## D4.2

**Licenses and contracts: mining and reasoning techniques**

<table>
<thead>
<tr>
<th>Grant Agreement nº:</th>
<th>690974</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Acronym:</td>
<td>MIREL</td>
</tr>
<tr>
<td>Project Title:</td>
<td>Mining and REasoning with Legal texts</td>
</tr>
<tr>
<td>Website:</td>
<td><a href="http://www.mirelproject.eu/">http://www.mirelproject.eu/</a></td>
</tr>
<tr>
<td>Contractual delivery date:</td>
<td>31/12/2017</td>
</tr>
<tr>
<td>Actual delivery date:</td>
<td>31/12/2017</td>
</tr>
<tr>
<td>Contributing WP</td>
<td>4</td>
</tr>
<tr>
<td>Dissemination level:</td>
<td>Public</td>
</tr>
<tr>
<td>Deliverable leader:</td>
<td>INRIA</td>
</tr>
<tr>
<td>Contributors:</td>
<td>Data61, CORDOBA, UNIBO</td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690974
Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Partner</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>15/12/2017</td>
<td>Serena Villata</td>
<td>INRIA</td>
<td>Initial draft</td>
</tr>
<tr>
<td>1.0</td>
<td>30/12/2017</td>
<td>Serena Villata</td>
<td>INRIA</td>
<td>Final Version</td>
</tr>
</tbody>
</table>

Contributors

<table>
<thead>
<tr>
<th>Partner</th>
<th>Name</th>
<th>Role</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>INRIA</td>
<td>Serena Villata</td>
<td>Editor</td>
<td>Main editor of the document</td>
</tr>
<tr>
<td>Cordoba</td>
<td>Laura Alonso Alemany</td>
<td>Contributor</td>
<td>Writing of specific sections</td>
</tr>
<tr>
<td>UNIBO</td>
<td>Monica Palmirani</td>
<td>Contributor</td>
<td>Writing of specific sections</td>
</tr>
</tbody>
</table>

Disclaimer: The information in this document is provided “as is”, and no guarantee or warranty is given that the information is fit for any particular purpose. MIREL consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.
Table of Contents

Executive Summary........................................................................................................................................... 4

1 Introduction ................................................................................................................................................... 5

2 Mining license and contract documents...................................................................................................... 6

  2.1 Mining obligations, permissions and prohibitions from licenses – Licentia 2.0.............................. 6

  2.2 Combining Natural Language Processing Approaches for Rule Extraction from Contracts . 11

3 Reasoning on licenses and contracts ............................................................................................................ 16

  3.1 Licenses compatibility and composition................................................................................................. 16

  3.2 Reasoning about contract rules using Semantic Web formalisms and languages ..................... 22

4 Conclusions .................................................................................................................................................. 30

References ...................................................................................................................................................... 31
Executive Summary

The goal of this deliverable is to report about the achieved results and the ongoing work of the MIREL project about the first use case, i.e., mining and reasoning on licenses and contracts. More precisely, this deliverable reports about two approaches proposed to mine information from textual licenses and contracts, and then it reports about two different approaches to reason about the knowledge contained in licenses and contracts. The reported approaches to mine and to reason on contractual and license data are heterogenous from each other both with respect to the methodology applied to address the task and to the kind of data they exploit. Obtained results are satisfying, even if they cannot be considered as definitive. This deliverable underlines in particular the difficulty in dealing with such a kind of data, in particular regarding the issue of mining information from license and contract texts.
1 Introduction

Licenses and contracts are two fundamental kinds of document in the legal domain. More precisely, we define these two entities as follows:

**Licenses**: A license is an official permission to do, use, or own something (as well as the document of that permission or permit). A license may be issued by different authorities, to allow an activity that would otherwise be forbidden. It may require paying a fee or proving a capability. The requirement may also serve to keep the authorities informed on a type of activity, and to give them the opportunity to set conditions and limitations. A licensor may grant a license under intellectual property laws to authorize a use (such as copying software or using a (patented) invention) to a licensee, sparing the licensee from a claim of infringement brought by the licensor. A license under intellectual property commonly has several components beyond the grant itself, including a term, territory, renewal provisions, and other limitations deemed vital to the licensor.

**Contracts**: A contract is a voluntary arrangement between two or more parties that is enforceable by law as a binding legal agreement. Contract is a branch of the law of obligations in jurisdictions of the civil law tradition. Contract law concerns the rights and duties that arise from agreements. A contract arises when the parties agree that there is an agreement. Formation of a contract generally requires an offer, acceptance, consideration, and a mutual intent to be bound. Each party to a contract must have capacity to enter the agreement. Minors, intoxicated persons, and those under a mental affliction may have insufficient capacity to enter a contract. Some types of contracts may require formalities, such as a memorialization in writing.

In this deliverable, we report about progresses reached by the MIREL consortium to address the two research goals of (i) mining and (ii) reasoning on licenses and contracts information.

*Mining licenses and contracts* approaches have the aim to automatically extract information from textual documents containing licensing information or contractual information. This task may ease the analysis of such a kind of documents by humans, and it may also support the subsequent reasoning phase which requires some specific pieces of information as input of the reasoning process.

*Reasoning on licenses and contracts* approaches have the aim to support automated inferences and compliance checking on such a kind of knowledge. In this context, Semantic Web languages and formalisms (e.g., ontologies) are proved to be of fundamental importance, as well as standard logical formalisms as deontic logic and defeasible logic.

In the reminder of this document, we report about two approaches to mine licenses and contracts information, and two approaches to reason on such a kind of data. The presented approaches are promising as well as the obtained results. However, many open challenges still arise and they are the current objective of the MIREL consortium.
2 Mining license and contract documents

In this section, we describe two approaches proposed in the context of the MIREL project to mine information from licenses and contracts, expressed in natural language.

The MIREL partners involved in this line of research are CORDOBA, Data61, and INRIA.

2.1 Mining obligations, permissions and prohibitions from licenses – Licentia 2.0

Licentia is a suite of services to support data producers and publishers in data licensing by means of a user-friendly interface that masks to the user the complexity of the legal reasoning process. Since its development in 2014, Licentia offers two services:

1. The user selects among a pre-defined list those terms of use and reuse (i.e., permissions, prohibitions, and obligations) she would assign to the data and the system returns the set of licenses meeting (some of) the selected requirements together with the machine readable licenses' specifications,

2. The user selects a license and she can verify whether a certain action is allowed on the data released under such license.

Licentia relies on the dataset of machine-readable licenses (RDF, Turtle syntax, ODRL vocabulary\(^1\) and Creative Commons vocabulary\(^2\)) available at [http://datahub.io/dataset/rdflicense](http://datahub.io/dataset/rdflicense).

In the context of the MIREL project, CORDOBA and INRIA released a new version of Licentia: Licentia 2.0. This new version includes a new service whose aim is to translate natural language licenses into RDF ones, where in particular obligations, prohibitions and permissions are highlighted. In the reminder of this section, we describe the methodology and results obtained in mining licensing information to extract deontic components.

Licenses and data rights are becoming a crucial issue in the Linked (Open) Data scenario, where information about the use and reuse of the data published on the Web need to be specified and associated to the data. In this context, the legal texts describing the licenses need to be translated into machine-readable ones to allow for automated processing, verification, etc. Such machine-readable formulation of the licenses requires a high degree of reliability. For example, if the original license states that action A is forbidden and this prohibition is not reported in the RDF version of the license then this could lead to misuses of the data associated to that machine-readable license. For this reason, we need highly accurate performance in the task, to guarantee highly reliable outputs. In this scenario, human intervention is unavoidable, to establish or validate the correspondence between concepts in ontologies and expressions in natural language. We proposed to ease this dependency by optimizing human intervention through an active learning approach. Active learning techniques [1] aim to get powerful insights on the inner workings of automated classifiers and resort to human experts to analyze examples that will most improve their performance. We showed the boost in performances introduced by different

---

1 [http://www.w3.org/ns/odrl/2/](http://www.w3.org/ns/odrl/2/)
2 [http://creativecommons.org/ns](http://creativecommons.org/ns)
improvements on a classical, machine learning approach to information extraction. More precisely, in the experimental evaluation of our framework, we showed that active learning produces the best learning curve, reaching the final performance of the system with fewer annotated examples than passive learning. However, the standard active learning setting does not provide an improvement in our study case, where very few examples are available. Indeed, if we choose to annotate first those instances where the classifier shows more uncertainty, the performance of the system does not improve quickly, and, in some cases, it improves more slowly than if instances are added at random. In contrast, selecting for annotation those instances where the classifier is most certain (reversed uncertainty sampling) does provide a clear improvement over the passive learning approach. It is well-known that uncertainty sampling does not work well with skewed distributions or with few examples, in those cases, density estimation methods work best. We showed that using reversed uncertainty sampling in this particular context yields results in the lines of density estimation methods.

Active learning [1] is a more “intelligent” approach to machine learning, whose objective is to optimize the learning process. This optimization is obtained by choosing examples to be manually labelled, by following some given metric or indicator to maximize the performance of a machine learning algorithm, instead of choosing them randomly from a sample. This capability is specially valuable in the context of knowledge-intensive Information Extraction, where very obtaining examples is costly and therefore optimizing examples becomes crucial. The process works as follows: the algorithm inspects a set of unlabeled examples, and ranks them by how much they could improve the algorithm’s performance if they were labelled. Then, a human annotator (the so-called “oracle”) annotates the highest ranking examples, which are then added to the starting set of training examples from which the algorithm infers its classification model, and the loop begins again. In some active learning approaches, the oracle may annotate features describing instances, and not (only) instances themselves. This latter approach provides even faster learning in some cases. Different strategies have been applied to determine the most useful instances to be annotated by the oracle, including expected model change, expected error reduction or density-weighted methods [2]. The most intuitive and popular strategy is uncertainty sampling [3], which chooses those instances or features where the algorithm is most uncertain. This strategy has been successfully applied to Information Extraction tasks. Uncertainty can be calculated by different methods depending on the learning algorithm. The simplest methods exploit directly the certainty that the classifier provides for each instance that is classified automatically. This is the information that we are exploiting. However, we did not only use uncertainty sampling, but also the exact opposite. We explored both prioritizing items with highest certainty and with lowest certainty.

We followed the intuition that, when a model is very small, based on very few data, it can be improved faster by providing evidence that consolidates the core of the model. This is achieved by choosing items with highest certainty, because they also provide the lowest entropy with respect to the model, and can help to redirect wrong assumptions that a model with very few data can easily make. When the core of the model is consolidated, items with highest uncertainty should provide a higher improvement in performance by effectively delimiting with more precision the decision frontier of the model. This phenomenon, which lies at the heart of well-known semi-supervised learning techniques like self-training (or bootstrapping), has also been noted by approaches combining density estimation methods when very few examples are available, and uncertainty sampling when the training dataset has grown. Some approaches have been applied to fight the problem of learning with few examples, by finding the optimal seed examples to build a training set [4, 5]. However, these approaches are complex and difficult to implement, thus lie
beyond the capacities of the regular NLP practitioner. In contrast, the approach presented here is conceptually simple and easy to implement, as it is a wrapper method over your best-know classifier. We developed an active learning tool inspired on Dualist [6]. As in Dualist, we provide a graphical user interface for the human oracle to answer the queries of the active learning algorithm. The base machine learning algorithm is also a Multinomial Naive Bayes, but our method for ranking instances is uncertainty/certainty sampling based on the confidence of the classifier. Features can also be labelled, using Information Gain to select them, but sequentially with respect to instances, not simultaneously as in Dualist. As an addition, our approach allows for multiclass labeling, that is, an instance can be labelled with more than one class. Our active learning framework source together with the dataset is available at https://github.com/crscardellino/nll2rdf-active-learner.

As a base to our system, we used NLL2RDF, an Information Extraction system for licenses expressed in English, based on a (passive) machine learning approach [7]. The final goal of the system is to identify fragments of text that allow to identify a prohibition, a permission or an obligation (or duty) expressed by a license. When these fragments are identified, they are converted into an RDF machine-readable specification of the license itself.

The architecture of the system is based on a machine learning core, with an SVM classifier that learns from examples. Examples are manually assigned to one of a predefined set of classes associated to the licenses ontology. Many vocabularies exist to model licensing information. Some examples include LiMO, L4LOD, ODRS and the well-known Creative Commons Rights Expression Language (CC REL) Ontology. So far the Linked Data community has mainly used the CC REL vocabulary, the standard recommended by Creative Commons, for machine-readable expression of licensing terms. However, more complex licenses information can be defined using the Open Digital Rights Language (ODRL) Ontology, that allows to declare rights and permissions using the terms as defined in the Rights Data Dictionary. This vocabulary, in particular, has not been specifically conceived for the Web of Data scenario, but it intends to provide flexible mechanisms to support transparent and innovative use of digital content in publishing, distribution and consumption of digital media across all sectors. ODRL allows to specify fine grained licensing terms both for data (thus satisfying the Web of Data scenario) and for all other digital media. The ODRL vocabulary defines the classes to which each text fragment needs to be translated by the system. It specifies different kinds of Policies (i.e., Agreement, Offer, Privacy, Request, Set and Ticket). We adopted Set, a policy expression that consists in entities from the complete model. Permissions, prohibitions and duties (i.e., the requirements specified in CC REL) are specified in terms of an action. For instance, we may have the action of attributing an asset (anything which can be subject to a policy), i.e., odrl: action odrl: attribute. For more details about the ODRL vocabulary, refer to the ODRL Community group. The overall architecture of the Natural Language module of Licentia 2.0 is visualized below.
The core of the system is based on passive machine learning. Given some manually annotated instances, a classifier is trained to assign each text fragment to one or more of the given ontological classes, including the class of instances that is not associated to any meaning in the reference ontology (i.e., ODRL in this case), which is the case for the majority of sentences in any given license. In the first approach, a Support Vector Machine classifier was used. Texts were characterized by the unigrams, bigrams and trigrams of lemmas, obtaining an f-measure that ranged from 0.3 to 0.78 depending on the class, with 0.5 average. Later on we included bigrams and trigrams of words that co-occur in a window of three to five words. This last feature is aimed to capture slight variations in form that convey essentially the same meaning. These additional features increased the average accuracy of the system to 76%, kappa coefficient of .7. Although the performance of the system was fairly acceptable in general, it was not acceptable considering that we are dealing with legal information, and that an error in the system could cause an actual misuse of the data. Moreover, we found that it was difficult to improve such performances given the complexity of the task. Finally, we wanted to make it easier to port this system to other domains (i.e., other kind of legal documents like contracts, or policies), and to do that it was crucial to optimize the annotation effort (only 37 licenses where considered and annotated).

For all these reasons, we decided to adopt an active learning setting. In the active learning setting, we decided to use a different classifier that allowed easy manipulation of its inner workings, so that we could implement active learning tweaks easily. A Multinomial Naive Bayes (MNB) classifier was the classifier of choice. As a baseline to assess the improvement provided by the active learning approach to the problem, we assessed the performance of the MNB in a Passive Learning setting. The performance of the MNB by itself was quite below that of SVMs, of 63% (kappa coefficient of .6). Since it is well-known that bayesian methods are more sensitive to noise than SVMs, we applied Feature Selection techniques as a preprocessing to this classifier. We calculated the IG of each feature with respect to the classes, and kept only the 50 features with most IG, as long as they all had an IG over 0.001, those with IG below that threshold were discarded. Feature Selection yields an important improvement in performances, reaching an accuracy of 72%. This performance, however, is still below that of SVMs, and that is why we study a third improvement: one vs. all classification. As pointed out above, MNB is highly sensitive to noise, which seems specially acute in this setting where we have only very few examples of many of the classes. To obtain better models in this context, we applied a one vs. all approach, where a different classifier is trained to distinguish each individual class from all the rest. This, combined with a separate Feature Selection preprocess for each of the classifiers yields a significant improvement in performances, reaching an accuracy of 83%, with a kappa coefficient of .8. This
allowed us to use MNB as a base classifier for active learning, without sacrificing loss in performance with respect to the SVM baseline.

We evaluated the performance of different learning methods using a manually annotated dataset of licenses. The corpus consists of the original labelled set of 37 licenses, and an unlabeled set of 396 licenses. It is composed of software licenses, source code licenses, data licenses, and content licenses; they are public as well as private domain licenses.

Results showed that the “most certain” strategy performs consistently better than the passive and most uncertain strategies, improving performance with fewer instances. The other two perform comparably if the number of instances added at each iteration is high, and the “most uncertain” approach performs even worse than the passive approach (random) if instances are added one at a time for each iteration. These results confirmed our hypothesis that, for models inferred from very few training examples, maximizing the entropy of examples is not useful, while providing more evidence to define the core of the classes does provide an improvement in performance.

In an error analysis we can indeed see that the classes with most error are the smallest classes. This shows the benefit of growing the set of annotated examples, and thus the utility of an active learning approach for this task. The best strategy to grow from a very small dataset, with classes with very few instances, seems to be by choosing instances that are very similar to those already labelled, which provides a faster improvement in the performance of the classifier.

When examples are selected applying the “most uncertain” strategy are, they mostly belong to the “null” class, that is, they do not signal any of the classes relevant for the problem. Most of the sentences in licenses do not belong to any of the classes defined by the ODRL vocabulary and are classified as “null”. Providing examples for the class “null” is especially harmful for the resulting model for two main reasons. First, it grows the majority class, while small classes are kept with the same few examples, thus adding the problem of having an imbalanced dataset to the problem of having small classes with few instances. Second, the class “null” is composed by many heterogeneous classes that are not included in the ODRL vocabulary, and therefore its characterization is difficult and may be misleading.

Besides this configuration of classes, which can be found in very different domains, the domain of IE in licenses and normative text in general may be specially prone to an improvement of performance by labeling most certain examples first, because licenses and legal texts in general tend to be very formulaic, repeating the same wordings with very few variations, and small differences in form may signal differences in meaning, much more than in other domains, where differences in meaning are signalled by bigger differences in wordings.

These results are the object of a journal paper on the Semantic Web Journal describing Licentia 2.0. The paper will be finalized by February 2018.
2.2 Combining Natural Language Processing Approaches for Rule Extraction from Contracts

Applying deontic reasoning techniques to real world scenarios has to face the challenge of processing natural language texts. On the one side, all contracts and legal documents of public institutions and companies are expressed in natural language, and it is very unlikely to have a structured (possibly machine-processable) representation of the deontic conditions contained in such documents. On the other side, automated reasoning techniques need to process formal conditions to infer further information, or to check whether the observed behavior is compliant with such conditions, or whether a violation occurred. Rules are not always clearly identifiable in legal texts, and this task is difficult even for humans, becoming challenging for an automated system. Defining systems able to tackle this task in an automated way is a main challenge that received a lot of attention in the past years from the legal information systems community, and heterogeneous approaches have been proposed, e.g., [8, 9]. This interest is due, not only to the difficulty for humans to address such a task, but also to the fact that the task is extremely time consuming for humans, and (even partially) automating it to reduce the amount of work demanded to humans would become a valuable support.

Despite the huge number of proposed approaches, the problem of extracting rules or conditions from legal texts like contracts is still open. In this approach, we started from the observation that, given the difficulty of the task, the adoption of a single Natural Language Processing (NLP) approach to solve it would not lead to satisfiable results, as witnessed by very limited adoption of the current frameworks.

Thus, we adopted and combined a set of NLP techniques. More precisely, our framework for automated rules generation exploits the Stanford Parser to obtain the grammatical representation of the sentences, and WordNet to deal with the variability of the language in expressing the deontic components in natural language legal texts. We combined this syntactic-based rules extraction approach, relying on the well-known Stanford Parser, together with a logic-based approach, exploiting the Boxer framework [10] for the extraction of logical dependencies between chunks of text. The results of the evaluation of our combined framework on a selection of contracts in the telecommunications field show the feasibility of the proposed approach, and foster further research in this direction. The advantage of our approach is that there is no need to learn how to extract the rules building a huge annotated data set of legal documents as for machine learning approaches. The combined NLP approach implemented in our framework adopts several components to automatically generate rules from natural language legal texts. In particular, it exploits the following elements: i) a lightweight ontology describing the deontic linguistic elements allowing for the identification of the obligations, permissions, and prohibitions in legal texts; ii) a lightweight ontology describing how the natural language text is structured, and how punctuation can be interpreted for helping the extraction of rules; iii) a NLP library, namely, the Stanford Parser library\(^3\), used for parsing natural language sentences to retrieve their grammatical representation; iv) a Combinatory Categorial Grammar (CCG) parser tool including the Boxer framework, used for extracting logical dependencies between chunks of text from the document.

---

\(^3\) We decided to adopt Stanford Parser as it is the reference parser for parsing natural language sentences in English.
The resulting combined framework is an extension of the approach presented in [11]. In particular, the following drawbacks have been addressed with respect to [11]: i) the deontic ontology has been extended by extracting from WordNet all synsets related to the meaning of the Obligation, Permission, and Prohibition concepts. In this way, we are able to improve the precision of the term annotation activities with the deontic labels; ii) the set of the patterns used for detecting deontic rules has been enriched; iii) a parallel branch integrating the functionalities of the CCG parser has been integrated to analyze the text from a different perspective. The analysis results obtained by the CCG parser are then merged with the output of the NLP-only branch for extracting the final set of rules. The Figure below shows the pipeline of the proposed framework.

![Diagram of the proposed framework](image)

After the preliminary steps consisting in the extraction of the text from source documents, and the composition of the separated sentences generated by the extractor, the structured representation of the text follows two parallel branches implementing two different analysis techniques. In the lower branch, the modules of the Stanford NLP library are applied for tagging sentence content, and building the related tree for extracting the terms contained in each sentence. Then, the deontic ontology is applied to annotate each term with the appropriate label, i.e., obligation, permission, prohibition. Finally, the system looks for patterns within the terms set of each sentence in order to compose the rules. In the upper branch, instead, the CCG parser is applied to the full sentence to extract logic relationships between terms. Then, the output of the CCG parser is used for confirming the rules extracted through the lower branch, and for discovering new relationships between terms that have not been detected by applying the patterns adopted by the NLP parser.

The analysis of the text starts with the extraction of sentences of interest that are then used for the text analysis. The extraction of such sentences is done by exploiting the structured nature of the text that generally characterizes legal documents where a bullet-based representation is used for describing normative conditions. As first step, we mapped single text chunks contained in the bullet representation of the document to the lightweight ontology. In this way, we are able to manipulate a linked structure of the text easing the extraction of the full sentences. By considering the structured representation of the text as a tree, we reconstructed the set of full sentences to analyze by starting from the root of the tree and by concatenating, for each possible path, the text chunks found until the leaves are reached. Consider for instance an excerpt of the a contract showing the structured representation of one of the norms contained in the document:
By performing the mapping between the text and the lightweight ontology, the resulting assignments are the “Level 1” to the first chunk, “Level 2” to the second and third ones, and “Level 3” to the others. By navigating through the tree representation, the sentences extracted from the text are the concatenations of the following text chunks (based on the ids written at left of each chunk): “1-2”, “1-3-4”, “1-3-5”, “1-3-6”, “1-3-7”. The punctuation elements are used as regulators for deciding where to split sentences in case of complex structures. Sentences extracted at this step are then used for the extraction of the single terms.

The extraction of rules from natural language legal texts requires the use of tools able to provide a grammatical structure of the text that may be exploited for inferring the different components of a logical rule. The facilities available for having an effective representation of sentences are very limited. By analyzing the state of the art, one of the most prominent library is the one provided by Stanford. Such a library includes a Tree Parser able to produce a tree-based representation of each sentence and to tag them with grammatical prefixes. Moreover, the parser includes also a facility able to produce a set of grammatical relations explained which dependency elapses between two terms. The role of the parser is to work out the grammatical structure of sentences, for instance, which groups of words go together and which words are the “subject” or the “object” of a verb. The Stanford Tree Parser is a probabilistic parser using knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. Even if statistical parsers still make some mistakes in exceptional cases, they commonly work very well and, currently, they are the most suitable solution for a preliminary text analysis. In the proposed approach, we decided to use the Stanford NLP library for parsing the extracted sentences, and to use the produced output as starting point for terms extraction. Given the parsed version of each sentence, the next step consists to extract relevant terms from them. With “term” we do not mean a single word (or compound name) having a meaning in a vocabulary, but we mean a complex textual expression representing an entire concept. The extraction of the terms follows the identification of the subordinate sentences identified by the parser. In general, we interpreted the beginning of a new sentence (or a subordinate one) as the beginning of a new term with some exceptions based on the content of the generated tree. Some examples are i) if an extracted term starts with the expression “to VERB”, the term is automatically concatenated with the previous one, and ii) if an extracted term contains only one token, such a token is directly concatenated to the succeeding one. This mainly happens when tokens like “where”, “what”, etc. are parsed. Consider again the sample sentence where the analysis of the parsed representation leads to the identification of the following terms:
The first row is not marked as actual term but as “implicit” term. Indeed, some text chunks occur in many sentences. Such terms, independently by their eventual deontic meaning, are marked only once; while, for the other sentences, they are considered as “implicit” terms and they are not marked. The role of the “implicit” terms is to appear as antecedent of rules when, in a sentence, no terms are detected as antecedent, but consequent are identified. Two terms are identified here.

After the extraction of terms, they have to be annotated with the deontic tags of Obligation, Permission, and Prohibition defined in the deontic lightweight vocabulary. We assigned the deontic tags by applying a text processing approach. For each extracted term, we first verify if one of the lemmatized version of the labels of the vocabulary is present in the sentence; if yes, the term is annotated with the corresponding tag. A further check is performed to verify if, for example in case of verb, the label and the “not” auxiliary have been split during the term extraction in two consecutive terms. Indeed, if this happens, the identified deontic tag has to be changed. For instance, for the labels “must” and “must not” the deontic tags used are, respectively, the “Obligation” and the “Prohibition” ones. In the example, the only term in which a deontic element is identified is the implicit one that is annotated with the “Obligation” tag due to the label “must”:

--- Suppliers must demonstrate fairness, and courtesy, objectivity and efficiency, by
a: Acknowledging a Complaint within 2 Working Days of receipt
b: where the Complaint is made by email

The last step consists in the definition of the rules obtained by combining the extracted and annotated terms. For creating the rules, we applied a set of patterns to the terms in order to detect what are the antecedent and the consequent of each rule. Some of the patterns we defined are listed below:

```
[O] Term1
WHERE Term2  Rule: Term2 => [O] Term1

IF Term1  

[O] Term1 
UNLESS Term2  Rule: Term2 => [P] NOT Term1

[O] Term1
WHEN Term2
AFTER Term3  Rule: Term2 AND Term3 => [O] Term1
```

It is important to highlight that, in case a deontic tag is used for annotating an implicit term, such a tag is inherited by the first term following the implicit one. This happens because implicit terms are not taken into account for generating the rules. Finally, by considering the annotated terms and by applying the first pattern due to the presence of the “where” label, the generated rule is:
The CCG parser has been integrated for performing a logical analysis of each sentence in order to find relationships between the contained words. Indeed, semantic representations produced by the CCG parser, known as Discourse Representation Structures (DRSs), can be considered as ordinary first-order logic formulas and can be exploited to find semantic connections between each word extracted from the sentences. The aim of the integration of such a component is to support the NLP pipeline described above in detecting the relationships between the sentences from which the Stanford parser is not able to extract any information through the application of the pattern-based mechanism. The exploitation of such logical relationships allows to improve the general effectiveness of the rules extraction system.

Graph connections are exploited for two purposes. First, for sentences where deontic rules have been extracted by the NLP-only pipeline, we verified if the CCG parser finds relationships between the terms involved in the rule. Second, for sentences where deontic rules are not detected by the NLP-only pipeline, in particular by the pattern-based mechanism, if the CCG parser identifies logical relationships between the terms contained in the sentence, a new deontic rule is created and stored in the rules set.

Concerning the lower branch evaluation, the extraction of the sentences is the first performed task; the number of extracted sentences was 28 out of the same number of sentences contained in the gold standard. Therefore, concerning the first output the precision and recall of the system are 100%. The second task is the identification of the terms within sentences. The gold standard contains 65 terms extracted by the analysts; our system was able to extract 59 terms whose 49 are correct. Therefore, the obtained recall is 90.78% and the precision is 83.05%, with a F-Measure of 86.74%. Concerning the assignment of the deontic annotation, 47 out of the 49 correctly identified terms have been annotated with the proper deontic component, leading to a precision of 95.92%. The last step consists in determining which of the 36 rules contained in the gold standard have a counterpart in the automatically generated rule set composed by 41 rules. A rule r in the automatically generated set has a counterpart if there is a rule s in the manually generated set such that the proposition in the right hand side (or consequent) of s is mapped to the consequent of r. The number of rules satisfying this condition is 33 out of 36 with a Precision of 80.49% and a Recall of 91.67%. Finally, the last operation is to determine which extracted rules have a full correspondence with the manually generated rules: 24 of the automatically extracted rules have a corresponding rule in the manually generated rule set. This means that, as final assessment, we obtained a recall of Precision of 66.67%.

Concerning the upper branch evaluation, the first evaluation performed on the upper branch of the pipeline was the measure of the agreement between the rules generated by the lower branch and the ones inferred from the output of the CCG parser. By starting from the output of the CCG parser, we firstly verified if words belonging to different terms are directly related by one of the logical modifiers used by the CCG parser for representing relationships between words. After the verification of the all generated relationships, we computed how many of them exist also in the set of the rules generated by the NLP parser used in the lower branch. The set of relationships between terms extracted by the CCG parser contains 51 relationships and, by transforming them in rules, 35 of them have a counterpart (as defined in the previous paragraph) in the gold standard. With respect to the lower branch, the CCG parser was able to find 2 new rules having a counterpart in the gold standard. This means that the recall increased to 97.22% (35 rules detected out of 36), but the precision decreased to 68.63% due to the high number of relationships extracted by the CCG parser. Indeed, the CCG parser works at a more fine-grained
logical-linguistic level with respect to the NLP-only parser; therefore, the detection of relationships between terms that are not actually a rule is easier. The text we analyzed contains 35 prima facie clauses, and some of these rules require being decomposed in two sub-rules to fully capture the nuances of the conditions under which the obligations hold. Furthermore, we point out that the number of rules required to capture a norm could depend on the logical formalism used to reason with the rule. For example, if a condition of activation of an obligation is disjunctive, it is represented by two rules in the manually generated rule set. However, the disjunction could be represented by a single proposition encoding both of them. Thus, the number of rules required to model a normative clause depends on the underlying logic. This means that we can take as reference for the computation of the recall not the actual number of rules in the reference rule set, but the number of prima-facie clauses. In this case, the extracted rules cover 31 out of the 35 prima facie clauses. Concerning error analysis, in most cases incorrect rules depend on the incorrect identification of the propositions in the first step, or on the fact that the rules contain implicit terms in the left-hand side to be derived from the right hand side.

These results have been accepted for publication in the Springer volume containing the post-proceedings of the AICOL-2017 workshop. This contribution involves MIREL partners, i.e., INRIA and Data61, in collaboration with FBK Trento.

3 Reasoning on licenses and contracts

In this section, we describe two approaches proposed in the context of the MIREL project to represent and reason on the information contained in licenses and contracts.

The MIREL partners involved in this line of research are UNIBO, Data61, and INRIA.

3.1 Licenses compatibility and composition

In the latest years, several data hubs have being created by public bodies from single cities through to supra national organizations like the European Union with the final aim to improve the transparency and efficiency of such public bodies and organizations with respect to citizens. A key issue concerning such data is that it has to be machine-readable, in such way that automated processing and inference engines can be used to retrieve information from these huge amount of data, aggregate and interlink it, and exploit it to produce and publish new data, supporting the growth of the Web of Data [12, 13]. In this scenario, licensing and copyright information become more and more important. On the one side, it is important to associate to the data published on the Web some sort of machine-readable information concerning the terms of use and reuse of the data. This information will provide benefits to both data publishers and consumers, as for both of them is important to establish in a clear and indisputable way what consumers are allowed or not allowed to do with the data, as underlined in the “7 Best Practices for Producing Linked Data”1. On the other side, there is also the need to have automated frameworks able to deal with machine-readable licensing information and to reason over it ensuring the conformance with the normative document behind such licenses. This need is supported both by human data consumers that can thus be supported in reasoning over licensing terms in order to not misuse the data, and by artificial data consumers that can be equipped with such licenses reasoning module and thus access and use only those data released under certain conditions.
Several challenges arise to represent and reason over licensing information expressed in a machine-readable format, from the definition of lightweight vocabularies like ORDL and Creative Commons up to the definition of specific ontology design patterns [14, 15], to checking the compatibility of a set of heterogeneous licenses and compose them into a unique license in a compliant way. In this approach, we are interested on reasoning over licensing information in the following scenario: consider a set of licenses associated to heterogeneous data sets whose information items are returned together for consumption (e.g., resulting from a single SPARQL query over distributed datasets released under different licenses), and assume that these datasets provide the consumer with their own licensing terms. Our goal is to tell to the consumer whether these licenses are compatible with each other’s, and if so, to return it a so-called composite license that is compliant with the deontic component (i.e., permissions, obligations, and prohibitions) of each single license considered for the composition.

The problem of verifying the compatibility of a set of terms belonging to heterogeneous normative documents (i.e., contracts) and then combine them is not new: it has been studied in the contexts of services composition [16, 17, 18]. However, an automated framework answering the issues described above is nowadays still missing.

The reader may wonder whether the problem of dealing with licenses in the Web of Data is an actual problem, instead of begin premature. To confirm the actuality of the issue, we have crawled the Linked Open Data (LOD) cloud to understand how many datasets are licensed, and, what are the more popular adopted licenses. We have analyzed a total of 1140 data sets, and the results are visualized below.

![License Distribution](image)

Only the 17.6% of the data sets is not licensed at all or the license is associated in such a way that it is not identifiable as different from the published data released under such license (e.g., the license is represented as a single triple in the data set without a clear separation from meta-data like those contained in the VoID and the data itself). Among licensed material, more adopted licenses are Creative Commons Attribution, Creative Commons Attribution ShareAlike, CCZero, followed by Open Data Commons licenses PDDL and OdBL.
First, we rely on the deontic logic paradigm [19], a branch of logic which aims at representing and reasoning over the notions of permissible, obligatory, and prohibited. Each license is characterized in terms of what (and under what conditions) is permitted, prohibited, and obligatory. Assuming that distributed datasets provide data consumers with their own licensing terms, we proposed and evaluated a deontic logic semantics which checks in an automated way whether a set of deontic conditions are compatible. Second, in order to compose the deontic components due to the single licenses, we defined and evaluated two composition heuristics, namely AND-composition (i.e., the composite license entails a deontic effect if all the licenses composing it entail such deontic effect) and OR-composition (i.e., the composite license entails a deontic effect if there is at least one license that entails such effect, and no license prevents it). Moreover, we studied whether these two heuristics can be used together to provide a more suitable treatment of the different deontic components of the licenses. Finally, we have implemented and evaluated our framework for licenses compatibility and composition using the SPINdle defeasible reasoner. The Figure below shows the synopsis of the framework.

As described in our scenario, the data consumer queries a SPARQL endpoint to obtain some data (step 1). We retrieve the (possibly numerous) license(s) associated to the triples of query result (steps 1-2), and if more than one license is associated to such triples, we evaluate their compatibility using SPINdle. If the licenses are compatible, then we compose them into a unique composite license. We first translate the retrieved licenses from RDF (we adopted the ODRL vocabulary to express the licenses) to the SPINdle syntax (step 4) and then the whole theory, containing all terms in the licenses to be composed, is loaded in SPINDdle again. SPINdle manages two kinds of composition heuristics (step 5): AND-composition and OR-composition. These two heuristics can be combined together to produce the composite license (step 6), allowing in such a way a different treatment for each deontic component. Finally, we return to the consumer the query result together with the URI of the machine-readable composite license we have generated (step 7).

The limitations of the proposed approach are the followings: (i) we do not consider quantitative composition heuristics [16], and (ii) the framework does not consider the possible additional terms of the licenses. Finally, note that our application scenario does not deal with dual-licensing (alternative licenses for the same data), but we deal with the compatibility and composition of different licenses associated to different triples returned by a SPARQL query.
The reader may argue about the need of defeasible deontic logic to check licenses compatibility and compose the licenses information into a single license. Licenses are legal documents with the terms under which the licensee can use the licensed material. The terms can specify the conditions under which a particular use of (part of) the licensed material is permitted, forbidden and what are the obligations the parties involved in the licenses have to comply with. The same type of use, e.g., “reproduction of a record”, can be, for example, permitted under some circumstances and forbidden in others. Accordingly, we cannot simply use a proposition “ReproduceRecord (r)”; the standard semantic interpretation (in the usual Tarskian sense) is that “record r has been reproduced”, but it does not tell what is the legal status of that particular reproduction or record ‘r’. It does not specify whether the reproduction was permitted or forbidden or it was obligatory to reproduce it. This short discussion shows that we needed mechanisms to qualify the “legal” intension or status of the terms and conditions specified in a license. Deontic Logic offers such a mechanism with the help of the modal operators of obligation, permission and prohibition, that can be used to deontically qualify propositions. In addition, given a license or a set of licenses to be composed, we can use its inference mechanisms to deduce what are the obligations, permissions and prohibitions in force for a specific case.

The second aspect is that legal reasoning is by its own nature defeasible. Typically norms and provisions describe general principles and conditions, but at the same time they can establish exceptions to such conditions or principles. Defeasibility is the property that a conclusion can be retracted if evidence to the contrary is provided. Thus, defeasibility can be used to express in a simple and natural way exceptions, that is, by specifying that a rule overrides another rule. Consider again the example given above, where we have a general provision that prohibits the reproduction of a particular record. At the same time, we can have another provision that under particular conditions allows for the reproduction of the record. Without defeasibility we would obtain that the reproduction is at the same time forbidden and permitted. Defeasibility allows us to use the more specific conditions to permit the reproduction overriding thus the prohibition. In case of a single license the knowledge engineer modeling a license can use her knowledge of what are the provisions in the license providing exceptions to other provision to encode exception handling in the formal representation of the license. However, this is not possible when multiple licenses are composed together and we do not know in advance the content of the other licenses.

Several vocabularies have been proposed in the last years to model licensing and copyright information. In particular, the following interlinked vocabularies provide high-level descriptions of licenses, with a particular attention to the Web of Data scenario: LiMO, L4LOD, and ODRS. More complex licenses information can be defined with one of the digital Rights Expression Languages like ODRL or MPEG-21, a machine-readable language that allows declaring rights and permissions using the terms as defined in the Rights Data Dictionary. Beside the vocabularies mentioned above, other few vocabularies have been proposed in the literature to model, to different extents, licensing and copyright information. The Waiver vocabulary, for instance, defines properties to use when describing waivers of rights over data and content, where a waiver is defined as “a voluntary relinquishment or surrender of some known right or privilege”. As discussed by Heath and Bizer [13], “licenses and waivers represent two sides of the same coin: licenses grant others rights to reuse something and generally attach conditions to this reuse, while waivers enable the owner to explicitly waive their rights to a dataset”. Licenses are usually connected to the data through the VoID description. In particular, the Dublin Core vocabulary is usually adopted to associate licenses to resources though the property dc:license. Two further vocabularies that define licensing terms associated to the data on the Web are the Description of a Project vocabulary (DOAP), and the Ontology Metadata vocabulary (OMV). More precisely, DOAP specifies a property doap:license referring to the URI of an RDF description of the license the software is distributed under; OMV defines the property omv:hasLicense, which provides the
underlying license model, and a class omv:LicenseModel, which describes the usage conditions of an ontology. The attachment of additional information like rights, licenses, or provenance to RDF triples may be done also by adopting named graphs. Moreover, the W3C Provenance WG [20] has defined the kind of information to be used to form assessments about data quality, reliability or trustworthiness.

Apart from vocabularies to model licensing and copyright information, other approaches have been proposed to ease and support the representation of such information. Nadah et al. [21] propose to assist licensors’ work by providing them a generic way to instantiate licenses, independent from specific formats, and then they translate the license expressed in generic terms into more specific terms compliant with the specific standards used by distribution systems, i.e., ODRL and MPEG Rights Data Dictionaries.

In our contribution, we did not aim at proposing neither another representation format for licenses and copyright information nor an automated framework to translate licenses from natural language texts to RDF. We adopted the ODRL vocabulary for representing the internal structure of the licenses, as the same deontic components we used to reason in our logic are adopted in ODRL to represent the actions that are prohibited/permitted/obligatory by the license. The related work that presents more connections with the framework we proposed is in the area of contracts compatibility and composition for services composition. Comerio [16] analyses which kind of qualitative and quantitative heuristics can be used for contracts composition in the context of services composition. Qualitative heuristics include AND- and OR-composition heuristics plus the Constraining Value one where the most constraining value among the ones offered by the contracts of the services to be composed is included in the composite service. Quantitative heuristics, instead, include MIN, MAX, AVG and SUM (the composite contract offers the minimum (resp. maximum, mean, sum) among the values offered by the contracts of the single services involved in the composition. In our approach, we did not consider quantitative heuristics, and we proposed to combine the AND- and OR-composition heuristics to address the issue of licenses composition. Gangadharan et al. [17] address the issue of service license composition and compatibility analysis, specifying a matchmaking algorithm, which verifies whether two service licenses are compatible. In case of a positive answer, the services can be composed and the framework determines the license of the composite service. Truong et al. [22] address a similar problem concerning data contracts: in contracts composition, first the comparable contractual terms from the different contracts are retrieved, and second an evaluation of the new contractual terms for the data mash-up is addressed. There are several differences w.r.t. these approaches: (i) the application scenario is different (service composition vs. Web of Data); (ii) we allowed for a normative reasoning which goes beyond basic compatibility rules by exploiting normative conformance. However, common points are the idea of merging the clauses of the different licenses/contracts, and the use of RDF for licenses/contracts representation. Pucella and Weissman [23] propose a logic to check whether the user’s actions follow the licenses’ specifications. We did not perform enforcement, but we derived the deontic conclusions from a set of licenses to be composed, providing a proof theory to guide such composition. They do not deal with composition and do not provide a deontic account of licenses’ conclusions. Furthermore, their logic is not able to handle conflicting licenses. Krötzsch and Speiser [24] present a semantic framework for evaluating ShareAlike recursive statements. In particular, they develop a general policy modelling language, then instantiated with OWL DL and Datalog, for supporting self-referential policies as expressed by CC. We addressed another kind of problem that is the compatibility checking and the composition of the deontic components from single licenses into a composite license. Gordon [18] presents a legal prototype for analyzing open source licenses compatibility using the Carneades argumentation system. Licenses compatibility is
addressed at a different granularity with respect to our purpose, and licenses composition is not considered. Moreover, in our approach we proposed a defeasible logic for both checking the compatibility of a set of licenses, and combining them into a unique license, and Carneades has been shown to be closely related to defeasible logic [25]. Finally, the Licensiu tool provides a service to check the compatibility of a predefined set of licenses. Similarly, the European Data Portal (EDP) Licensing Assistant supports the user by providing legal information on the usage of a specific dataset in terms of licenses that apply to the dataset. The idea of these tools is in line with our purposes, but the compatibility checking is not done in an automated way as it is pre-calculated and thus static. Moreover, none of them is proving a “synthetic” composite license. Daga et al. [26] proposes a tool that aims at reducing the effort required for license selection by means of the definition of an ontology of licenses, organized by their relevant features, to support the user. It is also worth mentioning that we have integrated the formal machinery described in this paper concerning licenses compatibility in Licentia.

Our main contributions were:

- **Framework.** In our previous work, only licenses composition is considered, while in this framework licenses compatibility is also addressed. The logic is thus composed by two main parts, the former addressing compatibility checking and the latter addressing composition.

- **Logic and Heuristics.** We proposed an extension of Defeasible Logic, extending earlier works, to handle license composition. The current version is much more compact than the one in [27], and proposed a new and more intuitive reading of AND-composition and OR-composition than the one in [28]. Both improvements allow us to easily generate composite licenses.

- **Evaluation.** We presented an evaluation of the composition heuristics, which was absent in [27], and we implemented the licenses composer into the SPINdle reasoner, differently from [28] where the evaluation resulted from a transformation only. With respect to [28], in addition to the performance evaluations, an empirical study showing the pairwise compatibility of 41 licenses has been performed.

In our framework, licenses are represented using RDF. After a license has been selected, its associated RDF triples is loaded and translated into a defeasible logic formalism based on its attributes. If, however, more than one license has been selected, then in order to verify the conformability of the deontic component of each licenses, the translated defeasible theories are first composed into a single defeasible theory based on some transformations, before generating the heuristics required, as depicted in the Figure below.
3.2 Reasoning about contract rules using Semantic Web formalisms and languages

The Linked Data principles [12] provide a standard approach to weave a Web of data, linking datasets across the world and virtually in any domain. The semantic Web frameworks additionally provide standard means to publish data (RDF), ontological knowledge (RDFS and OWL schemata), and to query and reason on them (SPARQL). Despite existing approaches to model legal ontological knowledge, little work has been devoted towards the definition of an end-to-end framework to represent, publish and query ontological knowledge from the legal domain using such standards. We studied how Semantic Web frameworks could apply to the formalization, publication and processing of legal knowledge, and in particular, normative requirements and rules. A linked data based deontic representation and reasoning allows us to (a) rely on Web standard to represent, exchange and foster interoperability between deontic rule bases and reasoning systems, (b) rely on existing standards (e.g. SPARQL) and infrastructures (e.g. triple stores) to implement deontic systems, and (c) combine linked data and semantic Web reasoning and formalisms (e.g., OWL) with deontic reasoning to support more inferences.

We performed a search on LOV, a directory of Semantic Web vocabularies and schemata, to see how legal concepts are covered in published ontologies. Among the retrieved vocabularies, we
identified that:

- the General Ontology for Linguistic Description (GOLD) includes a “Deontic Modality” concept but it is essentially defined from a linguistic point of view with the goal to perform natural language analysis.

- the Public Procurement Ontology (PPROC) has the notion of “Contract additional obligations” which is a class limited to describing the additional obligations a contract requires.

- the Open Standards for Linking Governments Ontology (OSLO) includes an upper class “permission”, but attached to the role of an individual in a society.

- the notions of rights, permissions and licenses are mentioned in schemata such as Dublin Core, Creative Commons or ODRL but to describe the possible uses of a digital resource and they remain at a descriptive non-formalized level.

Current ontologies are often limited to a specific domain of application and have very shallow coverage of deontic concepts. They are not designed with the goal to support deontic reasoning above Semantic Web frameworks. Their primitives are designed to annotate resources with the goal of documenting or supporting some degree of interoperability, but they are not intended to support Semantic Web based reasoning and processing of the normative requirements and rules. Closer to our goal is the LegalRuleML Meta Model4 providing primitives for deontic rule and normative requirement representation (Permission, Obligation, Prohibition). We started from this model and extended it with a new ontology focusing on the deontic aspects, integrating notions from an existing abstract formal framework for normative requirements of regulatory compliance, and previous on modal defeasible reasoning for deontic logic on the Semantic Web.

Among the many approaches to design an ontology, the writing of motivating scenarios is a very usual initial step of specifications to capture problems that are not adequately addressed by existing ontologies. Our motivating scenario was to support the annotation, detection and retrieval of normative requirements and rules. We wanted to support users in information retrieval with the ability to identify and reason on the different types of normative requirements and their statuses. This would be possible through ontology population approaches, but the lack of an existing ontology covering these aspects slows this process, as well as the further development of more advanced applications in legal computer science.

In a second step of ontology specification, a standard way to determine the scope of the ontology

4 http://docs.oasis-open.org/legalruleml/legalruleml-core-spec/v1.0/csprd01/legalruleml-core-spec-v1.0-csprd01.html
is to extract from the scenarios the questions and answers it should be able to support if it becomes part of knowledge-based system. These so-called competency questions place demands on the targeted ontology, and they provide expressiveness requirements. The competency questions we targeted for this ontology are: What are the instances of a given requirement and its sub-types, e.g. obligation? Is a requirement violated by one or more states of affairs, and if so, which ones? Is a given description of rules and states of affairs coherent? Which rules, documents and states of affairs are linked to a requirement and how?

To support the competency questions and relying on definitions from LegalRuleML and deontic reasoning, we identified a set of core primitives for an ontology capturing the different aspects of normative requirements, and supporting the identification and classification tasks. We called that ontology Normative Requirement Vocabulary (NRV), and made it available and dereferenceable following the Linked Data principles. The namespace is http://ns.inria.fr/nrv# with the preferred prefix nrv respectively submitted both to LOV and to http://prefix.cc.

The top class of the ontology is the Normative Requirement which is defined as the set of the requirements implying, creating, or prescribing a norm. Then we have a number of upper classes to capture different features of the requirements:

- Compensable Requirement, Non Compensable Requirement, Compensated Requirement are classes of requirements with different compensation statuses.

- the classes Violable requirement, Non Violable Requirement, Violated Requirement and Compliant Requirement characterize the requirements with respect to their relation to a Compliance or a Violation.

- the other classes follow the same logic, and they distinguish requirements with respect to their perdurance, persistence, co-occurrence and preemptiveness.

Using these upper classes, we positioned and extended three primitives from the LegalRuleML Meta Model (i.e., Prohibition, Permission, Obligation), each one inheriting from the appropriate super classes we introduced. For instance, Permission inherits from Non Violable Requirement and Non Compensable Requirement, while Obligation inherits from Violable Requirement and Compensable Requirement. Specializations of these classes are then used to introduce the notions of Achievement, Maintenance and Punctual. For the complete list of classes and their definitions, we refer the reader to the online documentation available at the namespace URL. These primitives and definitions provide the taxonomic skeleton of our NRV ontology.

We provide some formalization details (ontological commitment) and their translation into OWL (computational commitment). We will use the TriG syntax for RDF. We captured the disjointedness expressed in the upper classes representing exclusive characteristics of normative requirements (compensable / non-compensable, violable / non-violable, persistent / non persistent):
We initially considered the disjointedness of a compliant requirement and a violated requirement, however this disjointedness is not global but local to a state of affairs and therefore it does not translate to a general disjointedness of classes, i.e., a requirement may be violated by a state of affairs but compliant with another one at the same time. However, this led us to capture this issue as a property disjointedness, since a requirement cannot be violated and be compliant with the same state of affairs at the same time:

\[
\text{Obligations are an example of non disjoint union between achievements and maintenances, since a punctual requirement is both an achievement and a maintenance:}
\]

\[
\text{An overview of the NRV ontology and its core primitives, in particular Prohibition, Permission, Obligation and a number of upper classes to capture different features of a Normative Requirement, is visualized below.}
\]
We could now be tempted to define a compliant requirement with the following restrictions:

```
1 :CompliantRequirement a rdfs:Class ; rdfs:label "compliant requirement"@en ;
2  rdfs:subClassOf :ViolableRequirement ;
3  owl:equivalentClass [ a owl:Restriction ;
4   owl:onProperty :hasCompliance ;
5   owl:minCardinality 1 ] .
6  owl:equivalentClass [ a owl:Restriction ;
7   owl:onProperty :hasViolation ;
8   owl:maxCardinality 0 ] .
```

However, we removed the second part (lines 6-8) of the restriction since it re-introduces a disjunction between the compliant and violated requirement classes. The notions of compliance and violation are not generally disjoint but only disjoint locally to a state of affair, i.e., a normative requirement can be violated and compliant at the same time but with respect to different states of affairs. However, OWL definitions cannot rely on RDF 1.1 named graphs, which we will use for representing states of affairs. Therefore we will need another mechanism to capture this kind of constraints.

Because we used disjoint unions, the ontology is in OWL DL, i.e., SHOIN(D), more precisely in the AL(U)C(H)RN family, i.e., AL attributive language, (U concept union), C complex concept negation, (H role hierarchy), R limited complex role inclusion axioms, reflexivity, irreflexivity, role disjointedness, and N cardinality restrictions. We decided to declare the signature of properties (e.g., hasViolation, hasCompensation) at the ability level (e.g., violable requirement, compensable requirement), and not at the effective status level (e.g., violated requirement, compensated requirement) because each status will be local to a state of affairs. Therefore, in the end, we avoided too strong restrictions and signatures. If we remove cardinality restrictions, unions and disjointedness, the ontology becomes compatible with OWL EL and OWL RL which could be interesting for implementations relying on rule-based systems, especially when we consider some extensions.

Using the LegalRuleML Meta Model and the NRV ontology we represented normative requirements as Linked Data. Let us introduce an example, a rule contract stating that employees of CSIRO must wear their badges:
The ability to define contexts and group assertions was one of the main motivations for having named graphs in RDF 1.1. The notion of state of affairs at the core of deontic reasoning is naturally captured by named graphs where all the statements of each state of affairs are encapsulated as RDF triples in a named graph, identifying that precise state of affairs. We provide here four examples of states of affairs respecting (2 and 3) or breaking (1 and 4) the rules of the normative statements described above. The core idea is to represent each state of affairs as a named graph typed as a factual statement of LegalRuleML.

Since the notion of named graph that appeared with RDF 1.1 is absent from OWL 2 and its constructors, we needed to implement the reasoning on states of affairs by other means. The SPARQL language is both a standard and a language able to manipulate named graphs so we propose to use SPARQL rules. We explored the coupling of OWL reasoning with SPARQL rules to formalize and implement some deontic reasoning. Description Logics (DL) support reasoning on the description of concepts and properties of a domain (terminological knowledge or T-Box) and of their instances (assertional knowledge or A-box). They are the basis of the Web Ontology Language (OWL). The classical inferences supported by DL are instance checking, relation checking, subsumption checking, and consistency checking. While these inferences are useful to reason about deontic knowledge (e.g., a compensable requirement must also be a violable requirement), they do not cover all the inferences we want to support here in particular deontic rules (e.g., a requirement is violated by a state of affairs if, during a specific period of time, a given constraint does not hold). These rules rely on complex pattern matching including, for instance, temporal interval comparison that go beyond OWL expressiveness. As a proof of concept, the following rules check the violation or compliance of the statements made by the previous states of affairs. The core idea is to add to each named graph of each state of affairs the deontic conclusions of the legal rules relevant to it. By relevant we mean here that the state of affairs describes a situation that falls under the application conditions of that legal rule. The following rules update compliance and violation for the CSIRO badge requirement:
The following rules update compliance for the state of affairs after violations were checked:

\[
\text{INSERT} \{ \text{graph } ?g \{ ?n \text{ a nrv:CompliantRequirement} \}} \}
\]
\[
\text{WHERE} \{ ?g \text{ a lrmlmm:FactualStatement} .
\text{?n a nrv:ViolableRequirement} .
\text{graph } ?g \{ ?n \text{ nrv:hasCompliance } ?g \}
\text{minus} \{ \text{graph } ?g \{ ?n \text{ nrv:hasViolation } ?g \} \} \}
\]
\[
\text{DELETE} \{ ?g \{ ?n \text{ a nrv:CompliantRequirement} \}} \}
\]
\[
\text{WHERE} \{ ?g \text{ a lrmlmm:FactualStatement} .
\text{?n a nrv:ViolableRequirement} .
\text{graph } ?g \{ ?n \text{ nrv:hasViolation } ?g \} \}
\]

To validate and experiment with the ontology, the Linked Data and the rules, we used two established tools:

- the latest version of the Protége platform and the reasoners it includes were used to check the NRV OWL ontology which was found coherent and consistent.

- the latest version of CORESE was used to load the LegalRuleML and NRV ontologies, the Linked Data about the rules and the states of affairs, and the SPARQL rules to draw the conclusions as shown in the Figure below for the two first states of affairs concerning speed limitation. More precisely, the figure shows how to extract of the quadruples (N-Quads) produced by CORESE after all the reasoning on the two first states of affairs concerning speed limitation showing one violated state (white background) and one compliant one (blue background). The columns indicate the named graph of the state of affairs (?g), the subjects (?lx), the predicates (?lp), and the objects (?lv) of the triples in this named graph.
The results of this research have been published at the international conference JURIX 2017, and the involved MIREL partners are INRIA and Data61.

4 Conclusions

In this document, we reported about the progresses achieved by the MIREL project with respect to the open research issue of “mining and reasoning on licenses and contracts”. More precisely, we have presented two approaches developed to mine legal texts representing licensing information and contractual information, respectively. Then, we have described two approaches developed to reason on licenses, i.e., checking the compatibility and composing licenses, and on contracts, i.e., modeling contractual knowledge about rules using Semantic Web formalisms and languages only. The presented approaches are promising and the obtained results foster further research in this direction. This will be one of the challenges that the MIREL partners will keep to tackle in the second half of the project.
References


