISE'14)*, 2014, pp. 1–6.

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

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

[34] ——, "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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] F. DellOrletta, S. Marchi, S. Montemagni, B. Plank, and G. Venturi, "The splet2012 shared task on dependency parsing of legal texts," in *the 4th Workshop on Semantic Processing of Legal Texts (SPLeT'12)*, 2012, pp. 42–51.

[42] J. Saldaña, *The Coding Manual for Qualitative Researchers*. Sage, 2015.
[43] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, 1960.
[44] J. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.
[45] 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.
[46] N. Sannier, M. Adedjouma, M. Sabetzadeh, and L. C. Briand, "Automated classification of legal cross references based on semantic intent," in *Proceedings of the 22nd International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'16)*, 2016, pp. 119–134.
[47] 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.
[48] "Wiktionary," https://fr.wiktionary.org/.
[49] Princeton University, "About WordNet," http://wordnet.princeton.edu, 2010.
[50] 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.
[51] 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.
[52] 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.
[53] 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.
[54] 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.