SN Computer Science (2022) 3:345
https://doi.org/10.1007/s42979-022-01228-4

REVIEW ARTICLE

# From Legal Contracts to Formal Specifications: A Systematic Literature Review

Michele Soavi[1] · Nicola Zeni[1] · John Mylopoulos[2] · Luisa Mich[1]

Received: 2 September 2021 / Accepted: 21 May 2022

**Abstract**
The opportunity to automate and monitor the execution of legal contracts is gaining increasing interest in Business and Academia, thanks to the advent of smart contracts, blockchain technologies, and the Internet of Things. A critical issue in developing smart contract systems is the formalization of legal contracts, which are traditionally expressed in natural language with all the pitfalls that this entails. This paper presents a systematic literature review of papers for the main steps related to the transformation of a legal contract expressed in natural language into a formal specification. Key research studies have been identified, classified, and analyzed according to a four-step transformation process: (a) structural and semantic annotation to identify legal concepts in text, (b) identification of relationships among concepts, (c) contract domain modeling, and (d) generation of a formal specification. Each one of these steps poses serious research challenges that have been the subject of research for decades. The systematic review offers an overview of the most relevant research efforts undertaken to address each step and identifies promising approaches, best practices, and existing gaps in the literature.

**Keywords** Legal contract · Semantic annotation · Conceptual model · Systematic literature review · Requirement · Specification

## Introduction

The advent of software that partially monitors, automates, and controls the execution of legal contracts has gained increasing interest in Academia, Government, and Industry, thanks to the advent of smart contracts. Such software systems have become possible thanks to blockchain and Internet-of-Things (IoT) technologies [1]. A first step towards a systematic software engineering (SE) process for building such software systems is to translate automatically or semi-automatically legal contract text written in natural language (NL) into formal specifications that precisely define the terms and conditions (requirements) that a smart contract needs to monitor and control. The use of a formal specification is required to ensure that the smart contract is executed as intended by contracting parties [1]. Moreover, a formal specification entails that smart contract developers construct their system on the basis of an unambiguous account of the legal contract to be monitored, automated, and controlled [2].

This study aims to conduct a systematic literature review (SLR) on the process of translating a legal contract into a formal specification. Existing research tends to cover only part of the transformation process with a limited number of studies dealing with the full process. To make the review more focused, we have decomposed the translation problem into four sub-problems: (a) Structural and semantic annotation of legal text based on a legal ontology; (b) Discovery of relationships for concepts identified and annotated in step (a); (c) Formalization of terms used in the NL text into a domain model; (d) Generation of formal expressions that capture the terms and conditions of the contract. This refinement of the translation problem is intended to facilitate the discovery of relevant works in the literature that deal with one or more of the sub-problems, rather that focus only on studies that tackle the full translation problem.

Let us illustrate the four sub-problems with an example. Consider a simple sales contract between a meat producer

✉ Michele Soavi
michele.soavi@unitn.it

1 Department of Industrial Engineering, University of Trento, Via Sommarive 14, 38123 Povo, TN, Italy

2 School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Ave, Ottawa, ON K1N 6N5, Canada

Published online: 21 June 2022

**Table 1** A sale contract

| |
|---|
| O1: [Seller] shall deliver [qnt] quantity of meat of AAA quality to the warehouse of the [Buyer] before [delD] date (Delivery obligation) |
| O2: [Seller] shall ensure that the meat is transported in accordance with meat transportation standards (Transportation obligation) |
| O3: [Buyer] shall pay [price] amount within a week from the date of delivery (Payment obligation) |
| P1: If delivery is late, [Buyer] can charge [Seller] {delay-in-days} * $1000 as late fee (LateF power) |

**Table 2** Annotated sale contract

| |
|---|
| <\obl <\role [Seller]> shall deliver [qnt] quantity of <\asset meat> of AAA quality to the warehouse of the <\role [Buyer]> before [delD] date> |
| <\obl <\role [Seller]> shall ensure that the meat is transported in accordance with meat transportation standards> |
| <\obl <role [Buyer]> shall pay [price] amount within a week from the date of delivery> |
| <\powr If delivery is late, <\role [Buyer]> can charge <\role [Seller]> {delay-in-days} * $1000 as late fee> |

**Table 3** A conceptual model for the sale contract

| |
|---|
| O1: T, [Seller], [Buyer], T, 'deliver [qnt] quantity of meat of AAA quality to the warehouse of the [Buyer] before [delD] date' |
| O2: T, [Seller], [Buyer], 'meat is transported', 'meat is transported in accordance with meat transportation standards |
| O3: T, [Buyer], [Seller], T, 'pay [price] amount within a week from the date of delivery' |
| P1: 'If delivery is late', [Buyer], [Seller], 'pay {delay-in-days} * $1000 as late fee' |

and a supermarket chain that has clauses, as shown in Table 1.

Here, terms in square brackets [...] identify parameters of the contract, to be determined for each contract execution, while wiggly brackets {...} define quantities to be determined at execution time. The semantic and structural annotation step identifies structurally three *obligations* and a *power*. Powers are rights contract parties have to cancel, suspend, or create new obligations. Moreover, the contract identifies *roles* of parties, namely [Buyer] and [Seller], as well as *asset* meat. Accordingly, the output of a tool that addresses the first sub-problem may look, as shown in Table 2.

A semantic annotation requires a common vocabulary (ontology) for legal contracts [3]. A structural annotation requires a grammar for the syntactic structure of legal contracts [4].

To address the second sub-problem, we need to find relationships for each one of the concepts identified in step one. For example, each obligation must have a *debtor* who is obliged to fulfill it, and a *creditor* (beneficiary). The debtors and creditors of O1 and O2 are obviously Seller and Buyer, respectively, while in O3 roles are reversed. Note that in O2 and O3, there is no mention of a creditor, so this has to be inferred from context. Finally, for the power the creditor is the Buyer and the debtor the Seller. In addition, each obligation/power must have a *trigger* that initiates it, an *antecedent* that serves as precondition, and a *consequent* that signals successful completion of the obligation/power. P1 has a trigger 'If delivery is late', while others take a trigger 'true', indicated by T, and are initiated when contract execution starts. O1 has antecedent 'true' and consequent 'deliver [qnt] quantity of meat of AAA quality to the warehouse of the <\role [Buyer]> before [delD] date', while O2 has antecedent 'while meat is transported' and consequent 'meat will be transported in accordance with meat transportation standards. The output of the second step is a conceptual model of concepts and relationships, where for each obligation, we list trigger, debtor, creditor, antecedent, consequent, and for each power, we list trigger, debtor, creditor, antecedent, and consequent. The result is included in Table 3.

The third sub-problem concerns formalizing terms used in the contract in relation to the legal contract ontology used in step (a). In particular, 'meat' is an instance of class BeefC, representing a portion of beef, which is a specialization of MeatC, which has as instances quantities of meat, including beef, chicken, etc. In turn, MeatC is a specialization of FoodC, which is a specialization of Asset. Asset is one of the concepts in the contract ontology. Along the same lines, 'warehouse' refers to an address that is the target of the delivery, so it can be formalized as attribute warehAddr of class BuyerC, which is a specialization of Role, another class in the contract ontology. The result of this step is a *domain model* for the contract being formalized. Formalization of terms is intended to eliminate ambiguity. For example, 'meat' is multiply ambiguous (could it be chicken or lamb?). Likewise, does 'delivery'

**Table 4** Definitions used in the study

| | |
|---|---|
| **Natural language (NL)** | Any language spoken by humans (e.g., English and Italian) |
| **Natural language processing (NLP)** | Processing and analysis of NL text based on linguistic analyses, such as syntactic, semantic, or pragmatic |
| **Legal contract or simply Contract** | An agreement between two or more parties, involving obligations and powers meant to be legally binding and effecting changes to assets |
| **Legal document** | A document concerning a legal matter, such as contract, law, regulation, article of incorporation, or testament |
| **Legal ontology** | An ontology consisting of concepts and relationships for conceptualizing legal documents, such as the Legal Core Ontology (LCO) [6] |
| **Specification language** | A language used to describe *what* an artifact does, rather than *how* it does it [7]. Specification languages for contracts include the Business Contract Language (BCL) [7] and Symboleo [5] |
| **Requirement** | A functional capability or quality constraint wanted by a stakeholder for an artifact-to-be [8]. In the context of the systematic review, requirements for a legal contract are derived from NL text, the required lifecycle of a contract, and the technical requirements arising from its implementation platform and technologies used (e.g., IoT, blockchain) |
| **Semantic annotation** | The process of adding semantic annotations to text to mark instances of domain concepts [9]. The process uses a markup language, such as XML |
| **Conceptual model** | A directed labeled graph where nodes represent concepts or their instances, and links represent relationships |

in ‘‘If delivery is late’’ refer to the delivery action, or more likely to the event ‘delivered’. The formalization is based on an ontology for contracts that includes primitive concepts such as Role, Obligation, Power, Asset, Event, etc. and the relationships mentioned above, such as debtor, creditor, antecedent, and consequent.

The fourth sub-problem concerns translating NL expressions such as ‘meat is transported’ into expressions in a formal specification language. Symboleo [5] is such a language where true/false statements are defined in terms of *events*, instantaneous happenings, and *situations*, states-of-affairs that occur over a period of time. For example, the trigger of P1 “If delivery is late” is translated to “happens(delivered(Seller, meat, Buyer.delAddr), t) and (t after delD)” where delivered(…) is an event that happens at time t. The antecedent of O2 is translated to “occurs(transport(meat), int)”, which says that the situation transport(meat) has occurred during interval int, while the consequent is translated into “occurs(MTS(meat), int)”, which says that the situation meat-transported according to transportation standards (MTS) occurred during int, the interval of the transportation situation.

The SLR provides three main contributions. First, it proposes a process for translating a contract in natural language into a formal specification and uses it to define research questions that guide the review process. Second, it identifies the most significant papers for each research question that can guide researchers interested in the topic. Finally, it provides a direction for future research by pointing out main challenges and open problems that would require further research. The importance of the SLR is underlined by the increasing number of recent publications covering aspects of the translation process, as well as the increasing interest in smart contract applications, Also, the lack of systematic literature reviews on the topic, as well as related topics.

The rest of the paper is structured as follows. “Preliminaries” defines the terminology adapted in this review. “Research questions and scope” defines research questions to be answered by the SLR and its scope, that identifies the works included in the study. “Classification scheme” describes the classification scheme adopted for selected papers, while “Search and screening of selected papers” describes the search. “Classification of selected papers” classifies the papers identified based on research questions, and a further taxonomy. “Synopsis of the papers” describes the papers that have been analyzed, whereas the findings for each research question are summarized in “Summary of findings”. “Threats to validity” defines threats to validity for this study, while “Related work” discusses related work. Finally, “Conclusions” concludes and discusses future work.

## Preliminaries

As already hinted in the introduction, throughout this study, we use terms, such as ‘legal contract’, ‘legal document’, etc. These terms are elaborated in Table 4. Although the focus of the SLR concerns translation of contracts, legal documents have been included in the study, as well, since they concern similar concepts and their translation into a formal specification involves similar steps. Moreover, the limited number of papers focused on legal contracts suggested the need to widen the scope to also include legal documents.

Table 5 Research questions

| | |
|---|---|
| **RQ0** | What are the main approaches for translating legal documents into formal specifications? |
| **RQ1** | What legal ontologies have been used for the translation? |
| **RQ2** | What annotation approaches are used for semantic annotation of legal text? |
| **RQ3** | What are main approaches for mining relationships from annotated text? |
| **RQ4** | What are main techniques for formalizing NL terms into a domain model? |
| **RQ5** | What kinds of techniques have been studied for translating NL expressions into formal ones for legal documents? |

Table 6 Search query for each RQ

| | |
|---|---|
| **SQ-RQ0** | ("translation" OR "transformation" OR "from" OR "modelling" OR "formal specification") AND ("legal" AND ("text" OR "document") AND ("logic" OR "formal" OR "specification" OR "expression" OR "formalisation")) |
| **SQ-RQ1** | ("legal" OR "law" OR "contract" OR "regulatory" OR "obligation") AND ("ontology") |
| **SQ-RQ2** | ("semantic" AND "annotation") AND ("legal" OR "contract") AND ("text" OR "document") |
| **SQ-RQ3** | ("relation" OR "relationship" OR "concept") AND ("model" OR "extract" OR "mine" OR "identify") AND ("legal" AND ("text" OR "document")) |
| **SQ-RQ4** | ("model" OR "extract" OR "mine") AND ("domain" OR "conceptual model" OR "conceptual models" OR "conceptual modelling" OR "template") AND ("legal" AND ("text" OR "document")) |
| **SQ-RQ5** | ("from" OR "formalization" OR "translation" OR "transformation") AND ("legal" AND ("text" OR "document")) AND ("logic" OR "temporal" OR "formal" OR "specification" OR "expression") AND ("technique" OR "framework" OR "tool") |

The SLR follows the general methodology proposed in [10] and made more concrete in [11] and [12] for a Requirements Engineering topic. The methodology has as follows:

**Step 1**: Define research questions (RQs) that the SLR is intended to answer, as well as the scope of the SLR. Outcome: RQs, document libraries to be used, inclusion and exclusion criteria, search method (keyword search, snowballing, etc.), queries for each RQ

**Step 2**: Determine classification scheme for paper types, topics, and keywords for each scheme. Outcome: classification scheme

**Step 3**: Conduct search and screen retrieved papers with respect to relevance and inclusion/exclusion criteria. Outcome: selected papers

**Step 4**: Classify selected papers. Outcome: classification of selected papers

**Step 5**: Answer research questions on the basis of the classification. Outcome: SLR results.

## Research Questions and Scope

The RQs identified concern the translation of NL text into a formal specification for legal documents, as well as the four sub-problems discussed in "Introduction". The RQs represent the decomposition of the general process that we established (RQ0), aiming to understand how the different steps have been executed (RQ2 to RQ5). Since the topic of legal ontologies underlies several research questions, it has been added as a sixth research question (RQ1). For each of them, the SLR is intended to mine main approaches and research results that together characterize the state-of-the-art (Table 5).

The scope of the study includes the publication datasets of ACM, IEEE, and Springer where the vast majority of publications for the subject of the SLR have appeared. Beyond this, scope is determined by the RQs and is defined by search queries (SQ), one for each RQ (Table 6), as well as inclusion criteria.

Inclusion criteria are provided in Table 7.

## Classification Scheme

The classification scheme adopted for selected papers uses two criteria: (a) Which research questions does each paper address; (b) The type of each paper, e.g., proposal, implementation, evaluation, etc. The former criterion has already been discussed in the introduction. For the latter, we adopt the classification scheme used in [11], with amendments, since the subject of our review is substantially different from that in [11] (Table 8). This criterion allows us to classify papers according to the nature of their contribution(s).

The categories in each classification are not mutually exclusive. For example, a publication may tackle at the same time the sub-problems of semantic annotation and relationship mining, or may include a proposal, an

Table 7 Inclusion criteria

| Inclusion criteria |
|---|
| Describes a process/framework/tool to generate specifications from NL for a legal document, or deals with one or more of the four sub-problems, and |
| The paper is published in English, and |
| The legal document or contract analyzed in the paper is in a Latin alphabet (e.g., papers dealing with semantic annotation of Japanese or Chinese text have been excluded), and |
| Published in an international conference, journal, or book or published in a workshop or regional conference and has more than five citations, and |
| Has at least 10 citations from Google Scholar, for works published before 2015, and |
| Covers at least one category for each classification scheme criterion adopted in this SLR |

Table 8 Paper type criterion

| | |
|---|---|
| **Proposal** | Any paper that proposes a new approach for dealing with any of the RQs; evaluations of proposals, including case studies, experiments, or experience reports would not count as proposals |
| **Formalization** | Includes papers that offer axioms expressed in a formal logical language; pseudocode does not count as formalization |
| **Meta-study** | Papers that provide a significant overview of existing work on a topic relevant to the SLR. Examples include surveys, reviews, and sometimes vision papers; the category is reserved for papers that emphasize some forms of analysis |
| **Implementation** | Papers that present the development of a tool or implementation that facilitates the contribution of the work, no credit is given for incomplete implementations; however, the tool does not have to be implemented by the authors |
| **Evaluation-adequacy** | Includes papers that apply a proposal to a benchmark, case study, or illustrative example; whether an application of a proposal is a case study or an illustrative example depends upon depth and reality of the application |
| **Evaluation-empirical** | Papers describing a controlled empirical study with human subjects, or a survey based on a questionnaire collecting answers from target groups |
| **Evaluation-scalability** | Papers in this category evaluate the scalability of an implemented tool with respect to computational resources (time, memory), input size, or human effort |

implementation, and an empirical evaluation. Of course, it is not expected from any publication to cover all or most categories for either criterion.

## Search and Screening of Selected Papers

The systematic search based on the search queries in Table 6 took place between October 2020 and June 2021. Initially, relevant papers were identified by searching the ACM, Springer, and IEEE scholarly repositories with our adopted search queries. In a first attempt, we implemented RQ queries into the search facilities provided for advanced research for each repository. However, the impossibility to consistently apply the search queries for each database—arising from significant differences in the different search fields for advanced research—led to the decision to rely on ACM Guide to Computing Literature that allows the identification of papers published as well from Springer and IEEE. Moreover, we decided against using Google Scholar, because it does not support searching over abstracts only. The RQ search queries have been applied to the abstracts for all queries, except RQ1 on legal ontologies. The application of the queries for RQ1 to the abstracts resulted in a disproportionated number of results compared to the results of the other RQs. To obtain a reasonable result for RQ1, the search questions were applied to the title and including works with at least one citation. The results of the SQs were analyzed by the three expert authors to subsequently apply a variant to the SLR process to make sure that all relevant papers were included. As the result of the analysis underlined the lack of important publications dealing with one or more RQs, as well as the recent proceedings of the main conferences in the domain (i.e., ER, PoEM, RE, and REFSQ) from 2019 to 2021 which may not have been included yet in the ACM repository. As such, SQs have been refined and new papers have been added to the total number of papers being analyzed, also considering they have been published very recently. The refinement of the SQs and the indications of the three expert authors allowed the identification of the following works to ensure that main approaches to formalize specification languages for contracts were included [7, 39, 53, 72, 99], legal ontologies [6, 48, 54], and reviews [27, 77]. Concerning the proceedings of recent conferences, the works in [2, 5, 40, 80, 82, 87, 89, 90] have been included. The outcome of applying SQs for all six RQs and the refinement process led to the identification of a total of 447 publications. Among those, 60 papers were eliminated as duplicates. The full text of the remaining papers was considered for relevance to at least one of the RQs. As a result, a total of 233 papers were eliminated, along with another 13 papers that were analyzing legal documents or contracts not written in a Latin alphabet (e.g., Japanese, Chinese), while 8 were eliminated for being extended abstracts. Out of the remaining papers, 40 were excluded for not having at least 10 citations, even though they were published before 2015.

## Classification of Selected Papers

The remaining papers were analyzed and classified under one or more RQs as well as the classification scheme of Table 8. To ensure that the classification scheme was applied objectively, 25 of the papers were classified by two persons and results were compared and discussed. Only minor differences were found in this exercise, mostly based on diverging interpretations of RQs.

Out of the 93 papers, the 59 papers with the most relevance according to our classification scheme—i.e., RQ, paper type—have been summarized in the following section, except for meta-studies which have been classified and summarized in the related work section. The selection filtered works that include implementation results rather than just a proposal, specific works instead of a general framework, also works with an impact, measured by citation statistics.

The classification of the 93 papers is presented in Table 9, where the references in bold concern the 59 papers selected and included in the description provided in Section number 7. Most of the papers are dealing with more than one RQ. Significant research has been performed for legal ontologies, annotation, and mining of relationships from legal text, with 28, 37, and 34 papers, respectively. Less work has been performed concerning the general process of deriving formal specifications from a legal text (21 papers) or translating NL expressions into formal specifications (19 papers). Few of the papers in the table include an actual implementation. Generally, papers include a case study or an explanatory example, although they propose less frequently an experimental evaluation of a tool. Seldom, the experimental study includes a systematic evaluation of scalability. Frequently, experimental studies concern annotation of legal text.

## Synopsis of the Papers

The present section describes the 59 papers and their contribution to one or more RQs. When a paper included in the SLR refers to many RQs, it has been included in the subsection referring to the RQ which is more relevant accounting for the content of the paper.

### RQ0: What are the Main Approaches for Translating Legal Documents into Formal Specifications?

Few works relate to the whole process of translating legal documents into formal specifications. Among them, Sharifi et al. propose [5] Symboleo, a specification language for legal contracts and shows through examples how to manually translate legal contracts into formal specifications. Contracts are represented as collections of obligations and powers and are about roles, who are their debtors and beneficiaries, and assets that change state, usually ownership. The validity of a formal specification can be checked, as demonstrated in subsequent work on Symboleo that presents the validation of two standard business contracts with a compliance checker [80]. Symboleo represents the most complete, detailed approach we identified in the SLR to generate formal contract specifications. The proposed approach can help understand conceptually the translation process and the main difficulties. However, little detail is provided about the translation process and the process is completely manual.

Similarly, Grosof proposes SweetDeal [49], a rule-based technique to represent business contracts automated along their lifecycle. Ontologies are represented as DAML + OIL, an extension of OWL (Ontology Web Language) that is frequently used in Semantic Web. A complete explanatory example on how to generate specifications is provided and explained. Similarly to [5], it could represent a reference to understand how NL is translated into a formal specification. However, the process is mostly manual, and it does not address the identification and annotation of legal concepts and relationships.

Hashmi proposes a manual methodology to extract legal requirements from text, and formalize them in the Event Calculus [52]. The proposal relies on IF–THEN rules and includes process aspects together with rule types (e.g., determinative or prescriptive). The assumption is that extracting the abstract structure of a legal document facilitates tracking the implications of business processes, tracing requirements, and checking for compliance, all issues frequently ignored in legal document analysis. The proposal relies on Ciceronian rhetorical loci, focusing on who, why, what, when, and where of business processes identified from real-life cases. IF–THEN rules make translation easier than for other formalisms, but they are not expressive enough to capture all the nuances of legal concepts. Differently from other approaches presented in the SLR, it forces the reader to formalize the specification with logic that is more easily translated into a programming language.

He et al. [53] propose a different approach where the level of formality is decreased to enhance understandability for a non-technical audience (e.g., lawyers, business analysts). The proposal is based on SPESC, a specification language for smart contracts developed with collaborative design in

**Table 9** Papers selected in SLR allocated for RQs and classification

| **Description and reference** | **Research question** | | | | | | **Classification** | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RQ0 | RQ1 | RQ2 | RQ3 | RQ4 | RQ5 | Proposal | Formali-zation | Meta-study | Implemen-tation | Evaluation-adequacy | Evaluation-empirical | Evaluation-scalability |
| Agnoloni2010 [13] | | X | X | X | | | X | | | | | | |
| Ajani2007 [14] | | | X | X | | | X | | | | X | | |
| Alsaadi2019 [15] | | | X | | | | X | | | | X | | |
| **Amardeilh2005** [16] | | X | X | | X | | | | | X | | X | |
| **Amato2008** [17] | | | | X | | | X | | | X | X | | |
| **Araujo2013** [18] | | | X | | | | X | | | X | | | |
| Ashley2009 [19] | | | | | | X | X | | | X | X | | |
| Ashley2013 [20] | | | X | X | | | | | | X | | X | |
| Azzopardi2018 [21] | X | | | | | X | X | X | | | X | | |
| **Barabucci2009** [22] | X | X | X | | | | | | | X | X | | |
| **Bench1997A** [23] | | X | | | | | | | | X | X | | |
| Bench1997B [24] | | X | | | | | X | | | | X | | |
| **Biagioli2005** [25] | | | X | | | | X | | | X | | X | |
| **Boella2009** [26] | | | | | | X | X | X | | | X | | |
| **Boella2014** [27] | X | | | | | | | | X | | | | |
| **Boer2008** [28] | | X | | | | | | | | X | X | | |
| Branting2019 [29] | | | X | | X | | | | | X | | X | |
| **Breaux2013** [30] | X | | | | | | | | | X | | X | |
| **Breuker2005** [31] | | X | | | X | | X | | X | | | X | |
| Bueno1999 [32] | | | X | | X | | X | | | | X | | |
| Capuano2014 [33] | | X | | X | | | | | | | | X | |
| Ceci2011 [34] | | | X | | X | | X | | | | X | | |
| **Chalkidis2017** [35] | | | X | | | | | | | X | | X | |
| Chieze2010 [36] | | | X | | | | X | | | X | | | |
| **Corcho2005** [37] | | X | | | | | | | | X | X | | |
| Despres2006 [38] | | X | | | | | X | X | | | | | |
| **Dwivedi2021** [39] | X | X | | | | X | | X | | X | X | | |
| **Fischbach2021** [40] | | | X | X | | | X | | | X | | X | |
| **Fornara2009** [41] | | X | | | | X | | X | | | X | | |
| **Francesconi2010** [42] | | X | | | X | | X | | | | X | | |
| **Gangemi2005** [43] | | X | | | | | | X | | | X | | |
| Gao2012 [44] | | | | X | | | | | | | X | | |
| **García2017** [45] | | | X | X | | | X | | | X | | X | |
| **Governatori2006** [7] | X | | | | | X | | X | | | X | | |
| **Governatori2016** [46] | | | X | | | | | | | X | | X | X |

**Table 9** (continued)

| **Description and reference** | **Research question** | | | | | | **Classification** | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RQ0 | RQ1 | RQ2 | RQ3 | RQ4 | RQ5 | Proposal | Formali-zation | Meta-study | Implemen-tation | Evaluation-adequacy | Evaluation-empirical | Evaluation-scalability |
| **Grabmair2015** [47] | | | X | | | | X | | | | X | | |
| **Griffo2015** [6] | | X | | | | | X | | | | X | | |
| **Griffo2017** [48] | | X | | X | | | | | | X | X | | |
| **Grosof2003** [49] | X | X | | | | X | | X | | X | X | | |
| **Grover2003** [50] | | | X | X | | | X | | | | X | | |
| Hasan2017 [51] | | X | | | | | | X | | X | X | | |
| **Hashmi2015** [52] | X | | | | | X | | X | | | X | | |
| **He2018** [53] | X | | | | | | | X | | | X | | |
| **Kabilan2003** [54] | | X | | X | | | | X | X | | X | | |
| Kayed2005 [55] | | X | | | | | | | | X | | X | X |
| **Kiyavitskaya2007** [9] | | | X | | | | | | | X | | X | |
| **Kiyavitskaya2008** [56] | | | X | X | | | | | | X | | X | X |
| **Konstantinou1993** [57] | | | | | | X | X | | | | X | | |
| Lagioia2017 [58] | X | | | | | | | X | | | | X | |
| Lagioia2019 [59] | | | X | X | | | | | | X | X | | |
| **Lame2005A** [60] | | | | X | X | | | | | X | | X | |
| Lame2005B [61] | | X | | X | | | X | | | X | X | | |
| Lau2003 [62] | | | | X | | | X | | | X | X | | |
| **Lau2005** [63] | | | | X | | | X | | | X | X | | |
| **Lee2006** [64] | | X | | X | X | | X | | | X | X | | |
| **Lesmo2009** [65] | | | X | | | | | | | X | X | | |
| Levy2010 [66] | | | X | | | | X | | | X | X | | |
| **Libal2019** [67] | | | | | | X | X | X | | | X | | |
| Liu2015 [68] | | | | X | | | X | | | | X | | |
| **Maxwell2009** [69] | | | | | | X | X | X | | | X | | |
| **Mazzei2009** [70] | | | X | | | | | | | X | | X | |
| **Moens2007** [71] | | | | X | | | | | | X | | X | |
| **Montazeri2011** [72] | X | | | | | X | | X | | X | X | | |
| Moulin1990 [73] | | | | X | | X | X | | | | X | | |
| Nadah2007 [74] | X | X | | | | | | | | X | X | | X |
| **Neill2017** [75] | | | X | | | | | | | X | | X | X |
| Osborn1999 [76] | | | X | | | | X | | | | | X | |
| **Otto2007** [77] | X | | | | | X | | | X | | | | |
| Palmirani2009 [78] | | | | X | X | | X | | | X | | X | X |

Table 9 (continued)

| Description and reference | Research question | | | | | | Classification | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RQ0 | RQ1 | RQ2 | RQ3 | RQ4 | RQ5 | Proposal | Formali-zation | Meta-study | Implemen-tation | Evaluation-adequacy | Evaluation-empirical | Evaluation-scalability |
| **Palmirani2011** [79] | X | | | | | X | X | X | | X | X | | |
| **Parvizimosaed2020** [80] | X | | | | | X | | X | | X | X | | |
| **Quaresma2010** [81] | | | X | | | | | | X | | | X | |
| **Rabinia2020** [82] | | | | X | | X | X | X | | | X | | |
| Rao2015 [83] | | | | X | | | X | | | X | X | | |
| Rodrigues2015 [84] | X | X | | | X | | X | | | | X | | |
| Roegiest2018 [85] | | | X | | | | X | | | X | | X | |
| **Saias2005** [86] | | X | | X | X | | | X | | X | X | | |
| **Sainani2020** [87] | | | X | | | | X | | | | | X | |
| Sannier2017 [88] | X | | | X | | | X | | | X | | X | X |
| **Sharifi2020** [5] | X | | | X | X | X | X | X | | | X | | |
| **Sleimi2019** [89] | | | X | X | | | X | | | X | X | | |
| **Sleimi2020** [90] | | | X | X | | | X | | | | X | | |
| **Soavi2020** [2] | X | | X | | | | | | | X | | X | |
| Tilbrook2013 [91] | | | | | | X | X | | | | X | | |
| **Valente2005** [92] | | X | | | | | | | X | | | | |
| Villata2020 [93] | X | | | | | | X | | | | X | | |
| **Völker2008** [94] | | | | X | X | | | | | X | X | | |
| Walker2017 [95] | | | X | | | | | | | | X | | |
| Weber2009 [96] | | | X | X | | | X | | | | X | | |
| **Wyner2010** [97] | | | X | | | | | | X | | | | |
| **Yan2006** [98] | | X | | | | | X | | | | X | | |
| **Zeni2015** [4] | | | X | X | | | | | | X | | X | X |
| **Zeni2018** [99] | X | | X | X | | | | X | | X | | X | |
| | 21 | 28 | 37 | 34 | 14 | 19 | 48 | 22 | 7 | 49 | 58 | 28 | 8 |

mind. Here, specifications are manually derived from text and are expressed in an NL-like language using an extended BNF grammar. SPESC specifications are more abstract than smart contract code and have a general structure consisting of parties, contract properties, terms, and data type definitions. Similarly to [52], the proposed approach potentially support a developer in deciding how to implement the formal specification and it is not limited to the how. A recent work [39] presents a process for generating a smart contract. A multi-tier ontology is proposed to support translation to a domain-specific representation using a Smart Legal Contract Markup Language (SLCML).

Breaux et al. [30] present another intermediary approach that focuses on traceability to express legal requirements semi-formally. This is accomplished through a computational requirements document expressed in a specification language for legal requirements (LRSL). Similarly to [53], the objective is to provide legal requirements that are freely available to policymakers, business analysts, and software developers. An automated parsing tool checks for syntax and semantic errors requirements in LRSL. The parser applies deontic annotations based on a set of heuristics and creates a model to identify mandatory requirements. Specifications can be exported to different formats, including HTML, GraphML, and XML to allow for different types of analysis. The approach could be complementary to other approaches proposed to guarantee the completeness of the formal specification. Finally, NómosT has been implemented with the objective to build models of law semi-automatically; first, the text of a law is annotated—relying on GaiusT, a semantic annotation tool [4]—and then, it generates a model [99]. NómosT uses Propositional Logic, a semi-formal language that lacks quantifiers, modal operators, and other features that would significantly limit the applicability of the approach to legal contracts. The approach supports well the initial steps of the translation process, but it is less helpful for the whole process compared to other approaches, such as [5] or [52].

## RQ1: What Legal Ontologies Have Been Used for the Translation?

The use of legal ontologies for modeling legal documents dates back to the 90s; see for example [23]. Two main foci have been identified, sometimes used together: (a) the application of an existing ontology to a legal text for purposes of modeling and analysis, and (b) the identification of an ontology from legal texts. Concerning the first problem, Gangemi [43] proposed an influential Core Legal Ontology (CLO) to support information systems dealing with legal matters. The ontology is an extension of DOLCE+, which in turn is an extension of DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering), a foundational ontology that is used with the JurWordNet lexicon. CLO has been applied to compare and verify compliance of norms and to support text-mining. The work in [43] has significant pedagogical content to support understanding how an ontology is built.

An ontology that has seen significant legal applications is the Unified Foundational Ontology (UFO) extended to develop UFO-L [6] focusing on rights, duties, no rights, and permissions. This work emphasizes the importance of basing legal ontologies on legal theories and foundational ontologies. It represents the conceptual basis on how to build an ontology, although it does not detail practical implementation requirements. UFO-L has been used for modeling contract in [48], where it has been explored to bridge the gap between two different types of approaches for contract representation. Some approaches, such as ArchiMate, offer an opaque representation (e.g., not revealing rights and obligations), whereas others are devoted to formal representation. A service contract ontology is presented together with the extension of ArchiMate to reflect the proposed contract ontology. The details provided in the case study support well the development of other case studies for different domains.

Other ontologies have been specifically created for legal contracts. Among the first ones, Kabilan [54] proposed an ontology to efficiently link business process and contract management, improve business practices, and create alignment with the expectations of contracting parties. The proposed ontology is represented in UML and DAML, and consists of three levels: upper, domain, and template. The template level is intended to support modeling of specific types of contracts, such as rental or sales contracts.

Other approaches focus on providing tools and frameworks to build a legal ontology. Among them, Corcho et al. [37] propose a framework for building a legal ontology based on the METHONTOLOGY methodology and WebODE, a workbench for ontology engineering used in different domains. METHONTOLOGY is rooted in software and knowledge engineering methodologies. It supports legal professionals in building ontologies—by adapting a class taxonomy for the legal field—without significant involvement of knowledge engineers. The modeling process starts from the building of a glossary to represent concepts and relationships. The proposal has been applied for the development of several legal ontologies in Spain. Similarly to [48, 54], it provides a detailed case study to support the understanding of the process.

Yan et al. [98] underline the need for semantic information to be able to automatically execute a contractual agreement. The authors use OWL to formalize concepts and relationships and the Protégé-2000 tool for the implementation. Another approach aims at facilitating the management and representation of legal documents in XML [28]. A Legal Knowledge Interchange Format (LKIF) is provided as a reusable and extensible core ontology that also represents an interchange format to use for computer implementation and contract management. The ontology is based on a Description Logic. Despite the significant level of formality provided by the ontology and the MetaLex XML standard, LKIF, similarly to [98], appears to be better suited for ontological analysis and less as a reusable ontology for contracts.

### RQ2: What Annotation Approaches are Used for Semantic Annotation of Legal Text?

Semantic annotation of legal text represents the most recurring objective identified in the SLR relative to the problem-at-hand. Most proposals rely mainly on NLP and Machine Learning (ML) techniques. The problem of semantically annotating unrestricted NL text presents significant technical difficulties. For specific domains, such as the legal one, where specialized and less ambiguous language is used, the semantic annotation problem is more manageable, with encouraging results, see for example [65] below.

An approach adopted in several works relies on grammar rules to annotate legal text. For example, Kiyavitskaya et al. [9] propose Cerno, a tool for semi-automatically generating annotations from regulations using a domain ontology, and patterns of lexical indicators for each concept of the ontology. In an experimental evaluation, it was shown that Cerno slightly increased the quality of annotation while decreasing substantially annotation times for human annotators. In follow-up work, in [56], text is annotated to identify legal concepts (such as actors, rights, obligations, etc.), and then, a semantic model is constructed from the annotation and transformed into a set of functional and non-functional requirements. The first steps of the process concerning semantic annotation rely on heuristics and a frame-based model to identify deontic terms that can be rewritten using a controlled NL. Along similar lines, Soavi et al. [2] build ContracT—a specialization of Cerno [9] and GaiusT [4]—to support human annotators in semantic and structural annotation of legal contract text. The tool is based on an ontology for contracts derived from UFO-L, and it has shown to improve the annotation process for concepts such as parties, assets, temporal conditions, whereas difficulties were encountered in annotating powers and obligations. The approaches in [2, 4, 9, 56] are generally based on the definition of a grammar for semantic annotation that has to be re-defined for different domains.

OWL is used in [18] to represent linguistic information; the approach relies on a parser of structural information that represents relationships between different chunks of text. Extraction rules are formalized in a Description Logic and capture syntactic and semantic information. Significant implicit knowledge entails lower practical usability. Lesmo et al. [65] annotate legal provisions—rights or obligations—to understand the implications arising from the amendment of laws. NLP techniques are used to generate a set of metadata to compactly describe the modifications. The process helps to identify sections of the provisions which have been modified; subsequently, syntactic analysis and semantic annotation are performed. The relationships among provisions are determined thanks to categories based on the words identified in the provisions (e.g., synonyms for deletion or replacement of a provision). The proposal has been evaluated with several laws with positive results for integration and substitution amendments, whereas deletions require further study. Similarly, IF–THEN rules are exploited by Mazzei et al. [70] to identify the semantic content of sentences in laws that imply a modification of an existing provision. The approach relies on the pairing of deep syntactic parsing with rule-based shallow semantic analysis. The process is enhanced with the annotation of metadata and identifies candidate locations of modificatory provisions. The approaches in [65, 70] are useful for understanding the implications of modifying legal text.

Rule-based approaches perform better when a constrained language limited to a specific domain is used, e.g., for sales contracts. Quaresma [81] proposes a mixed approach, based on linguistic information—most notably morphological and syntactic—and ML to extract information from legal texts. Top-level concepts, such as organizations and dates, are identified using a Support Vector Machine (SVM) classifier, whereas an NL parser is used for entity recognition. The approach, applicable to a very specific task in the annotation process, is applied to different languages. Results are encouraging for concept classifications and the identification of dates, mixed for locations, and poor for the identification of organizations and cross-references. Moreover, results differ depending upon the NL of the text. Among the most appreciated works and similarly relying on SVM, Biagioli et al. [25] classify provisions and extract arguments—a set of reasons supporting a certain point of view—with SVM classification and NLP techniques with promising results. Neill et al. [75] test the use of probabilistic tools to extract deontic modalities from legal text. To avoid ambiguity, logic is commonly used; however, logical rules do not allow the level of expressivity required in many domains such as the financial regulations. Therefore, the authors test a data-driven approach, to classify deontic modalities, relying on Artificial Neural Networks (ANN), as well as non-Neural Networks, which are briefly reviewed and tested. The approach shows encouraging results, particularly for the pre-trained ANN. Similarly to [81], the approaches proposed by [25, 75] apply to very specific tasks in the annotation process. A comparison of different approaches in information extraction relying on ML is performed by Sainani et al. [87] that extract requirements from large software engineering contracts. The aim is to automate the extraction and classification of such requirements to improve contract management for companies. The authors compare different ML approaches (such as SVM, Random Forest, and Naives Bayesian) to their approach based on Bidirectional Encoder Representations from Transformers (BERT), which reaches an f-score higher than 80%. The proposal is useful to understand the potential different uses of ML for information extraction. Chalkidis [35] explores how deep learning can support semantic extraction to identify contract concepts and structural elements. In the experiment, Bidirectional Long Short-Term Memory (BILSTM)—with a logistic regression layer and that operates on word, POS tag, and token embeddings—outperforms linear sliding windows classifiers, without the need for manually written rules. The approach is tested on a set of contracts with promising results and suggests the opportunity to be improved with further stacked layers. The approach is supported by the availability of 3500 English annotated contracts released by the authors.

The importance of understanding the semantics of legal text using ML approaches—the black box problem—has been tested with Legal Unstructured Information Management Architecture (LUIMA) [47]. LUIMA is a law-specific extraction tool for automatic annotation using ML for sentence annotation and reranking together with basic retrieval that relies on Apache Lucene. The system is based on UIMA—a framework used in different contexts (e.g., IBM Watson,[1] a question-answering computer system)—to prove that the pre-processing to identify semantics can outperform information retrieval processes that do not account for semantics. LUIMA is the most complete tool for semantic annotation using ML that has been identified in the SLR.

Finally, a few open-source tools have been proposed with a collaborative, holistic view and that offer detailed documentation, to manage legal documents using XML standards. As such tools are not performing semantic annotation, they should be mostly considered as a support for contract management after the annotation process has already been performed. Akoma Ntoso [22] supports the annotation process at three different layers: NL text, structure, and metadata. Similarly, LegalRuleML is used to verify compliance of business processes and legal norms through semantic analysis [46]. Similarly to Akoma Ntoso, a legal document is represented in three different layers: metadata, statements, and context. Metadata refers to information about the document, such as legal sources and temporal properties; statements are formal representations of the norms; context refers to the relationships in the document including metadata and statements.

[1] https://www.ibm.com/watson.

## RQ3: What are Main Approaches for Mining Relationships from Annotated Text?

The mining of relationships in text is accomplished through syntactic and contextual analysis. For example, the identification of a debtor relationship for obligation O1 in Table 3 is determined by looking for the subject of verb ‘shall deliver’, i.e., the Buyer, while the identification of the creditor for O3 on the same table is determined by noting that two roles were identified in the contract and Seller has already been assigned the role of debtor for O3, so it cannot also be the creditor.

Comparably to approaches discussed for other RQs, the use of templates has been adopted as an intermediate step to identify relationships and ease the generation of requirements. Sleimi et al. [90] emphasize that legal texts often omit relationships included in annotation ontology, as with O3 discussed above. Templates are intended to fill three main gaps: statements with no-counterpart, statements with a correlative, and statements with an implied statement. Different approaches for legal requirement templates are reviewed and NLP-based rules for the templates are defined. Such rules improve the performance of relationship mining algorithms. Similarly, Lee et al. [64] rely on templates to identify relationships and present a technique to extract, model and analyze security requirements written in NL. Their analysis is based on a Problem Domain Ontology (PDO), and it is applied manually with a checklist. Subsequently, PDO is applied to a template to extract relationships among requirements, and to increase the understandability of information available in different documents. The approach—tailored for security requirements but potentially useful for any legal document—is evaluated for adequacy, although it requires a time-consuming process. The approaches in [90] and [64] could be used complementarily to increase the quality of the identified requirements.

Other authors suggest identifying relationships to improve retrievability. Sleimi et al. [89] markup text to generate semantic annotation and build Resource Description Framework (RDF) triples—as a representation of a conceptual model—that are queried with SPARQL. The toolchain system, a set of complementary software components, is experimentally tested in an industrial environment for recall and precision by requirement analysts. The work suggests that the creation of a conceptual model of legal metadata could ease access to legal content. To manage contract content, Lau et al. [63] focus on retrievability to consolidate regulations—using a shallow parser—into an XML format; the authors rely on text-mining tools and manually defined rules to extract elements. The approach compares sections of text to identify relatedness by analyzing matching terms, features, and structure matches, relying on domain knowledge and legal corpus knowledge. A few limitations are highlighted as mismatches for phrases used in different contexts or using different terminologies. In [45], layout rules are applied to improve retrievability for structure elements using XML markup obtained with NLP tools, JAPE (Java Annotation Pattern Extraction), and GATE (General Architecture for Text Engineering), with the latter providing an open-source set of reusable algorithms and GUIs for NLP. JAPE and GATE are integrated into a tool named CLIEL (Commercial Law Information Extraction based on Layout). The system is tested on 97 commercial laws with different approaches: Layout Insensitive, Majority Sense Baseline, and the Layout Sensitive strategy proposed; the best results are obtained with the last one. Approaches based on retrievability generally support the identification of relationships, although such relationships still require to be inferred.

The identification of causal relationships in requirements text is considered by Fischbach et al. [40]. A tool-supported approach named CiRA (Causality detection in Requirement Artifact) is tested with regular expressions, ML and Deep Learning (DL) approaches, the last one using BERT, which obtains the best results. The approach is useful to identify cue phrases for causes, but the labelling for causality may refer as well to deduction; causality can be inverted, and several causality relationships may exist. The above ambiguities can lead to significant differences in identifying relationships and further ambiguities may need to be managed.

Relationships between chunks of legal texts have been explored to identify arguments in sections. Notably, Moens et al. [71] identify arguments using n-grams, adverbs, modals, couple of words, text statistics, punctuations, and keywords to annotate legal text and subsequently identify and classify them. The best results are obtained using a multinomial Bayes classifier and a maximum entropy model previously trained where training has a significant impact on performance. In a similar approach, the SUM project aims at identifying the relationships between parts of legal texts to automatically summarize legal documents [50]. The approach relies on NLP techniques together with a combination of a rule-based and statistical methods to identify the most relevant parts of the text to be summarized. The classifier supports structural analysis to identify parts of the text as candidates for the summary. The capability to summarize is tested for adequacy after having described different linguistic tools and statistical measures for relatedness. The approaches of [71] and [50] are helpful to identify relationships among different sections of legal contracts, but they do not deal with the identification of relationships among the concepts identified with semantic annotation (i.e., ontology).

## RQ4: What are Main Techniques for Formalizing Natural Language Terms into a Domain Model?

The formalization of NL terms into a domain model is frequently considered as a task in the process of generating a domain ontology from legal text. Among the papers that address this problem, Saias [86] relies on NLP techniques to extract such models defined in OWL using syntactic, semantic, pragmatic analyses—where information is inferred with an abductive inference mechanism—and first-order logic, leading to the identification of concepts and relationships. Relationships are identified with unsupervised ML techniques that aim at learning subcategories for heads (e.g., republic for the republic of Ireland) and modifiers (e.g., president of the republic). The methodology is based on a parser that uses a Constraint Grammar formalism transformed into XML markups and Prolog terms. However, the approach does not define what kinds of relationships exist among concepts. Another relevant approach relies on statistical measures based on similarity and relatedness [60]. After having extracted all the terms from a sample of French legislation, terms are divided into syntactical categories and analyzed to support the identification of semantic relationships (e.g., book, chapter, general provisions). Considering the reliance on statistical measures the approach may better work for large legal corpora. The approaches presented in [86] and [60] are well documented and may support the understanding of most of the processes required to create a domain model.

Amardeilh et al. [16] present a method for semi-automatically building a domain model from a contract by populating an existing ontology with a knowledge management tool. A conceptual tree is derived from the text to map the information extracted to a concept of the domain ontology. Subsequently, knowledge acquisition rules are extracted to perform the mapping between linguistic annotation and ontological concepts. The rules are tested on 36 reports and an average of 3 acquisition rules per concept is identified. The method is tested for precision and recall in the identification of topics, attributes, associations, and roles; results highlighted more difficulties in identifying attributes and roles. Differently, Amato et al. [17] rely on a simplified NL, based on laws that are codified into pre-defined structures, to propose a process that transforms a legal document into an ontology based on RDF. The NLP system translates a legal document into tuples for a relational database by relying on different ontological and linguistic knowledge levels. The process supports the identification of structural, lexical, and domain ontology elements. It is intended for the management of notary documents and has been experimentally tested over a collection of around 100 legal documents with encouraging results. The use of simplified NL may imply a decrease in semantic richness in the translation of a legal document into a domain model.

Other approaches focus on building domain models that can potentially be reused with different languages. Notably, Francesconi et al. [42] suggest an approach for knowledge acquisitions and ontology modelling based on an existing ontology that is refined for a specific legal document with NLP techniques. They focus on the existing relations between the two layers—lexical and ontological—to define multilingual ontological requirements. The challenge of adapting ontologies or domain models for different languages is considered also by [94] in TextToOnto using extensible languages processing frameworks, such as GATE. The open-source tool has been created to support legal experts in identifying legal ontologies from text. Despite the difficulties required by the translation in different languages, they may support interoperability and the adoption of ontologies and domain models.

## RQ5: What Kinds of Techniques Have Been Studied for Translating NL Expressions into Formal Ones for Legal Documents?

The translation of legal documents into formal expressions has received less attention than other RQs. Research on this RQ date as far back as 1993 [57], relying on Conceptual Graph formalism based on Sowa for knowledge representation. The use of formal logic expressive enough to represent legal contracts is a recurring challenge. Among the first attempts to translate NL expressions into a formal specification for a contract, Governatori [7] proposes the Business Contract Language (BCL) based on Propositional Deontic Logic. This work describes the process of deriving a formal system from contract provisions that account for the identification of ambiguities in a contract, determine the existence of missing or implied statements, and analyze the expected behaviours of the parties and existing relationships between parts of the contracts (e.g., clauses). In the formal system, a contract is represented as a set of deontic terms for obligations, prohibitions, and permissions. Other approaches focus on the pre-treatment of text. Montazeri [72] supports the automatic translation of a contract in NL into a formal language using a Grammatical Framework (GF). The contract is manually rewritten in structured English that can be automatically translated into a formal language. The GF has been implemented to define and manipulate grammars and to understand the implications of translating a contract into different languages. The formal language is based on deontic, dynamic, and temporal logic. Despite the significant manual effort required, the works in [7] and [72] offer a well-explained framework of reference for the generation of formal expressions. Libal [67] introduces a logical structure tool—based on deontic logic—called Normative Detachment Structure with Ideal Conditions. The logical structure is extracted from a manually normalized text that encompasses ideal normative statements, normative conditionals, and existing relationships. The work suggests that the logical representation of contrary-to-duty is consistent and reflects the logical independence of the components of text while avoiding complexity. The ability to derive actual and ideal obligations has been only preliminarily tested and requires further exploration.

Fornara [41] relies on a domain-independent ontology that can be used to generate specifications for open interaction systems and accounts for social commitments, temporal propositions, events, agents, roles, and norms. The proposal is to monitor the variation over times of such commitments using the Event Calculus. Different axioms for the temporal propositions are presented, together with an explanatory example for a contract. Differently from the other works presented in the section, this work focuses on the implications of time evolution regarding starting points and deadlines.

A structured approach to managing legal documents using RuleML has been proposed to facilitate the sharing of legal information between legal documents, business processes, and software [79]. RuleML has been extended with new modules to represent and model legal phrases including metadata. The approach is based on a Defeasible Logic and has been implemented, although it does not support the level of formality found in other approaches. The use of RuleML is an advantage of this work, as it relies on a well-tested and documenting tool.

Formal representations have been derived from requirements extracted from a legal text in [82] using a goal-oriented approach. The authors propose a method to model formal requirements based on Formal Legal_GRL (FLG) using the Goal-oriented Requirements Language (GRL). A logic-based approach is used to deal with modalities and conditionals in legal text. Legal requirements are extracted and annotated using deontic logic for obligations and permissions. Similarly, for legal requirements, Boella et al. [26] propose a logical framework for formal representation and modelling based on an extension of a Defeasible Logic to model extensive and restrictive interpretations. The running example suggests that amendments to Law may result in changes to the adopted ontology. Finally, Maxwell [69] proposes a methodology for extracting production rules from legal texts that generate a raw translation and refactors the rules to enhance understandability. The approaches of [82] and [69] focus on deriving formal requirements from legal text and are accordingly more general than the subject matter of this review.

A summary has been included in Table 10 detailing for each paper analyzed, the methodologies, tools, and resources adopted from the literature and proposed. The papers are

**Table 10** Table summary of methods, tools, and resources used and proposed by papers relevant to SLR

| RQ | Description and reference | Methodology proposed | Methodology used | Tools proposed | Tools used | Resources proposed | Resources used |
|---|---|---|---|---|---|---|---|
| RQ0 | Sharifi2020 [5] | Symboleo | – | Symboleo PC | – | Ontology for contracts | – |
| RQ0 | Parvizimosaed2020 [80] | – | Symboleo | Symboleo CC | Symboleo PC | – | Ontology for contracts |
| RQ0 | Grosof2003 [49] | SweetDeal | – | SLCP/RuleML | RuleML | – | DAML + OIL, OWL |
| RQ0 | Hashmi2015 [52] | Hashmi methodology | – | – | – | – | CMF |
| RQ0 | He2018 [53] | – | – | – | – | SPESC formal language | Solidity |
| RQ0 | Dwivedi2021 [39] | SCLML | – | – | Protégé, HermiT-tool reasoner, LiquidStudio | SCL ontology | XML, Solidity |
| RQ0 | Breaux2013 [30] | Breaux methodology | – | Breaux automated parsing | – | LRLS—specification | GraphML, XML, HTML |
| RQ0 | Zeni2018 [99] | – | NomosT | NomosT | GaiusT, StanfordNLP | – | WordNet, GoogleNgrams |
| RQ1 | Bench1997 [23] | Bench methodology | – | – | – | – | Ontolingua |
| RQ1 | Gangemi2005 [43] | Gangemi methodology | – | – | – | DOLCE-Lite-Plus | CLO, JurWordNet, DOLCE+ |
| RQ1 | Griffo2015 [6] | – | UFO | – | – | UFO-L ontology | UFO, LCO |
| RQ1 | Griffo2017 [48] | Griffo methodology | – | ArchiMate extended | ArchiMate | UFO-S ontology | UFO-S, UFO-L |
| RQ1 | Kabilan2003 [54] | Multi-tier ontology framework | – | Ontology-modeling tool | Protégé 2000, DAML + OIL, Duet | Ontology modeling language | – |
| RQ1 | Corcho2005 [37] | Methontology | – | – | WebODE, ODE | – | – |
| RQ1 | Yan2006 [98] | Framework of contract ontol | – | – | Protégé2000, XML | – | OWL |
| RQ1 | Boer2008 [28] | – | – | MetaLex XML, LKIF | XML | MetaLex XML, LKIF | Description Logic |
| RQ2 | Kiyavitskaya2007 [9] | Cerno framework | Breaux methodology | Lightweight text analysis | – | – | WordNet, GoogleNgrams |
| RQ2 | Kiyavitskaya2009 [56] | Cerno framework ext | Cerno framework | GaiusT | Lightweight text analysis | – | – |
| RQ2 | Soavi2020 [2] | Four-step process | – | Contratto | GaiusT, StanfordNLP, spacy | – | WordNet, FrameNet, Symboleo ontology |
| RQ2 | Zeni2015 [4] | – | Cerno framework | GaiusT refined | GaiusT, Annotator Schema Generator, Database mapper | – | WordNet, GoogleNgrams |
| RQ2 | Araujo2013 [18] | Araujo methodology | – | – | Palavras parser | – | Extraction ontology, domain ontology |
| RQ2 | Lesmo2009 [65] | – | – | NLP-based system | Deep syntactic analysis, shallow semantic interpretation | – | NormeInRete |
| RQ2 | Mazzei2009 [70] | Mazzei methodology | – | – | XML, TUP | – | NormeInRete |
| RQ2 | Quaresma2010 [81] | Quaresma methodology | – | – | SVM, natural language parser | – | EurLex |

**Table 10** (continued)

| RQ | Description and reference | Methodology proposed | Methodology used | Tools proposed | Tools used | Resources proposed | Resources used |
|---|---|---|---|---|---|---|---|
| RQ2 | Biagioli2005 [25] | – | – | – | SVM, provision automatic classifier and extractor | – | NormeInRete |
| RQ2 | Neill2017 [75] | Neill methodology | – | – | Artificial neural network, distributional semantic model, GATE | – | – |
| RQ2 | Sainani2020 [87] | – | Constructive Grounded Theory | – | Bidirectional encoder representations, transformers | – | – |
| RQ2 | Chalkidis2017 [35] | – | – | – | BILSTM, LSTM, CRF | – | – |
| RQ2 | Grabmair2015 [47] | – | – | LUIMA | UIMA, Apache Lucene, Watson IBM | – | – |
| RQ2 | Barabucci2009 [22] | – | Akoma Ntoso standard, GRDDL | – | Akoma Ntoso | Vocabulary of common structures | – |
| RQ2 | Governatori2016 [46] | Governatori methodology | Business Process Management, Legal RuleML | – | LegalRuleML | – | – |
| RQ3 | Sleimi2020 [90] | Sleimi requirement templates and rules | – | – | Tregex, Java | – | – |
| RQ3 | Lee2006 [64] | Lee methodology | – | – | GENeric Object Model, OKBC | Problem Domain Ontology | – |
| RQ3 | Sleimi2019 [89] | – | Sleimi methodology | Toolchain | SPARQL | – | – |
| RQ3 | Lau2005 [63] | – | – | Shallow parser | – | Regulation repository | – |
| RQ3 | Garcia2017 [45] | – | NLP | CLIEL | Jape, GATE | – | – |
| RQ3 | Fischbach2021 [40] | CiRA | – | Fischbach tool | BERT, SVM, Naive Bayes classifier | – | – |
| RQ3 | Moens2007 [71] | – | – |  | Naive Bayes classifier | – | Araucaria |
| RQ3 | Grover2003 [50] | Grover methodology | NLP | Automatic text summarisation system | XML-based tool | – | – |
| RQ4 | Saias2005 [86] | Saias methodology | NLP | EVOLP, ISCO | – | – | OWL, Prolog |
| RQ4 | Lame2005 [60] | Lame methodology | – | – | Syntex | – | – |
| RQ4 | Amardeilh2005 [16] | Amardeilh methodology | – | – | ITM, IDE | – | RDF and OWL language |
| RQ4 | Amato2008 [17] | Amato methodology | NLP analysis | – | Jena API | – | ItalWordNet and JurWordNet |
| RQ4 | Francesconi2010 [42] | Francesconi methodology | – | xmLegesClassifier, xmLegesExtractor | GATE, T2K | – | LOIS, Archivio DoGi, JurWordNet |
| RQ4 | Völker2008 [94] | – | – | Text2Onto | GATE, JAPE, TreeTagger | – | WordNet |

Table 10 (continued)

| RQ | Description and reference | Methodology proposed | Methodology used | Tools proposed | Tools used | Resources proposed | Resources used |
|---|---|---|---|---|---|---|---|
| RQ5 | Konstantinou1993 [57] | – | – | ILAM module | – | – | NOMOS system |
| RQ5 | Governatori2006 [7] | – | – | – | – | Formal Contract Language (FCL) | BCL |
| RQ5 | Montazeri2011 [72] | AnaCon framework | – | Cont_ParserScriptGen, testGrammarCl, Comparison, Cont_GF_Cl, CLAN | | – | Grammatical Framework |
| RQ5 | Libal2019 [67] | Normative Detachment Structure with Ideal Conditions (NDSIC) | Deontic Logic | MleanCoP | – | – | – |
| RQ5 | Fornara2009 [41] | | – | – | Protege 4.0 SWRL rules, Java program | TBox, Rbox and Abox ontology | – |
| RQ5 | Palmirani2011 [79] | – | – | – | – | LegalRuleML language | RuleML language |
| RQ5 | Rabinia2020 [82] | Formal Legal GRL | – | – | GRL's tool, jUCMNav | FLG Procedure | – |
| RQ5 | Boella2009 [27] | Boella methodology | Extended defeasible logic | | – | Extension of DL | – |
| RQ5 | Maxwell2009 [69] | Production Rule Modeling methodology | Manual analysis | – | – | Prolog rules | Prolog language |

ordered according to their appearance in this section and for the main RQ they address.

## Summary of Findings

The SLR supports the identification of the main research efforts and existing gaps concerning the six RQs. Below, we summarize the results of our work and underline the most recurring, novel, and cited works.

### RQ0: What are the Main Approaches for Translating Legal Documents into Formal Specifications?

A limited number of works have considered the full objective of translating legal documents into formal specifications. More generally, limited interest in formal specifications has been identified despite an increase in the last years with the development of smart contract technologies. As such, regarding the overall transformation process, it should be considered the possibility to complement different approaches concerning the four steps presented in the SLR. Most of the few approaches that perform the full process at first identify concepts and relationships in a legal text to subsequently formally represent them with formal logics described in works related to RQ5. The most recurring approach relies on the annotation of the concepts using an ontology to generate a skeleton of specifications and formal expressions in [2, 5, 80]. Other approaches are tailored to the generation of models of law [99], Internet contracts [49], or rely on conditional structures [52] to extract and model contract elements. In recent work, the full process to generate formal specifications for a smart contract is presented [39]. The translation process is often simplified through the use of normalized or controlled NL to better identify and represent requirements for formal specifications [30, 64, 89, 90]. Finally, significant initiatives have been identified that aim at providing tools supporting management and formal representation of legal documents and contracts [22, 46]. However, none of them covers the full process of translating a legal text into formal specifications, and they generally require significant and time-consuming manual support. Moreover, the ability to generate a formal language expressive enough to represent the nuances of legal texts would still need significant effort.

### RQ1: What Legal Ontologies Have Been Used for the Translation?

Significant work has been done to apply, create, or manage a legal ontology for legal documents or contracts. Existing approaches can be divided into two categories frequently considered together: ontology elements are identified from text, or vice versa, text is annotated based on an existing ontology. The SLR identified significant misalignment in defining the concept of ontology itself, as it is frequently interchangeably used with terms such as ‘domain model’ or ‘conceptual model’. Generally, ontologies offer a higher degree of reusability for different contracts, whereas domain and conceptual models are specialized for a given type of contract (e.g., for renting or employment). The creation of ontologies or domain models from text has been investigated by [16, 17, 37, 60, 81]. Other approaches rely on multi-tier ontologies to move from the abstract to the specific elements, where the specific elements could also represent a domain model [39, 54]. The multi-tier ontology allows dealing with the different levels of abstraction required at the different layers (e.g., technical implementation, business logic) and eases communication among different audiences in contract automation (e.g., lawyers and programmers). A frequently used tool for modelling ontologies is Protégé, an ontology editor and knowledge acquisition system that has been used by [18, 41, 98]. The use of Protégé allows relying on a free, open-source platform that is supported by a broad community of users. Significant research has taken place to focus on specializing an existing ontology for different contexts to increase the comparability and reusability of ontologies. This has been mainly performed with the support of OWL, a language to represent ontologies that have been widely used [18, 28, 41, 42, 86, 99]. Other approaches use tools compatible with OWL [4, 22] and most notably [37] that proposes METHONTOLOGY, which has been adapted to different domains without significant technical support. DAML + OIL, the predecessor of OWL, is used by [49] in a popular work with a rule-based technique to automate the execution of business contracts along their lifecycle. Ontologies are similarly implemented using LKIF by [28]. A legal ontology frequently applied is CLO, which gained traction also thanks to [43] that relies on DOLCE and a JurWordNet lexicon. Similarly, there are ontologies adapted from CLO [42] or its evolution UFO-L CLO [2, 5, 6, 48, 80]. The SLR seems to suggest the importance of using a common standard to express an ontology, frequently implemented in OWL, and based on an existing ontology, such as CLO to rely on a solid, well-documented, and comparable basis. Moreover, the review suggests the existence of the open question of having legal ontologies that have sufficient expressiveness to properly represent legal documents, also considering complementary tools for contract management (e.g., ArchiMate [48]), to increase their adoption and use by legal practitioners.

## RQ2: What Annotation Approaches are Used for Semantic Annotation of Legal Text?

Significant research effort has been undertaken concerning semantic annotation of legal text, although some challenges still have to be overcome to efficiently identify concepts from legal texts. Frequently, annotation is implicitly considered as the foundational element to perform the general process of translating a legal contract into a formal specification. Two main approaches have been identified; the first and most widespread refers to the implementation of NLP based rules, used in a varied range of tools to support computers in understanding and modelling NL documents [2, 4, 9, 18, 45, 50, 52, 56, 70]. The second type of approaches refer to the use of ML techniques, not necessarily specialized for NL, that apply algorithms to learn and improve from experience [25, 71, 75, 81, 87] and that frequently rely on SVM. Although NLP is a research area of ML, frequently, the approaches proposed by the authors are generally classified under one category or another, or seldom with complementary approaches (e.g., [47]). The approaches referring to ML, are more recent due to the computational power required for supervised and unsupervised learning. In ontology-based semantic annotation—the most widespread approach—the process identifies instances of the concepts defined in the ontology, whereas structure annotation is commonly referred to on the definition of grammar. A legal text is parsed to perform syntactic analysis based on grammar rules to identify the structure of a phrase and how words relate to each other and mainly involves tokenization and PoS (Part of Speech) tagging [4, 17, 42, 45, 47, 81, 87, 94]. Structural annotation is mostly performed relying on the definition of a grammar to identify structure elements of a contract and, less frequently on ML techniques [35]. Rules IF_THEN are used by [52, 70] to identify semantic content. Accuracy in structural annotation performs better than in semantic annotation as it implies a lower level of ambiguity. Finally, for annotation, it is worth underlying significant tools such as Akoma Ntoso [22] and LegalRuleML [46] that take a holistic approach in the lifecycle of a contract and that use XML markup. The interest in those approaches mostly relies on the open-source, collaborative approach that lays the foundations for collaboratively developed, solidly tested tools and with a broader spectrum of uses, differently from the majority of works identified in the research.

## RQ3: What are Main Approaches for Mining Relationships from Annotated Text?

The identification of relationships in legal text has been frequently considered as a sub-task of semantic annotation; in this case, the proposed approaches share significant commonalities with the annotation process. The mining of relationships in the legal context has been studied with two objectives. The first objective relates to the identification of relationships within a phrase (e.g., noun, verbs, and adjectives) [4, 52, 70], whereas the second refers to the identification of relationships between different sections of legal text (e.g., legal provisions, clauses, etc.) [46, 56, 65]. As such, the identification of relationships within a phrase appears to support more significantly the objective of generating formal specifications compared to the identification of relationships among different sections of a text. The latter approach is useful as a complementary analysis to identify dependencies among different parts of the text to infer contextual knowledge. The identification of relationships among different parts of legal text has been performed for legal reasoning in [26, 67, 71], or for the identification of structural parts of a document (e.g., title, clause, and section) in [45, 82]. Legal reasoning—which received significant academic interest—seems more tailored for the analysis of a legal document as a whole, rather than the generation of formal specifications from NL. Among other approaches for mining relationships from text, checklist and templates [50, 64] or the identification of causality in text [40] represent a useful intermediate step to support the identification of relationships. However, none of them fully cover the requirements to identify the relationships between the concepts identified in semantic annotation. The proposed approaches tend to be fragmented, applicable only to certain concepts and frequently based on manual identification, suggesting the need for further research in approaches tailoring specifically the identification of relationships.

## RQ4: What are Main Techniques for Formalizing NL Terms into a Domain Model?

The formalization of NL terms into a domain model is frequently presented as part of the effort of generating an ontology from a legal contract; as such, there frequently is overlapping with RQ2. In several cases, the construction of a domain model has been identified as a subtask in the process of building an ontology using patterns and conditional rules [16, 17, 37] or in the context of multilingual ontologies [42] that relies on formal logic or graph-theoretic representations. The proposed approaches apply supervised ML techniques, and are represented in first-order logic [86] or rely on statistical measures to identify the relatedness of terms [61]. An alternative approach could be represented by the adoption of an existing ontology that is subsequently refined into a domain model for a specific contract as suggested in [42]. A manual approach relies on the use of checklists and templates [64]. Still, unresolved remains the development of a lexicon, or more generally the development of tools to automatically or semi-automatically generate a domain model from NL terms. Finally, an increased conceptual alignment in the scientific community concerning the concepts of domain models, conceptual models and ontologies would create a better understanding and applicability of the existing approaches.

### RQ5: What Kinds of Techniques Have Been Studied for Translating NL Expressions into Formal Ones for Legal Documents?

A challenge recurringly mentioned in the SLR refers to the difficulty in conveying the expressiveness of legal documents into formal expressions. The translation of NL expressions into formal ones is mainly based on knowledge representation languages and frequently relies on Deontic Logics (e.g., obligations and permissions) [7]. One of the first efforts for legal documents relates to propositional logic in BCL and represents the more complete and appreciated research effort [7]. BCL includes a formal system that allows identification of ambiguities, determination of missing or implied statements, analysis of the expected behaviours of the parties, and existing relationships between different parts of the contract. Propositional logic is subsequently adopted by [67]. However, propositional logic has important limitations with respect to expressiveness and does not support the formalization of many elements of legal contracts. First-Order Logic, on the other hand, has been used by [41, 86]; its limitations concern the representation of categories and quantification of relations. Event Calculus—that similarly to smart contracts relies on an event-based logic—allows the representation of temporal events and their inter-relationships, and is used by [80]. Sleimi et al. [90] rely on templates for an intermediate step to formalize NL from legal documents. The approaches of [67, 72] rely on controlled or normalized text before the generation of formal expressions. Another approach is proposed by [69] and relies on production rules. The use of templates, normalized text, or production rules—despite being performed mainly manually—seems to represent a necessary intermediate step to increase accuracy in generating formal specifications. The last years saw the development of approaches concerning the generation of specifications for smart contracts [39] or the drafting of contracts in NL-like specifications that can be directly implemented into a smart contract [53], but the latter approach entails a lower level of formalisation. On the other hand, a higher degree of understandability may ensure that the formal specification properly reflects the intentions of the contracting parties during the lifecycle of the contract. In general, selected papers have focused on proposals of specification languages, whereas more significant efforts need to be devoted to techniques—including automated ones—to translate NL expressions into such specification languages.

## Threats to Validity

The SLR was conducted following the guidelines proposed by Kitchenham [10]. These guidelines also call for consideration of threats to validity. Such threats are discussed in this section.

### Study Completeness

This threat relates to the question “How complete is the list of papers considered in the study?” The threat is mainly related to the search queries used for each RQ to retrieve relevant papers. Given the significant number of studies and different approaches which could potentially contribute to a given RQ, it is challenging to define search queries that are neither too strict nor too broad. The selection of papers was performed to identify the most relevant works based on the number of citations, favouring implementations and authors with a track record on the topic of this SLR. Even so, the process implied a certain level of subjectivity, and therefore, the selection of the papers may have been different if performed by different authors. To mitigate this threat, queries and selected papers were presented to three experts on the topic of the SLR who suggested missing papers. As a result, queries were revised to ensure that suggested relevant papers would be selected in a second round of searching.

Another threat is related to the choice of the ACM Guide to Computing Literature as the repository used to determine relevant works. Even though the repository contains papers published in the three most important publishers in the field (i.e., ACM, IEEE, and Springer), other significant works may have been produced with a different publisher or may have not been included in the repository. Similar considerations apply for recent papers which have been included as the most relevant papers presented in 2019–2021, and for those illustrating formal specification languages that had not been identified by the first search. The point is: as executed, the SLR did address the completeness threat by including recent papers and papers on formal specification languages and legal ontologies that were not accounted for by the RQ queries. Moreover, the SLR is based on a proposed decomposition of the overall process of generating formal specifications from a text in natural language. Such decomposition is subjective and other authors may have proposed other approaches and steps.

### Classification and Presentation

The classification of papers according to the RQs and the types of issues they address also contains elements of subjectivity. To mitigate this threat, a sample of the papers of Table 9 were discussed between two authors and a few discrepancies, mostly on the interpretation of RQs, have been highlighted. However, the classification of papers into RQs entails significant levels of subjectivity. This subjectivity was exacerbated by the use of different terminologies among different scientific communities in the domain. For example, the terms 'ontology' and 'conceptual model' are frequently used interchangeably in the RE community. Moreover, the framework proposed in the SLR is based on RQs that correspond to subproblems of the process of generating specifications from legal documents. Papers that did not match this problem decomposition were hard to classify. Here, we relied on the experience of the three senior authors of this study which have been working on the topic for more than a decade.

### Synopses of Papers

The synopses of the most relevant papers included in the SLR is, of course, subjective as it also depends on the background and understanding of the reader. The synopses were generated by the first author. To mitigate this threat, synopses were spot-checked by other authors for accuracy and presentation consistency.

## Related Work

This section is intended to cover papers that are themselves reviews one of the topics identified by the SQs in the SLR and have been classified as meta-studies. It does not cover non-review papers that are addressing problems on the translation process itself. These papers are discussed here. No review was identified that specifically refers to the translation of contracts in natural language into formal specifications.

Regarding ontologies, Valente [92] reviews different types of legal ontologies in the domain of Artificial Intelligence (AI) and law. The author proposes a classification based on the goal of the ontology, the need to structure information, legal reasoning, search and extraction of semantics, and understanding a domain. Valente focuses on helping to understand the implications of the use of different ontology types. Ontologies with light structures frequently rely on graphical representation and taxonomies, whereas highly structured ontologies are usually based on formal representation languages. Breuker [31] identifies the main concepts recurring in different legal ontologies to be able to reuse or combine them for different domains. The reuse of legal ontologies may be useful for information retrieval purposes, for annotation and tagging, to enhance the ability to query a legal document for related terms (e.g., identification of synonyms), or to avoid ambiguity.

Wyner et al. [97] explore different annotation approaches to extract an argument from legal texts to support decision-making by judges. The focus of the authors is on the use of context-free grammars to extract arguments together with NLP tools and ontologies to identify parties of a legal case and their relationships. A significant gap is identified in the meta-study and concerns the lack of any support for the time-consuming task of extracting an ontology from a legal text.

Otto et al. [77] review studies on compliance of legal text and survey the main research efforts in modelling and using legal text for system development, which are based on various forms of logic, goal modelling, and semi-structured representations. The focus of this paper is to determine the applicable regulations and to create compliance policies concerning those regulations. Based on the analysis, the authors propose a broad set of requirements for a system aiming to support compliance auditors.

Boella et al. [27] review the different approaches to represent legal knowledge to support requirements engineering activities. The authors underline the importance of the use of legal experts in the process as existing tools and approaches do not fully account for the implications of the law. Most of the proposed approaches adopt a 'textualist' view of the law, frequently associating a norm to a statement and lacking a holistic view of the nature of the law. Formal approaches are rarely used by law practitioners as they do not fully reflect the expressiveness of legal text and NL.

## Conclusions

We present an SLR on the process of translating contracts expressed in NL into formal specifications. The SRL relies on a framework to perform the process which is based on four steps: (a) structural and semantic annotation, (b) identification of relationships, (c) domain modeling, and (d) generation of a formal specification. Furthermore, the SLR investigates legal ontologies and the implication of their use in the four steps. Much work has been performed concerning annotation of legal text, both structural and semantic and the use of ontologies. Remaining steps have received significantly less attention. In this respect, the SLR has identified existing research gaps for the problem at hand.

For future research, the study suggests focusing on three main open challenges. First, for each of the four steps, the tools and the approaches proposed tend to have a significant level of domain dependence. That implies that their ability to support a translation step significantly decreases when applied to a different domain. As a result, a significant number of tools and approaches are proposed but a very limited amount among them can be efficiently applied to different domains. Second, the challenge of building tools that automatically identify semantic and structure elements from the contract text with near expert performance. Finally, it is essential to ensure that formal specifications fully capture the intent of legal documents to enhance adoption by legal practitioners as well as the quality of software systems that support the practice of Law. Further involvement of legal professionals in developing the tools and the approaches proposed in this SLR would be highly beneficial. With the ability to better define specifications for a developer that properly reflect the content of the contract and support the identification of activities that should be performed offline for elements for which contract automation presently has significant limitations (e.g., litigation). The ability to overcome such challenges could benefit from the recent interest and developments of smart contracts and blockchain for automated contract execution. In our future work, we are planning to test the use of a contract template that is implemented using Symboleo [5], a formal specification language for legal contracts.

**Funding** Open access funding provided by Università degli Studi di Trento within the CRUI-CARE Agreement.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

1. Christidis K, Devetsikiotis M. Blockchains and smart contracts for the Internet of Things. IEEE Access. 2016;4:2292–303.
2. Soavi M, Zeni N, Mylopoulos J, Mich L. ContracT–from legal contracts to formal specifications: preliminary results. In: IFIP working conference on the practice of enterprise modeling; 2020. pp. 124–137.
3. Guarino N, Oberle D, Staab S. What is an ontology. In: Handbook on ontologies; 2009. pp. 1–17.
4. Zeni N, Kiyavitskaya N, Mich L, Cordy JR, Mylopoulos J. GaiusT: supporting the extraction of rights and obligations for regulatory compliance. Requir Eng. 2015;20(1):1–22.
5. Sharifi S, Parvizimosaed A, Amyot D, Logrippo L, Mylopoulos J. Symboleo: towards a specification language for legal contracts. In: 2020 IEEE 28th international requirements engineering conference (RE); 2020.
6. Griffo C, Almeida JP, Guizzardi G. Towards a legal core ontology based on Alexy’s theory of fundamental rights. In: Multilingual workshop on artificial intelligence and law, ICAIL; 2015.
7. Governatori G, Milosevic Z. A formal analysis of a business contract language. Int J Coop Inf Syst. 2006;15(4):659–85.
8. I. S. C. Committee. IEEE Standards Glossary of Software Engineering Terminology (IEEE Std 610.12 1990). Los Alamitos: IEEE; 1990.
9. Kiyavitskaya N, Zeni N, Breaux TD, Antón AI, Cordy JR, Mich L, Mylopoulos J. Extracting rights and obligations from regulations: toward a tool-supported process. In: Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering; 2007.
10. Kitchenham B, Pfleeger S, Pickard L, Jones P, Hoaglin D, Emam KE, Rosenberg J. Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng. 2002;28(8):721–34.
11. Horkoff J, Li T, Li F-L, Salnitri M, Cardoso E, Giorgini P, Mylopoulos J. Using goal models downstream: a systematic roadmap and literature review. Int J Inf Syst Model Des. 2015;6(2):1–42.
12. Horkoff J, Aydemir FB, Cardoso E, Li T, Maté A, Paja E, Salnitri M, Piras L, Mylopoulos J, Giorgini P. Goal-oriented requirements engineering: an extended systematic mapping study. Requir Eng. 2019;24(2):133–60.
13. Agnoloni T, Tiscornia D. Semantic web standards and ontologies for legislative drafting support. In: ePart'10 Proceedings of the 2nd IFIP WG 8.5 international conference on Electronic participation; 2010.
14. Ajani G, Lesmo L, Boella G, Mazzei A, Rossi P. Terminological and ontological analysis of European directives: multilinguism in law. In: Proceedings of the 11th international conference on Artificial intelligence and law; 2007.
15. Alsaadi M, Lisitsa A, Qasaimeh M. Minimizing the ambiguities in medical devices regulations based on software requirement engineering techniques. In: Proceedings of the second international conference on data science, e-learning and information systems; 2019. pp. 1–5.
16. Amardeilh F, Laublet P, Minel J-L. Document annotation and ontology population from linguistic extractions. In: Proceedings of the 3rd international conference on knowledge capture; 2005.
17. Amato F, Mazzeo A, Penta A, Picariello A. Building RDF ontologies from semi-structured legal documents. In: 2008 international conference on complex, intelligent and software intensive systems; 2008.
18. Araujo DAD, Rigo SJ, Müller C, Chishman RLDO. Automatic information extraction from texts with inference and linguistic knowledge acquisition rules. In: Web Intelligence/IAT Workshops; 2013.
19. Ashley KD. Ontological requirements for analogical, teleological, and hypothetical legal reasoning. In: Proceedings of the 12th international conference on artificial intelligence and law; 2009.
20. Ashley KD, Walker VR. Toward constructing evidence-based legal arguments using legal decision documents and machine learning. In: Proceedings of the fourteenth international conference on artificial intelligence and law; 2013.
21. Azzopardi S, Pace GJ, Schapachnik F. On observing contracts: deontic contracts meet smart contracts. In: Legal knowledge and information systems; 2018. pp. 21–30.
22. Barabucci G, Cervone L, Palmirani M, Peroni S, Vitali F. Multilayer markup and ontological structures in Akoma Ntoso. In: AICOL-I/IVR-XXIV'09 Proceedings of the 2009 international conference on AI approaches to the complexity of legal systems:

complex systems, the semantic web, ontologies, argumentation, and dialogue; 2009.
23. Bench-Capon TJM, Visser PRS. Ontologies in legal information systems; the need for explicit specifications of domain conceptualisations. In: Proceedings of the 6th international conference on Artificial intelligence and law; 1997.
24. Bench-Capon TJ, Visser PR. Open texture and ontologies in legal information systems. In: Database and expert systems applications. 8th international conference, DEXA'97; 1997.
25. Biagioli C, Francesconi E, Passerini A, Montemagni S, Soria C. Automatic semantics extraction in law documents. In: Proceedings of the 10th international conference on Artificial intelligence and law; 2005.
26. Boella G, Governatori G, Rotolo A, Torre LVD. Lex minus dixit quam voluit, lex magis dixit quam voluit: a formal study on legal compliance and interpretation. In: AICOL-I/IVR-XXIV'09 Proceedings of the 2009 international conference on AI approaches to the complexity of legal systems: complex systems, the semantic web, ontologies, argumentation, and dialogue; 2009.
27. Boella G, Humphreys L, Muthuri R, Rossi P, Torre LWN. A critical analysis of legal requirements engineering from the perspective of legal practice. In: Requirements engineering and law (RELAW), 2014 IEEE 7th international workshop on; 2014.
28. Boer A, Winkels R, Vitali F. Metalex XML and the legal knowledge interchange format. In: Computable models of the law. Berlin: Springer; 2008. pp. 21–41.
29. Branting K, Weiss B, Brown B, Pfeifer C, Chakraborty A, Ferro L, Pfaff M, Yeh A. Semi-supervised methods for explainable legal prediction. In: Proceedings of the seventeenth international conference on artificial intelligence and law; 2019.
30. Breaux TD, Gordon DG. Regulatory requirements traceability and analysis using semi-formal specifications. In International working conference on requirements engineering: Foundation for software quality. In: International working conference on requirements engineering: foundation for software quality; 2013. pp. 131–157.
31. Breuker J, Valente A, Winkels R. Use and reuse of legal ontologies in knowledge engineering and information management. Law and the Semantic Web, Springer. 2005. pp. 36–64.
32. Bueno TCD, Wangenheim CG, Mattos EDS, Hoeschl HC, Barcia RM. JurisConsulto: retrieval in jurisprudencial text bases using juridical terminology. In: Proceedings of the 7th international conference on Artificial intelligence and law; 1999.
33. Capuano N, Maio CD, Salerno S, Toti D. A methodology based on commonsense knowledge and ontologies for the automatic classification of legal cases. In: Proceedings of the 4th international conference on web intelligence, mining and semantics (WIMS14); 2014.
34. Ceci M, Palmirani M. Ontology framework for judgment modelling. In: AICOL'11 Proceedings of the 25th IVR congress conference on AI approaches to the complexity of legal systems: models and ethical challenges for legal systems, legal language and legal ontologies, argumentation and software agents; 2011.
35. Chalkidis I, Androutsopoulos I. A deep learning approach to contract element extraction. In: JURIX; 2017. pp. 155–164.
36. Chieze E, Farzindar A, Lapalme G. An automatic system for summarization and information extraction of legal information. Semantic processing of legal texts, Springer. 2010. pp. 216–234.
37. Corcho O, Fernández-López M, Gómez-Pérez A, López-Cima A. Building legal ontologies with METHONTOLOGY and WebODE. Law and the Semantic Web, Springer. 2005. pp. 142–157.
38. Despres S, Szulman S. TERMINAE method and integration process for legal ontology building. In: IEA/AIE'06 Proceedings of the 19th international conference on advances in applied artificial intelligence: industrial, engineering and other applications of applied intelligent systems; 2006.
39. Dwivedi V, Norta A, Wulf A, Leiding B, Saxena S, Udokwu C. A formal specification smart-contract language for legally binding decentralized autonomous organizations. In: IEEE access; 2021.
40. Fischbach J, Frattini J, Spaans A, Kummeth M, Vogelsang A, Mendez D, Unterkalmsteiner M. Automatic detection of causality in requirement artifacts: the CiRA Approach. In: REFSQ; 2021.
41. Fornara N, Colombetti M. Ontology and time evolution of obligations and prohibitions using semantic web technology. In: DALT'09 Proceedings of the 7th international conference on Declarative Agent Languages and Technologies; 2009.
42. Francesconi E, Montemagni S, Peters W, Tiscornia D. Integrating a bottom–up and top–down methodology for building semantic resources for the multilingual legal domain. Semantic processing of legal texts, Springer. 2010. pp. 95–121.
43. Gangemi A, Sagri M-T, Tiscornia D. A constructive framework for legal ontologies. Law and the Semantic Web, Springer. 2005. pp. 97–124.
44. Gao X, Singh MP, Mehra P. Mining business contracts for service exceptions. IEEE Trans Serv Comput. 2011;5(3):333–44.
45. García-Constantino M, Atkinson K, Bollegala D, Chapman K, Coenen F, Roberts C, Robson K. CLIEL: context-based information extraction from commercial law documents. In: Proceedings of the 16th edition of the international conference on articial intelligence and law; 2017.
46. Governatori G, Hashmi M, Lam H-P, Villata S, Palmirani M. Semantic business process regulatory compliance checking using LegalRuleML. In: EKAW 2016 20th international conference on knowledge engineering and knowledge management, vol. 10024, 2016.
47. Grabmair M, Ashley KD, Chen R, Sureshkumar P, Wang C, Nyberg E, Walker VR. Introducing LUIMA: an experiment in legal conceptual retrieval of vaccine injury decisions using a UIMA type system and tools. In: Proceedings of the 15th international conference on artificial intelligence and law; 2015.
48. Griffo C, Almeida JP, Guizzardi G, Nardi JC. From an ontology of service contracts to contract modeling in enterprise architecture. In: IEEE 21st international enterprise distributed object computing conference (EDOC); 2017.
49. Grosof BN, Poon TC. SweetDeal: representing agent contracts with exceptions using XML rules, ontologies, and process descriptions. In: Proceedings of the 12th international conference on World Wide Web; 2003.
50. Grover C, Hachey B, Hughson I, Korycinski C. Automatic summarisation of legal documents. In: Proceedings of the 9th international conference on Artificial intelligence and law; 2003.
51. Hasan MM, Aganostopoulos D, Loucopoulos P, Nikolaidou M. Regulatory requirements compliance in e-government system development: an ontology framework. In: 10th international conference on theory and practice of electronic governance. 2017. pp. 441–449.
52. Hashmi M. A methodology for extracting legal norms from regulatory documents. In: Science & engineering faculty; 2015.
53. He X, Qin B, Zhu Y, Chen X, Liu Y. SPESC: A Specification Language for Smart Contracts. In: 2018 IEEE 42nd annual computer software and applications conference (COMPSAC); 2018.
54. Kabilan V, Johannesson P. Semantic representation of contract knowledge using multi tier ontology. In: SWDB, 2003. pp. 395–414.
55. Kayed A. Building e-laws ontology: new approach. In: OTM'05 proceedings of the 2005 OTM confederated international conference on on the move to meaningful internet systems; 2005.
56. Kiyavitskaya N, Zeni N, Breaux TD, Antón AI, Cordy JR, Mich L, Mylopoulos J. Automating the extraction of rights and obligations

for regulatory compliance. In: ER '08 proceedings of the 27th international conference on conceptual modeling; 2008.

57. Konstantinou V, Sykes J, Yannopoulos GN. Can legal knowledge be derived from legal texts? In: Proceedings of the 4th international conference on Artificial intelligence and law. 1993. pp. 218–227.
58. Lagioia F, Micklitz HW, Panagis Y, Sartor G, Torroni P. Automated detection of unfair clauses in online consumer contracts. In: Legal knowledge and information systems: JURIX 2017: the thirtieth annual conference, vol. 302; 2017.
59. Lagioia F, Ruggeri F, Drazewski K, Lippi M, Micklitz HW, Torroni P, Sartor G. Deep learning for detecting and explaining unfairness in consumer contracts. In: Legal knowledge and information systems, IOS Press; 2019. pp. 43–52.
60. Lame G. Using NLP techniques to identify legal ontology components: concepts and relations. Law and the Semantic Web, Springer. 2005.
61. Lame G, Desprès S. Updating ontologies in the legal domain. In: Proceedings of the 10th international conference on Artificial intelligence and law. 2005.
62. Lau GT, Law KH, Wiederhold G. Similarity analysis on government regulations. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining; 2003.
63. Lau GT, Law KH, Wiederhold G. Legal information retrieval and application to e-rulemaking. In: Proceedings of the 10th international conference on Artificial intelligence and law. 2005.
64. Lee S-W, Gandhi R, Muthurajan D, Yavagal D, Ahn G-J. Building problem domain ontology from security requirements in regulatory documents. In: Proceedings of the 2006 international workshop on Software engineering for secure systems. 2006.
65. Lesmo L, Mazzei A, Radicioni DP. Extracting semantic annotations from legal texts. In: Proceedings of the 20th ACM conference on hypertext and hypermedia. 2009.
66. Levy F, Guisse A, Nazarenko A, Omrane N, Szulman S. An environment for the joint management of written policies and business rules. In: 2010 22nd IEEE international conference on tools with artificial intelligence; 2010.
67. Libal T, Pascucci M. Automated reasoning in normative detachment structures with ideal conditions. In: Proceedings of the seventeenth international conference on artificial intelligence and law. 2019.
68. Liu G, Buntine W, Yang X, Fu W. Research on domain-oriented latent policy lineage mining method. In: 2015 eighth international conference on internet computing for science and engineering (ICICSE). 2015.
69. Maxwell JC, Anton AI. Developing production rule models to aid in acquiring requirements from legal texts. In: 2009 17th IEEE international requirements engineering conference. 2009.
70. Mazzei A, Radicioni DP, Brighi R. NLP-based extraction of modificatory provisions semantics. In: Proceedings of the 12th international conference on artificial intelligence and law. 2009.
71. Moens M-F, Boiy E, Palau RM, Reed C. Automatic detection of arguments in legal texts. In: Proceedings of the 11th international conference on Artificial intelligence and law, 2007.
72. Montazeri SM, Roy NKS, Schneider G. From contracts in structured English to CL specifications, vol. 68. 2011, pp. 55–69. *arXiv preprint* arXiv:1109.2657.
73. Moulin B, Rousseau D. Knowledge acquisition from prescriptive texts. In: Proceedings of the 3rd international conference on Industrial and engineering applications of artificial intelligence and expert systems. 1990.
74. Nadah N, Rosnay MD, Bachimont B. Licensing digital content with a generic ontology: escaping from the jungle of rights expression languages. In: Proceedings of the 11th international conference on Artificial intelligence and law. 2007.
75. Neill JO, Buitelaar P, Robin C, Brien LO. Classifying sentential modality in legal language: a use case in financial regulations, acts and directives. In: Proceedings of the 16th edition of the international conference on articial intelligence and law. 2017.
76. Osborn J, Sterling L. JUSTICE: a judicial search tool using intelligent concept extraction. In: 7th international conference on Artificial intelligence and law. 1999. pp. 173–181.
77. Otto P, Anton A. Addressing legal requirements in requirements engineering. In: 15th IEEE international requirements engineering conference (RE 2007). 2007.
78. Palmirani M, Brighi R. Model regularity of legal language in active modifications. In AICOL-I/IVR-XXIV'09 Proceedings of the 2009 international conference on AI approaches to the complexity of legal systems: complex systems, the semantic web, ontologies, argumentation, and dialogue. 2009.
79. Palmirani M, Governatori G, Rotolo A, Tabet S, Boley H, Paschke A. LegalRuleML: XML-based rules and norms. In: International workshop on rules and rule markup languages for the semantic web. Berlin: Springer; 2011.
80. Parvizimosaed A, Sharifi S, Amyot D, Logrippo L, Mylopoulos J. Subcontracting, assignment, and substitution for legal contracts in Symboleo. In: ER; 2020. pp. 271–285.
81. Quaresma P, Gonçalves T. Using linguistic information and machine learning techniques to identify entities from juridical documents. Semantic processing of legal texts, Springer. 2010. pp. 44–59.
82. Rabinia A, Ghanavati S, Humphreys L, Hahmann T. A methodology for implementing the formal legal-GRL framework: a research preview. In: International working conference on requirements engineering: foundation for software quality. 2020.
83. Rao PRK, Devi SL. Automatic identification of conceptual structures using deep boltzmann machines. In: Proceedings of the 7th forum for information retrieval evaluation on; 2015.
84. Rodrigues CMO, Azevedo RR, Freitas FLG, Silva EP, Barros PVS. An ontological approach for simulating legal action in the Brazilian penal code. In: Proceedings of the 30th annual ACM symposium on applied computing. 2015.
85. Roegiest A, Hudek AK, McNulty A. A dataset and an examination of identifying passages for due diligence. In: The 41st international ACM SIGIR conference on research & development in information retrieval. 2018.
86. Saias J, Quaresma P. A methodology to create legal ontologies in a logic programming information retrieval system. Law and the Semantic Web, Springer. Law and the Semantic Web, Springer. 2005. pp. 185–200.
87. Sainani A, Anish PR, Joshi V, Ghaisas S. Extracting and classifying requirements from software engineering contracts. In: 2020 IEEE 28th international requirements engineering conference (RE). 2020.
88. Sannier N, Adedjouma M, Sabetzadeh M, Briand L. An automated framework for detection and resolution of cross references in legal texts. Requir Eng. 2017;22(2):215–37.
89. Sleimi A, Ceci M, Sannier N, Sabetzadeh M, Briand L, Dann J. A query system for extracting requirements-related information from legal texts. In: 2019 IEEE 27th international requirements engineering conference (RE). 2019.
90. Sleimi A, Ceci M, Sabetzadeh M, Briand LC, Dann J. Automated recommendation of templates for legal requirements. In: 2020 IEEE 28th international requirements engineering conference (RE). 2020.
91. Schwitter R, Tilbrook M. Dynamic semantics at work. In: New frontiers in artificial intelligence. Berlin: Springer; 2003. pp. 416–424.
92. Valente A. Types and roles of legal ontologies. Law and the Semantic Web, Springer. 2005. pp. 65–76.

93. Villata S. Digital enforceable contracts (DEC): Making smart contracts smarter. In: Legal knowledge and information systems: JURIX 2020: the thirty-third annual conference, Brno, Czech Republic; 2020.
94. Völker J, Langa SF, Sure Y. Supporting the construction of Spanish legal ontologies with Text2Onto. 2008. pp. 105–112.
95. Walker VR, Han JH, Ni X, Yoseda K. Semantic types for computational legal reasoning: propositional connectives and sentence roles in the veterans' claims dataset. In: Proceedings of the 16th edition of the international conference on articial intelligence and law. 2017.
96. Weber-Jahnke JH, Onabajo A. Finding defects in natural language confidentiality requirements. In: 2009 17th IEEE international requirements engineering conference. 2009.
97. Wyner A, Mochales-Palau R, Moens M-F, Milward D. Approaches to text mining arguments from legal cases. Semantic processing of legal texts, Springer. 2010. pp. 60–79.
98. Yan Y, Zhang J, Yan M. Ontology modeling for contract: using OWL to express semantic relations. In: 2006 10th IEEE international enterprise distributed object computing conference (EDOC'06). 2006.
99. Zeni N, Seid EA, Engiel P, Mylopoulos J, Nómos T. NómosT: building large models of law with a tool-supported process. Data Knowl Eng. 2018;117:407–18.