ContracT – from Legal Contracts to Formal Specifications: Preliminary Results

Michele Soavi, Nicola Zeni, John Mylopoulos, Luisa Mich

ContracT – From Legal Contracts to Formal Specifications: Preliminary Results

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

[1] Department of Industrial Engineering, University of Trento
Via Sommarive 14, 38123 Povo (TN), Italy
michele.soavi@unitn.it, nicola.zeni@unitn.it, luisa.mich@unitn.it
[2] School of Electrical Engineering and Computer Science, 800 King Edward Ave.
Ottawa ON Canada K1N 6N5
jmylopou@uottawa.ca

Abstract. We are interested in semi-automating the process of generating a formal specification from a legal contract in natural language text form. Towards this end, we present a tool, named ContracT, that annotates legal contract text using an ontology for legal contracts. In the last part of the paper, we present results from a preliminary empirical evaluation of the tool that provided encouraging results in identifying contract concepts in text and discuss critical points to be tackled in future studies.

Keywords: Legal contract, Structure model, Ontology for contracts, Semantic annotation.

1 Introduction

Legal contracts constitute for millennia the main vehicle for conducting business transactions world-wide. They are established (aka 'formed' in Law) through a systematic process, followed by an execution ('performance' in Law), possibly dispute resolution and final termination. Contracts exist in natural language text and are generally written in legal terms using legal concepts.

We are interested in transforming legal contract text into a formal specification for two reasons. Firstly, there is much interest in Law to formal analysis of legal contracts to ensure they are consistent with the interests of all contracting parties. Formal analysis tools, such as model checkers [5] and SMT/OMT solvers [15], have come of age in the past decade and are used routinely to analyze various kinds of artifacts from hardware to software and business process designs. But these tools only work with a formal specification of the artifact to be analyzed. Secondly, there is a new class of software systems called smart contracts [19] that automate the execution and monitoring of legal contracts. Software Engineering principles call for a formal specification of any software system before proceeding with its implementation. The use of a formal specification is required to avoid ambiguity, a common trait of legal documents and contracts [7]. The core of such a specification is exactly the formal specification

of the legal contract whose execution is to be automated and monitored (e.g. smart contract).

Based on experiences with our earlier work [20,21] we envision the generation of a formal specification from natural language text as a three step process: (a) Annotate the text using the ontology of the specification language chosen; for legal contracts this ontology would include concepts such as 'role', 'obligation', 'power' and 'asset'; (b) Generate a skeleton of a specification that includes all and only the roles, obligations, powers, etc. of the legal contract; (c) Generate formal expressions for the constituents of obligations and powers from legal text fragments. Considering that formalization of legal text is a laborious and error-prone process, we envision our task as one of semi-automating the generation of formal specifications from natural language with a tool that improves the quality of generated specifications while reducing substantially manual effort of the annotators.

The main goal of this paper is to report results on step (a) defined above. Towards this end, we adopt GaiusT, a semantic annotation system for legal documents, to develop ContracT, a semantic annotation tool for laws and regulations. The tool uses two models to conduct the annotation process: a structure model that defines the structure of legal contracts, and an ontology of legal concepts to be identified through annotation. We then experiment with this tool to determine how well it does its annotation of legal text and report on a preliminary evaluation of ContracT performed to test the support provided to human annotators in the annotation of contracts.

The rest of the paper is organised as follows. Section 2 describes the structure and ontology models, while Section 3 describes the prototypical version of ContracT. Section 4 presents the evaluation results of an experiment using ContracT to annotate two contracts. Section 5 discusses related work, while Section 6 concludes and presents planned future work.

2 Defining the Annotation Models

2.1 Defining the Elements of a Contract

To define the structure model and a contract ontology to annotate contracts to support their translation into formal specifications we first provide an understanding of the legal notion of contract. In the most simple form, a contract is an agreement between two or more parties, meant to be legally binding and effecting a change in their legal position. A contract – similarly to a business process – is targeted at obtaining a specific outcome by requiring a specific behaviour to the involved parties. However – differently from a business process – a contract is legally binding for contracting parties and includes provisions for penalties and the corresponding compensations whenever each party is not compliant with the behaviour prescribed in the contract [7]. Typically, a contract is going through different stages in its lifecycle from its formation to the moment where the effects of the contract have ceased. Finally, the judgement of the par-

ties in the contractual agreement could differ also accounting for recurring cases of ambiguities in contracts and laws in general [7].

2.2 Defining the Structure Model

Identifying the structure of a natural language document is preliminary to semantic annotation as it supports the recognition of potential annotation units. The goal of the structure analysis is to improve the accuracy of the following conceptual annotation, capturing the semiformal structure of a contract. Given the variety of contracts and their different length ranging from a few sentences to hundreds of pages there is not a standard template. From an analysis of a set of contracts, we identified the following text units: **Title**, Subtitle, **Contract identifier**, Preamble, Definition, Part, **Clause**, **Signature**, **Date**, Cross-reference, Glossary and Term.

Usually a contract does not include all that text units. Mandatory units are highlighted in bold. The model includes also cross-references, necessary to create internal links between parts of the same document and external links with other documents, which are necessary to interpret contracts. In particular external links are useful to identify the related documents to be given in input in the process to translate contracts into specifications for automatable contracts.

2.3 An Ontology for Contracts

For the identification of the concepts to be included in the conceptual model we analysed the annotation models introduced in our previous research projects, GaiusT 2.0 and NómosT. This analysis is important to define the annotation rules. Given that those models were defined to support semantic annotation of generic legal documents, to annotate contracts, a specific subset of legal documents, they have to be specialised and integrated with new rules whenever necessary. To this end we also analyzed several existing ontologies covering legal domain, such as Public Procurement Ontology (PPROC) [1], OASIS LegalXML [2] and PROMS [3], but they are either too specialized or too broad and do not cover relevant concepts of a contract. As a result of our analysis we identified and adapted X UFO-L CLO, short for eXtended UFO-L Core Legal Ontology (CLO) [16], an ontology specific for contracts that extends UFO-L, a foundational ontology for the legal domain [9]. UFO-L represents a specialization of UFO, a descriptive foundational ontology that extends the ontology of UFO and conceptualizes its concepts for the legal domain. UFO is based on the Alexy's Theory of Constitutional Rights that concerns the structure and domain of constitutional rights [1].

The resulting annotation model includes the concepts in the first column of Table 1. The second column reports the concepts in X UFO-L CLO. The

[1] `http://contsem.unizar.es/def/sector-publico/pproc.html`.
[2] `http://www.legalxml.org/legacy/index.shtml`.
[3] `https://promsns.org/def/agr/agr.html`.

columns for GaiusT 2.0 and NómosT report the names used for concepts similar to those in Column 1 and help to highlight additions and changes needed for ContracT. For concepts included in the conceptual model defined for ContracT,

Table 1. Conceptual models comparison

ContracT	**X UFO-L CLO**	**GaiusT 2.0**	**NómosT**
Contract	✔	✔ Legal document	✔ Legal document
Asset	✔	✔ Resource	✔ Resource
Party	✔	✔ Actor	✔ ✘
Role	✔	✔	✔
Obligation	✔	✔	✔ Duty
Right	✘	✔	✔
Situation	✔	✘	✔
Condition	✘	✔ Temporal/Exception	✔ Antecedent/Consequent
Action	✔ Event	✔	✔
Power	✔	✘	✘

relevant relationships with GaiusT 2.0 and NómosT are the following (for a full description of the concepts see [9]):

Contract The first concept is used to label the document as such.
Asset An *Asset* is equivalent to *Consideration* in the common law [14]. In GaiusT 2.0 and NómosT the more general concept of *Resource* was used.
Party *Parties* are legal agents (persons or institutions) who are in relationship with an *Asset* (e.g. "own") in a *Contract*. In GaiusT 2.0 the more general concept of *Actor* is used. In NómosT the concept of *Party* is not defined.
Role In GaiusT 2.0 *Actors* play *Roles* in the concept of *Prescribed behaviour*.
Obligation In GaiusT 2.0, an *Obligation* is an *Action* that a *Party* is conditionally or *unconditionally* required to perform. In NómosT the concept of *Duty* was used.
Right In GaiusT 2.0 and in NómosT a *Right* is an *Action* that a *Party* is conditionally or *unconditionally* permitted to perform.
Situation *Situation* is a concept intoduced in NómosT.
Condition *Conditions* establish circumstances under which deontic concepts (*Right* and *Obligation*) are applicable. GaiusT 2.0 distinguishes

Temporal and *Exception*. NómosT uses *Antecedent* and *Consequent* conditions.
Action *Action* is not present in X UFO-L CLO but it is used by GaiusT 2.0 and NómosT in many of the annotation rules to be implemented as patterns.
Power It is a *Right* a *Party* can exert in specific *Situation*. *Power* is not defined in GaiusT 2.0 nor NómosT.

The result of the analysis highlighted that for ContracT it is necessary to partially adapt the rules defined for GaiusT 2.0 and NómosT and to introduce new patterns to implement such rules for the concept of *Power*.

The core of the conceptual model for ContracT is represented in Figure 1 [16]. In the present version we use a subset of the concepts defined in the ontology proposed by [16]: *Party*, *Role*, *Contract*, *Asset*, *Power* and *Obligation*, highlighted in Figure 1 in light blue.

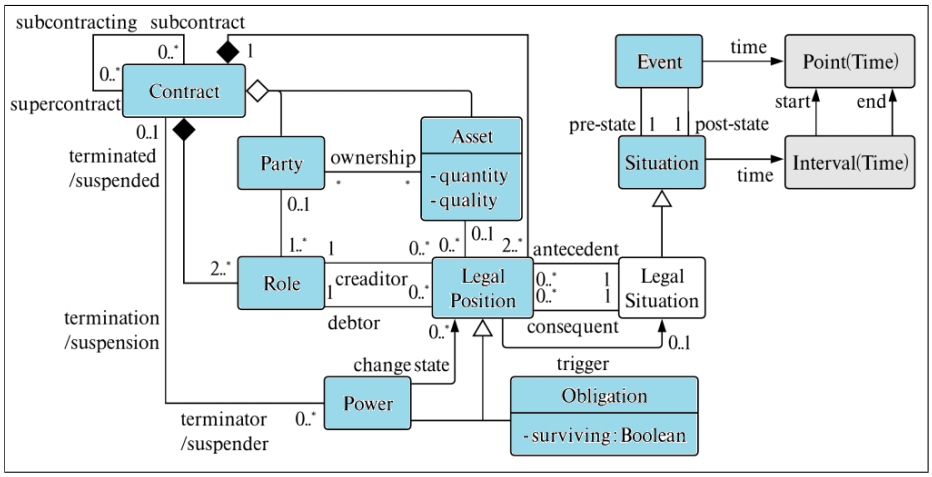

Fig. 1. Contract Ontology

3 ContracT

The two models defined to annotate contracts – structure and conceptual model – have been used to design and implement the first version of ContracT, an exploratory prototype. In particular, for the items in the models, patterns and identifiers have been defined and checked against those already used in GaiusT 2.0, adapting them or introducing new ones when necessary. The result is a new set of patterns.

Regarding structure annotation, text units defined in Section 2.2 have been translated into patterns for ContracT. Each pattern is represented in extended

Backus-Naur form (EBNF) notation and describes allowable combinations of text units and structure elements (Table 2). For each structure unit we have identified indicators and patterns to be used to define rules for ContracT, based on feature-based boundary detection of text fragments. A few examples of identifiers and patterns are listed in the second column in Table 2. Some of them have been adapted from those implemented in GaiusT 2.0 (e.g. Title and Date) others are new (e.g. Preamble and Signature). To test the ContracT module for structure annotation it was applied to three different types of real contracts – sale, shipment and rent. The goal was to identify critical issues and aspects to be addressed in future versions. Some annotations of the mandatory text units of a contract – *Title*, *Contract identifier*, *Part*, *Clause* – were almost all correctly identified. The `Title` pattern assumes that the title is the first not empty line of the input document, which is not always the case. For example, a contract document, can have a header before the title; to ignore it another pattern using formatting information (e.g. the fact that it is repeated on each page of the document) should be added. As regards `Contract identifier`, if it is given without the term *Contract* it is not identified; a pattern for numbers only or even worse a pattern using alphanumerical code would give false positives. Patterns for the text units `Part` and `Clause`, correctly identified explicit hierarchies.

Serious problems arose in the identification of *Date* text units, mostly because of the variety of date formats to be handled. *Definition* and *Glossary* have not been tested since in the contract examples such parts were not present. Rules for *Cross-references*, in particular for external references, defined in GaiusT 2.0 did not work because in a contract many different ways are existing to cite laws and regulations.

Regarding the semantic annotation of the concepts in Table 1, new rules and patterns have been introduced for *Asset* and *Party*. For *PrescribedBehaviour* the values of *Obligation* and *Right* for the type attributes existing patterns in GaiusT 2.0 were used. A set of auxiliary patterns – Resource, PositiveModal, NegativeModal and Action – were also used to extract instances of the concepts. The list of patterns used in the ContracT prototype is given in Table 3 in EBNF notation, with combinations of concepts, sub-patterns or structure elements.

Following these preliminary tests, we assume that to increase the ability to identify text units and concepts, domain knowledge, related to the type of a contract could be added. To preliminary test that assumption, we introduced a conceptual model for 'rent' contracts, instantiating a subset of concepts of the conceptual model in Figure 1. For example, the parties in a rent contract play the Landlord and Lessee roles; the type of asset in a rent contract can be listed as values for one of the Asset attributes. In general terms, this assumption corresponds to the definition of a library of contract templates to support the annotation of specific types of contracts. The possibility to use contract templates is particularly relevant for standardized, recurring and widespread contracts for which the cost to create a template is justified by the significant number of times the template could be reused.

The approach is similar to what proposed in [4] and envisaged, among others, by the International Swaps and Derivatives Association[4].

Table 2. EBNF grammar for contract text structure

⟨*CRLF*⟩ ::= '`\r\n`'
⟨*Digit*⟩ ::= '`0`'| '`1`' | '`2`' | '`3`' | '`4`' | '`5`' | '`6`' | '`7`' | '`8`' | '`9`'
⟨*ContracTitle*⟩ ::= ⟨*String*⟩ ⟨*CLRF*⟩
⟨*ContractIdent*⟩ ::= (['`Contract n.`']) (⟨*Digit*⟩|⟨*String*⟩) (⟨*CLRF*⟩)
⟨*ContractSubTitle*⟩ ::= ⟨*ContracTitle*⟩|(⟨*ContractIdent*⟩* | ⟨*ContracTitle*⟩*)(⟨*CLRF*⟩)
⟨*ContractPreamble*⟩ ::= ⟨*String*⟩
⟨*ContractPart*⟩ ::= ⟨*Digit*⟩ '`.`' ⟨*String*⟩ ⟨*CRLF*⟩ | ⟨*Digit*⟩ '`)`'
⟨*ContractDefinition*⟩ ::= ⟨*ContractPart*⟩ | '`Definition`' | '`Definitions`' ⟨*CRLF*⟩
⟨*Definition*⟩ ::= ⟨*Identifier*⟩ ('means' | 'refers to') ⟨*text*⟩
⟨*Definitions*⟩ ::= (⟨*Definition*⟩ ',')* ⟨*Definition*⟩
⟨*ContractClause*⟩ ::= ⟨*ContractPart*⟩ ⟨*Digit*⟩ '`.`' ⟨*Digit*⟩
⟨*ContractSignature*⟩ ::= '`Signature`' ⟨*String*⟩ ⟨*CRLF*⟩
⟨*ContractDate*⟩ ::= '`Date`' ⟨*TemporalItem*⟩
⟨*StartDate*⟩ :: = ⟨*TemporalItem*⟩ | ⟨*TemporalExpression*⟩
⟨*ContractXreference*⟩ ::= not implemented
⟨*ContractGlossary*⟩ ::= ⟨*ContractPart*⟩ | '`Glossary`' ⟨*CRLF*⟩

4 Experiment

4.1 Experiment Setup

To obtain a preliminary evaluation of the effectiveness of ContracT to support semantic annotation, an experiment has been conducted on two real-life business contracts: Freight Agreement and Rent agreement. The contracts were selected in such a way as to have an approximately equal number of statements. Six persons attending the public law course at the University of Trento agreed to participate in the experiment. The participants were divided into two groups

[4] `https://www.isda.org/tag/smart-contracts/`

Table 3. Examples of EBNF concepts patterns

⟨*Asset*⟩ ::= (⟨*Resource*⟩ | ⟨*Amount*⟩)

⟨*Party*⟩ ::= (Agreement | ⟨*Signature*⟩) ⟨*Actor*⟩ ⟨*String*⟩

⟨*Role*⟩ ::= (⟨*Party*⟩ ⟨*String*⟩)

⟨*Currency*⟩ :: currency symbols

⟨*NegConnector*⟩ :: '`not`' ⟨*expr*⟩ | '`any`' ⟨*expr*⟩

⟨*Condition*⟩ :: '`if`' | '`then`' | '`else`' ⟨*expr*⟩

⟨*PositiveModal*⟩ ::= '`can`' | '`could`' | '`may`' | '`permit`' | '`right`' ...

⟨*NegativeModal*⟩ ::= '`have to`' | '`must`' | '`shall`' | ⟨*NegConnector*⟩
| ⟨*PositiveModal*⟩

⟨*Amount*⟩ ::= ⟨*Currency*⟩ ⟨*Digit*⟩.

⟨*TemporalExpression*⟩ ::= ((0[1-9])|(1[0-2]))((0[1-9])|(1⟨*Digit*⟩)|(2⟨*Digit*⟩)
| |(3[0-1]))(⟨*Digit*⟩4)

⟨*StartDate*⟩ :: = ⟨*TemporalItem*⟩

⟨*TypeOfDate*⟩ ::= ⟨*StartDate*⟩ | ⟨*EndDate*⟩ | ...

⟨*Obligation*⟩ ::= (⟨*Party*⟩ | ⟨*Role*⟩){1,2} ⟨*Action*⟩{0,1} ⟨*NegativeModal*⟩
| ⟨*Action*⟩{1,2} ⟨*Asset*⟩{1,2} ⟨*TemporalExpression*⟩ {0,1}.

⟨*Power*⟩ ::= (⟨*Party*⟩ | ⟨*Role*⟩){1,2} ⟨*Action*⟩{0,1} ⟨*PositiveModal*⟩
| ⟨*Action*⟩{1,2} ⟨*Asset*⟩{1,2} ⟨*TemporalExpression*⟩ {0,1}.

⟨*Exception*⟩ ::= '`only`' | '`but`' | '`except`' | '`without`' | '`limit`'
| '`restrict`' | '`exclusion`' | '`other than`' | '`unless`' ... .

and asked to individually annotate the contracts using an approach frequently used in experiments concerning the annotation of natural language. To mitigate the impact of the learning effect, in the first round, a group received the original text of the contract whereas the other group received the same text augmented with annotations generated by ContracT. In second round, the tasks were inverted using a different contract. The participants were requested to annotate the following concepts: *Party*, *Power*, *Obligation*, *Condition*, *Temporal condition*, *Asset Internal* and *External references*. Prior to the experiment, the participants were provided with the definition of the concepts to annotate. As a result, each participant annotated a contract from scratch and revised a contract previously annotated using ContracT (in Figure 2 an excerpt of the Freight Agreement contract annotated by the tool). To support insertion and modification of the semantic tags in the input documents, the participants were provided with a user-friendly web-based tool.

4.2 Experiment Results

The results of the experiment have been compared to the manual annotation performed by an expert, a PhD student in Law. This resulted in a total of 44 legal concepts to annotate in the Rent agreement and 39 in the Freight agreement. The outcome of the annotation of the participants is compared to the expert annotation for recall and precision in performing ContracT supported and manual annotation (Tables 4 and 5) as defined by [12]. The statistical measures are derived from the calculation of recall and precision, obtained from the comparison of the annotation of the experiment participants to the manual annotation performed by the expert. In the experiment, precision is the fraction of the concepts correctly annotated by the experiment participant out of the total concepts identified by the experiment participant. Recall is the fraction of the concepts correctly annotated by the experiment participant out of the total number of concepts in the text as identified by the expert. One of the problems in evaluating annotation results is related to the different lengths of the marked text units. Annotated concepts were considered correct when overlapping on the most meaningful words. The results of the experiment did not identify a reduction in annotation time when supported by ContracT, as the average annotation time was 23:41 minutes for ContracT assisted annotation compared to 22:16 minutes for manual annotation from scratch, with no significant differences between the two contracts. The support of ContracT appeared more meaningful in terms of quality of annotation for both contracts for most of the concepts to annotate. Except for precision in the annotation of power, the support of ContracT implied an improved or similar performance for all the concepts to annotate. Generally, precision is better than recall for all concepts. First of all, the results highlighted a high level of variability for the different concepts. That is not surprising as some concepts are more complex to be defined and henceforth to be identified (e.g., power).

Temporal conditions and obligations are the concepts better identified. Conversely, powers and conditions were identified with a higher level of difficulty and variability among the participants with the two concepts frequently interchanged or overlapping. Besides, seldom recurring elements such as internal and external references, were less frequently identified both in assisted and manual annotation. ContracT, despite the slightly better performance, incurred similar difficulties and strengths compared to human annotators in the identification of the concepts. Measures concerning assets and parties have been excluded from the statistics in Table 4 and Table 5 as they are all composed by just one word, frequently recurring and annotated correctly by all participants and ContracT. In this case, the support of ContracT resulted helpful in terms of reduction of time of execution and avoiding the annotation of repetitive words which may eventually lead to annotator fatigue. Moreover, it appeared that ContracT could be useful accounting for the difficulty of human annotators to recall all the instances of the concepts to annotate and not focus only on some of them. Furthermore, the lower performance in manual and assisted annotation for the Freight agreement compared to the Rent agreement may indicate that ContracT could be

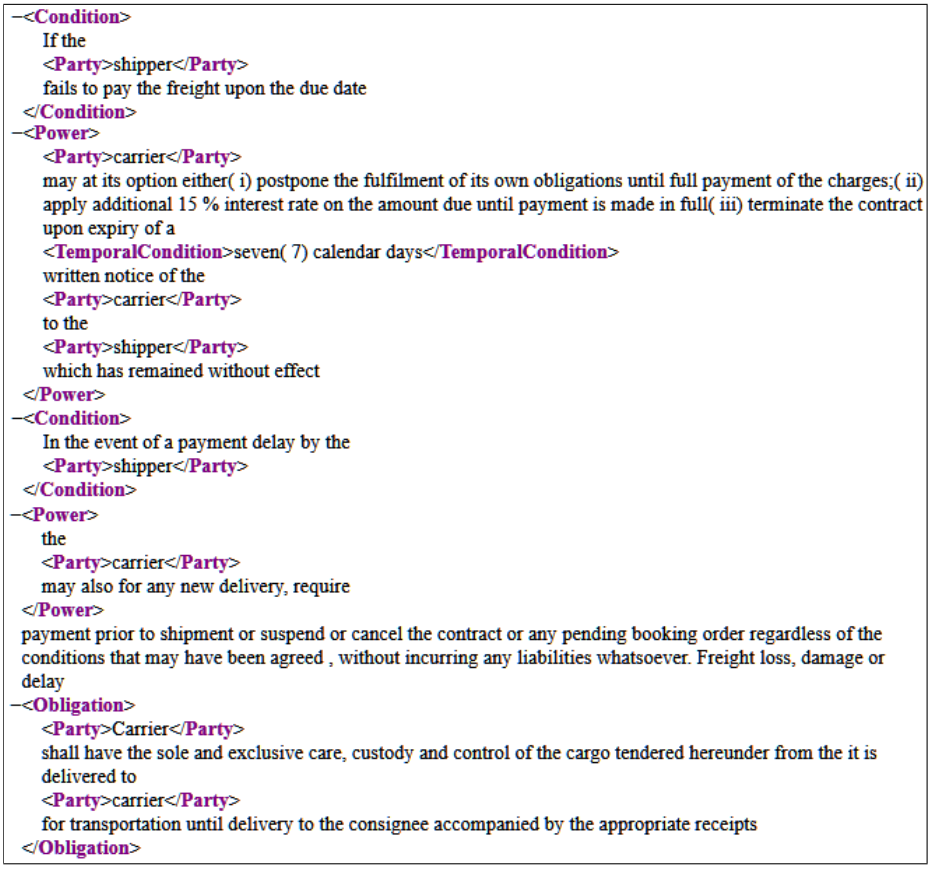

Fig. 2. An excerpt of the Freight Agreement contract annotated by ContracT.

more useful for contracts with a higher degree of complexity, a significant number of parties involved, lower familiarity of the annotators on the subject taking into account that the Freight agreement refers to a contract mostly used in B2B context. The experiment appears to underline the importance of recall over precision for practitioners in annotation of legal contracts as support to speed up the annotation process while avoiding false positive instances. For future developments, the results suggest using ContracT as a support to human annotators for more complex, time-consuming or recurring contracts (e.g. rental agreements for a rental agency) and for which annotation fatigue may lead to a significant decrease in quality of manual annotation. Furthermore, the refinement of the rules to identify the concepts presented in Table 5.

4.3 Threats to Validity

The objective of the experiment was to test the benefits of using ContracT to support a human annotator in the identification of legal concepts in the text. The results have been influenced by a few elements revealed by the participants in the interview after the experiment. Two participants admitted having resistance in using the web-based tool to annotate the contract and thus leading to longer annotation times for assisted annotation. One participant misinterpreted two concepts to annotate (i.e. obligation for power and viceversa). The misinterpretation significantly influenced the results of the experiment as in the evaluation of the annotation, a discrepancy between a human annotator and ContracT is always resolved in favour of the annotation performed by the human annotator.

In general, the most important threats to validity of the experiment is that the concepts to annotate have a significant level of ambiguity and are frequently interpreted in a different way by the different annotators. The annotation of ContractT is based on the recognition of specific patterns (e.g. modal verbs for obligations) but the vocabulary used may diverge in different contracts and thus lead to a decrease in the performance of automatic annotation. As such, an assessment of the performance of the tool is highly variable depending on the type of contract and on the participants to the experiment. For this reason, we consider the results of the experiment encouraging although they need to be confirmed by further testing.

External validity of our study is concerned with the generalizability of the results to other contracts. The results of our investigation are encouraging but preliminary, so they need to be confirmed by other experiments including a larger set of participants, both expert and non-expert, other types of contracts or relying on the implementation of contract templates.

Internal validity – factors affecting subject performance during the study is also very important. The skills of the subjects involved in the experiments were appropriate to the objective of our preliminary investigation. Moreover, there was no bias of the subjects towards the topics covered by the contracts used for the experiments.

5 Related Work

The possibilities to extract requirements from texts have been investigated for decades [2]. An increasing number of projects and commercial applications exist for metadata extraction in legal documents and semi-automation in the drafting and execution of legal contracts.

A number of projects – eBrevia,[5] LawGeex,[6] Prose[7] and Concord[8] – have been developed to support the extraction of information from contracts, to speed

[5] `https://ebrevia.com/`
[6] `lawgeex.com`
[7] `https://tryprose.com/`
[8] `https://www.concordnow.com/`

Table 4. Comparison of manual and assisted annotation of Rental Agreement and Freight Agreement contracts.

	Rental Agreement		**Freight Agreement**	
Measure	**Manual**	**Assisted**	**Manual**	**Assisted**
Recall	0.47	0.75	0.31	0.40
Precision	0.57	0.73	0.34	0.41
Fallout	0.40	0.53	0.46	0.49
Accuracy	0.51	0.71	0.40	0.47
Error	0.49	0.29	0.60	0.53
F-Measure	0.52	0.74	0.33	0.41

Table 5. Average Recall and Precision for concepts annotation of both contracts.

	Temporal cond. condition		**Power**		**Obligation**		**Condition**	
Measure	**Man.**	**Ass.**	**Man.**	**Ass.**	**Man.**	**Ass.**	**Man.**	**Ass.**
Precision	0.79	1.00	0.75	0.63	0.76	0.90	0.41	0.78
Recall	0.52	0.69	0.45	0.52	0.48	0.80	0.17	0.36
F-Measure	0.66	0.84	0.60	0.57	0.62	0.85	0.29	0.57

up their review or to ensure compliance. Among them, a few are based on Ricardian contracts in an approach to represent contracts that are readable both by a human and a machine [9]. However, none of these projects addresses the structure and content analysis steps where the objective is to transform a contract in natural language into formal specifications.

A number of academic studies refers to the extraction of requirement from legal documents and not legal contracts which represent, when available, the focus of this section. The task to identify the contract structure and text units has been undertaken in a project described in [3] using an experimental approach to the extraction of contract elements combining machine learning and manually written post-processing rules. The paper offers the idea that annotation rules can be learned from a benchmark and differently from ContracT does not rely on a structure model for the definition of the elements of a contract. Similarly, NómosT has been implemented with the objective to build models of law semi-automatically; at first the text of a law is annotated and then it is generated from a model [21]. In the more extensive context of legal analytics Stranieri et al. used handcrafted features and a Support Vector Machine (SVM) to segment French law [18].

[9] `https://www.schoenherr.eu/publications/publication-detail/ricardian-contracts-a-smarter-way-to-do-smart-contracts/`

To automatically identify concepts in a contract we have defined a conceptual model, aka an ontology for contracts. There are a number of upper ontologies for the legal domain, i.e. covering any kind of legal documents. The most comprehensive, a specialization of the Unified Foundational Ontology (UFO), is the eXtended UFO-L Core Legal Ontology (CLO) [16] based on Alexy's Theory of Fundamental Rights [1]. An increasing number of works focus on service contracts. Nardi et al. [13] propose a core reference ontology called UFO-S for a broad use for services. UFO-S is grounded in UFO and aims to provide a conceptual model for services which is domain independent. Griffo et al. [11] relies on UFO-S and extends ArchiMate EA language to reflect UFO-S ontology. However, the process of extraction of contract elements is not automated. Similarly, concerning service contracts, Griffo [10] et al. explores an approach bridging the gap between contract languages – for formal representations of contracts – and other approaches, as ArchiMate does not allow the representation of rights and obligations.

A further conceptual model to support the automatic extraction of software requirements from legal documents is proposed by [17]. The author attempts to harmonise the variety of semantic legal metadata proposed in RE to derive extraction rules based on constituency and dependency parsing. However, the proposed approach is not specialized for legal contracts and does not address structural analysis.

6 Conclusions and future works

In this paper we describe the process of translation of legal contracts in natural language into formal specifications. Focusing on the first step of the process, we defined two annotation models based on a contract ontology for structure and semantic content, respectively. The models were used in an exploratory prototype, ContracT. The application of the prototype and the experiment highlighted a variety of critical issues to be addressed in our future work before moving on to the other steps of the project. First of all, new patterns for both models have to be introduced. In particular, we will test the implementation of contract templates to support the annotation of specific types of contracts. Subsequently, we will focus on the other two steps of the process to generate a formal specification from legal contract – generate a skeleton of a specification that includes all and only the roles, obligations, powers, etc. of the legal contract and generate formal expressions for the constituents of obligations and powers from legal text fragments – using a specification language for contracts.

As regards specification languages a preliminary evaluation allowed to identify, among others, Business Contract Language (BCL) [8], a domain language permitting a significant level of abstraction required by business contracts; Azzurra [6], a specification language that models contracts as processes and is founded on social concepts as roles and commitments and Symboleo [16], the one which we will use, and where contracts are represented as a collection of obligations and powers.

References

1. Alexy, R., Rivers, J., Rivers, P.: A Theory of Constitutional Rights. Oxford University Press (2002), https://books.google.it/books?id=4G1U8JMSJDcC
2. Allen, T., Widdison, R.: Can computers make contracts? Harvard Journal of Law and Technology **9**, 25–52 (01 1996)
3. Chalkidis, I., Androutsopoulos, I., Michos, A.: Extracting contract elements. In: Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law. pp. 19–28. ACM (2017)
4. Clack, C.D., Bakshi, V.A., Braine, L.: Smart contract templates: essential requirements and design options (2016), http://arxiv.org/abs/1612.04496
5. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. MIT Press, Cambridge, MA, USA (2018)
6. Dalpiaz, F., Cardoso, E., Canobbio, G., Giorgini, P., Mylopoulos, J.: Social specifications of business processes with azzurra. In: 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS). pp. 7–18. IEEE (2015)
7. Governatori, G., Idelberger, F., Milosevic, Z., Riveret, R., Sartor, G., Xu, X.: On legal contracts, imperative and declarative smart contracts, and blockchain systems. Artificial Intelligence Law **26**(4), 377–409 (2018). https://doi.org/10.1007/s10506-018-9223-3, https://doi.org/10.1007/s10506-018-9223-3
8. Governatori, G., Milosevic, Z.: A formal analysis of a business contract language. International Journal of Cooperative Information Systems **15**(04), 659–685 (2006)
9. Griffo, C., Almeida, J.P.A., 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)
10. Griffo, C., Almeida, J.P.A., Guizzardi, G., Nardi, J.C.: From an ontology of service contracts to contract modeling in enterprise architecture. In: 2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC). pp. 40–49. IEEE (2017)
11. Griffo, C., Almeida, J.P.A., Guizzardi, G., Nardi, J.C.: Service contract modeling in enterprise architecture: An ontology-based approach. Information Systems p. 101454 (2019)
12. Manning, C., Schutze, H.: Foundations of statistical natural language processing. MIT press (1999)
13. Nardi, J.C., de Almeida Falbo, R., Almeida, J.P.A., Guizzardi, G., Pires, L.F., van Sinderen, M.J., Guarino, N., Fonseca, C.M.: A commitment-based reference ontology for services. Information systems **54**, 263–288 (2015)
14. Pejovic, C.: Civil law and common law: Two different paths leading to the same goal. Victoria University of Wellington Law Review 817 **32(3)**, 7–32 (2001), http://www.nzlii.org/nz/journals/VUWLawRw/2001/42.html
15. Sebastiani, R., Trentin, P.: Optimathsat: A tool for optimization modulo theories. Journal of Automated Reasoning **64**(3), 423–460 (2020)
16. Sharifi, S.S., Parvizimosaed, A., Amyot, D., Logrippo, L., Mylopoulos, J.: Symboleo: A specification language for smart contracts. In: 28th IEEE Requirements Engineering Conference (RE'20), Zurich, September 2020 (to be presented) (2020)
17. Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L., Dann, J.: Automated extraction of semantic legal metadata using natural language processing. In: 2018 IEEE 26th International Requirements Engineering Conference (RE). pp. 124–135 (Aug 2018). https://doi.org/10.1109/RE.2018.00022

18. Stranieri, A., Zeleznikow, J.: Knowledge discovery from legal databases, vol. 69. Springer Publishing Company, Incorporated (2011)
19. Szabo, N.: Smart contracts: building blocks for digital markets. EXTROPY: The Journal of Transhumanist Thought,(16) **18**, 2 (1996)
20. Zeni, N., Kiyavitskaya, N., Mich, L., Cordy, J.R., Mylopoulos, J.: GaiusT: supporting the extraction of rights and obligations for regulatory compliance. Requirements Engineering **20**(1), 1–22 (2015). https://doi.org/10.1007/s00766-013-0181-8, https://doi.org/10.1007/s00766-013-0181-8
21. Zeni, N., Seid, E., Engiel, P., Mylopoulos, J.: Nómost: Building large models of law with a tool-supported process. Data & Knowledge Engineering **117**, 407–418 (2018)