Legal datasets integration: keep it simple, keep it real

Gioele Barabucci, Angelo Di Iorio, Francesco Poggi

March 9, 2013

Abstract

Governments and public institutions are increasingly publishing their documents and data in freely available knowledge-bases, often as Linked Data silos. One of the goals is to connect and integrate heterogeneous legal sources, by exploiting Semantic Web technologies. Very often such integration is partial and difficult, due to the heterogeneity (and some conceptual flaws) of the datasets. In this paper we present ALLOT, a lightweight legal ontology based on the Akoma Ntoso ontological model, and we focus on how ALLOT can be used to align different datasets and to query all of them as a whole. The minimalistic approach of ALLOT makes dataset designers and users work with simple but effective concepts and queries.

1 Introduction

The number of freely available legal datasets is increasing at high speed. Citizens can easily access a lot of information about public administration, official budgets, parliamentary hansards, regulations and so on. Such openness brings them undeniable benefits in terms of transparency, participation, control and availability of new services.

Linked Data silos [26] make it also possible to connect such legal data, by exploiting Semantic Web technologies. The goal is to allow users to query multiple legal KBs as a whole. Much more meaningful information will be in fact available to the users. For instance, they could compare costs from different administrations, could evaluate and compare statistics about politicians from different countries, could compare productivity and speed of legislative processes, and so on.

Given the heterogeneity of such legal data - that differ not only for their storage format but also for their ontological model and terminology - this integration is not straightforward. The most common approach consists of three steps: (i) a meta-ontology is selected to create an uniform view on all data, (ii) statements compatible with that ontology are generated from each available dataset, and (ii) queries are written by using the conceptual structure of that ontology. Metalex and LKIF ontologies[13], for instance, can be successfully adopted for this task.

The choice of the which ontology is used as the intermediary ontology is a key factor for the success of this approach. The more that ontology is expressive, in fact, the more it will support sophisticated modeling and queries. At the same time, it has to be sufficiently easy to use. A full-fledged and complex ontology might become a slowing factor for such alignment.

The challenge of this paper is to provide designers a very simple ontology that, on the one hand, is easy to learn and, on the other, is powerful enough to allow sophisticated queries on multiple legal KBs. The critical aspect is that these queries are compact and easy to write. This simplicity will come with a trade-off: many queries will be easier to write using this ontology, but few will be either non expressible or hard to create. Our aim is to make the most common queries easy, leaving the less common queries to other more complex and heavy-weight ontologies.

In fact the paper presents ALLOT, a lightweight ontology that models legal entities such as people, events, roles, etc. ALLOT is only composed of a few classes and properties, being designed around the idea of Akoma Ntoso TLCs. Akoma Ntoso is an open legal XML standard for parliamentary, legislative and judiciary documents[3]. Its designers gave implementers the maximum degree of freedom, allowing them to use any ontology and any formalism, in an open scheme we refer to as "the Akoma Ntoso non-ontology". Nonetheless, the open-scheme prescribes some disjoint and very abstract classes, called Top Level Classes (TLCs). ALLOT formalizes TLCs in a set of OWL classes, then implements these classes in terms of other well-known ontologies (FOAF, LKIF, WordNet, etc.).

This makes ALLOT fully interoperable with Linked Data and a good candidate for querying heterogeneous KBs in a simple and effective way, as shown in the last part of the paper. The discussion on ALLOT gives us also the chance to discuss some strengths and weaknesses of the overall "Akoma Ntoso non-ontology" approach.

The paper is then structured as follows. In section 2 we go through the most prominent legal ontologies, focusing on their relation to external entities; in section 3 we give an overview of the Akoma Ntoso standard, its idea of non-ontology and the way it deals with external entities; in section 4 we present ALLOT and show, in section 5, how ALLOT can be integrated with existing legal ontologies; section 6 discusses how ALLOT queries have been successfully exploited to query heterogeneous datasets, and section 7 sketches conclusions and future research directions.

2 Legal and para-legal ontologies

The construction and integration of legal ontologies is a hot research topic. Legal ontologies are usually divided in two categories: core and domain-specific. Ontologies in the former group describe the "common conceptual denominator of the field" [19] and provide definitions of general legal concepts such as *norm*, *legal role*, *legal entity*, while ontologies in the latter group cover more context-oriented concepts such us *punishment* (useful when dealing with criminal law), *work of mind* (useful in copyright regulations) and similar abstract ideas peculiar to certain field. The overall picture is completed by top/upper ontologies that define even more abstract notions (e.g. *place*, *event*, *change*, *containment*) referred by core/domain ontologies.

We propose an orthogonal classification of legal ontologies centered around their relation to legal sources and external entities. In particular, we divide them in three groups: document-centric, content-centric and integration-centric.

2.1 Document-centric ontologies

We classify as *document-centric* those ontologies whose main goal is to describe the documental part of the legal documents. In particular, document-centric ontologies are used to model the evolution of legal sources, from their creation through their modifications and steps in the legal processes they belong to.

It is not a case that such ontologies usually exploit bibliographic models already established outside the legal domain. The most common approach relies on the IFLA's Functional Requirements for Bibliographic Records (FRBR) [27]. FRBR is a general model for describing the evolution of any document. It works for both physical and digital resources, and it is not tied up with a particular metadata schema.

The refinement of the FRBR levels in the context of legislative documents was first proposed in the MetaLex Ontology [12]. MetaLex is a generic and extensible framework for the XML encoding of the structure of, and metadata about, legal sources. It also includes an ontology that classifies bibliographic entities using the FRBR layers and defines different types of reference between such entities. Great relevance is given to events and actions in order to model the activities of creation, publishing and revision (amendment) of legislative documents. MetaLex also acts as the base for other ontologies, for example for the description of amendments in the data published by *legislation.gov.uk* initiative.

Some *document-centric* are *domain-specific*. For instance, the Ontology of Greek Public Administration [32] describes the documents and entities managed within the administrative office, and deals with peculiarities of this context.

2.2 Content-centric ontologies

Most of the existing legal ontologies can be classified as *content-centric*. These ontologies primarily define legal concepts carried by legal sources, these concepts are used to identify and make explicit the meaning that can be found in the legal text present inside a document. Some of the things they describe are the relations between different concepts (e.g. "abigeat" is a particular kind of "theft"). They are particularly important as they form the basis for legal knowledge acquisition and legal reasoning.

One of the most relevant to our work is LKIF-Core Ontology [23]. The Legal Knowledge Interchange Format (LKIF) is a format enabling the translation between legal knowledge bases, that use their own models and formalisms. Within the overall initiative, a central role is played by a core ontology that acts as main reference for such translation processes. The ontology defines 'basic concepts of law' and is organized in three levels, each covering orthogonal aspects of the domain of interest: Top, Intensional and Legal. The upper level is borrowed from the upper level of the LRI-Core ontology. The intentional level of LKIF-Core models the behavior of agents, while the legal one introduces domain-specific concepts such as rights, powers, norms, legal agents, etc.

CLO (Core Legal Ontology) [22] is another widely adopted content-centric core ontology. It defines legal entities and relations by exploiting classes and properties from the DOLCE foundational ontology [28]. The internal organization in three levels (Top-level, Core and Domain-specific), as well as its overall goal, makes CLO similar to LKIF-Core. The fact that the upper level uses a foundational approach is a very important difference. CLO is strongly based on the rich axiomatization and reification provided by DOLCE, and is an extremely powerful logically-sound framework.

While the ontologies mentioned so far define core concepts, other content-centric ontologies focus on specific domains. For instance, the ontology of Dutch criminal law, called OCN.NL [8], describes concepts related to depositions and hearings and "anchors" these concepts to more general ones defined in LRI-Core.

2.3 Integration-centric ontologies

The third group includes ontologies that give much relevance to the integration of legal sources and concepts to external entities, that exist outside the legal domain and are independent of it. These ontologies are widely used, for instance, to model organizations and people involved in the creation and consumption of legal information. The Parliament Ontology (PARL) [6] and Central Government Ontology (CGOV) [1] are two examples developed within the *data.gov.uk* initiative: they respectively model the organization of the UK Parliament and the Central Government. The nature of these ontologies led designers to reuse straightforwardly the FOAF (Fried of a Friend) ontology [14] to model people, their activities and their relationships with other people and documents. A similar approach has been used for the formalization of "Ontologia della Camera dei Deputati" (OCD) that describes the organization of the Chamber of Deputies in the Italian Republic [4].

In all these cases, designers need to align different ontologies and express relations among terms and concepts: terms are sometimes used in different contexts with slightly different meanings, in other cases apparently different terms refer to the same concept, in others a term is a specialization of another one, and so on. A very common solution, adopted by the aforementioned projects as well, consists of reusing the SKOS vocabulary [10] to express those relations.

This last point help us introducing another branch of legal ontologies we classify as integration-centric: the 'lexical ontologies'. A lexical ontology consists of a set of terms related to each other by semantic relations – such as hyperonymy, meronymy, instance-of, etc. – and aggregated in 'synset' when they refer to the same concept. Such ontologies, even if not rigorous and powerful as a formal ones, are able to capture the most relevant and used concepts in a given domain. One of the most relevant is Jur-IWN (Jur-ItalWordnet) [31]. The project aims at extending EuroWordnet, the European counterpart of WordNet, with legal information. It defines a large set of legal terms that can be recognized within texts and related to each other. These concepts are directly mapped into the main entities of CLO.

The integration between textual resources and core ontologies have also been successfully exploited for micro-ontologies merging [19]. In this paper, authors propose a construction method to build legal ontologies by merging small sets of entities and relations (i.e. micro-ontologies) that describe a narrow domain-specific context, and by aligning them to the CLO core ontology.

Notice that most of the ontologies we mentioned falls exclusively in one category. For instance, an ontology for criminal law might be classified as *content-centric* as it define concepts that are relevant even outside the legal documents they are referred by (for instance *crime*, *punishment*, *court*, etc.) but also *document-centric* as it defines notions like *act*, *article*, *comma* for modeling the legal sources where a criminal law was published in.

3 The Akoma Ntoso framework and non-ontology

The goal of this section is to recap the Akoma Ntoso guidelines, on top of which ALLOT is based, and to provide some background for the following discussion about ALLOT.

Akoma Ntoso is an open XML standard for parliamentary, legislative and judiciary documents [9]. The documents are strongly organized in layers, each addressing a specific problem. The *text* layer provides a faithful representation of the original content of the legal text, the *structure* layer provides a hierarchical organization of the parts present in the text layers, the *metadata* layer enriches underlying layers with ontological information so that semantic data can be shared and semantic tools can perform automatic reasoning on them. One of the peculiar features of Akoma Ntoso metadata layer is the ability to record multiple (and even contrasting) interpretations of the same legal text; this feature make Akoma Ntoso documents extremely rich sources of legal information, not only plain legal texts.

The Akoma Ntoso specifications, instead of forcing document authors to stick to one particular ontology, specify a very broad set of general guidelines that an ontology should conform to in order to be used in conjunction with Akoma Ntoso documents. We nickname these informal guidelines the Akoma Ntoso *non-ontology*.

3.1 TLC: Top Level Classes

The informal ontological structure defined by Akoma Ntoso for representing metadata is grounded in a basic set of concepts called Top Level Classes (TLC). The word "informally" is used because, on purpose, there is no mandated, exhaustive and shared ontology that defines these classes and the relation among them: what exists is a guideline that allows users (especially producers) of Akoma Ntoso documents to develop their own ontology according to their particular needs or to adopt one of the already existing ontologies, as long as compatible with the principles behind the TLC.

These top level classes do not have a formal definition, they only have a broad description, useful to identify in very general terms what is their purpose and how one TLC differs from another.

All this informality is needed to allow a great degree of flexibility in what can be expressed in the metadata layer of Akoma Ntoso documents, in order to adapt any legal document to many different ontological representation of concepts. It is the duty of a third party (e.g. the document creator or the document users) to associate a clear and formal semantics to each class using a specific formalism (e.g. OWL). This semantical detachment is an important feature that allows Akoma Ntoso to maintain documents understandable and consumable independently from the passing of time: future toolmakers ("The 'future toolmaker' is 10 years old now." [29]) will have clues about the intended meaning of a marker even in the unfortunate case the formal ontology is no longer available.

Any ontology that is used to model an Akoma Ntoso knowledge base must be compatible with its non-ontology. Compatibility with the non-ontology is a straightforward concept: an ontology is compatible as long as it is possible to associate every TLC to at least one of its "class" and this does not cause any "inconsistency". In this definition the terms "class" and "inconsistency" must be interpreted in a very liberal way. In lay terms, a class is a way to group individuals and inconsistency is a state of unrecoverable error in an automated reasoner. The exact meaning of these two terms depends on the technology used to implement the ontology: in OWL "classes" refers to a class and "inconsistency" to a Description Logic inconsistency, in Topic Maps "class" would be topic types and the meaning of "inconsistency" would depend on the tool used to define the consistency constraints.

The basic set of concepts required by Akoma Ntoso are the TLC: Top Level Classes. There are 14 top level classes: 10 main TLC used to describe external resources and 4 document-related TLC based on FRBR.

The ten main top level classes allow document creators to identify individual entities present in the document:

Concept any non-tangible notion or idea: e.g. "the approval of an act", "peace", "child".

Event something that "happened", "will happen", "may happen"or "have lasted": e.g. "World War II", "the coming into force of act 27", "Sunday 26th of August 2012".

Organization a recognizable group of individuals; organizations can be formal or informal, have a strong degree of internal organization or be completely anarchic, have their own name or be anonymous, have their own legal status or be impromptu groups: e.g. "the workers' union", "France", "the Socialist party", "the proponents of bill 103/32".

Person a human being, regardless whether they are alive or deceased, named or unnamed, fictional or real: e.g. "John Doe", "the person with ID RSSMRA72-H12L116B".

Place a location that can be referred to also using geographical coordinates: e.g. "the Rio river", "Marrakesh", "the entrance to the Black Forrest".

Process a series of actions or steps directed to some end: e.g. "the approval of act 317", "the election of the 11th president of the senate".

Reference a reference to a resource; usually the resources referenced are other documents, at one of the FRBR level.

Role a part played by a person, an organization or an agent in general, in a certain situation: e.g. "member of parliament", "speaker", "head of office", "bill proposer".

Term a word or group of words whose meaning is defined in a formal and precise manner: e.g. "opening sentence", "rebuttal", "impeachment".

Object anything that can referred to but that does not fit the other top level classes.

There are also 4 additional document-related top level classes that mimic the FRBR group-1 abstraction levels: Work, Expression, Manifestation and Item.

These TLCs have been devised in such a way that choosing which TLC fits which entity is an obvious decision in all cases except the most intricate and perverse.

Another of the main points behind the use of TLCs instead of complete and more refined ontologies is that this solution allows for a gradual evolution of the tools that operate over Akoma Ntoso documents. Having to describe each reference in terms of a TLC creates a minimal set of semantic data that can be used as a starting point to reason over these documents. With this small set of assertions even the simplest tools can answer basic queries like "what are the people referenced in this document?". With a little more effort, smarter tools can extract more information from the same references (for instance dereferencing the entities' URIs) and answer more complex queries like "what are the people referenced in this document that do not belong the the Lib party?". Tools that are even more smart can use external KBs to link these assertions to related assertions in other documents in other datasets, making it possible to reason over the all the ontological information made available.

The simplicity of the TLC model allows systems to start from the basics, using simple tools and simple ontologies to describe the resources they references, and later, when the need arises, to switch over to more powerful tools and complicated ontologies.

3.2 The Akoma Ntoso naming convention

The independence of Akoma Ntoso from a specific ontology is only one part of its design. Further flexibility, maintainability and long-term preservation are guaranteed by the way Akoma Ntoso deals with external entities. Instead of adopting an immutable naming structure, Akoma Ntoso implements a flexible scheme. Its naming convention is actually built on top of the *non-ontology* and TLC. Although Akoma Ntoso documents' authors are not forced to follow a precise ontology, in fact, they are still required to identify external entities with one of the top level classes described above. Users are required to indicate one of these classes for each external entity (as type of the reference). This fact creates a minimal basis of knowledge made available to tools that operate on these documents.

URIs are built using the following scheme:

`/ontology/{top level class}/{global name}`.

Each URI starts with the fixed part `/ontology/`. This first component identifies the pointed resource as an entity that is not a document (the Akoma Ntoso naming convention requires every document URI to begin with the identifier of the issuing country). The second part of the URI identifies which top level class the entity belongs to. The last part is a global name used to identify the exact entity we are referring to. Designers are free to use their own schemes but global names must, as the whole URI, be univoque identifiers: an entity can be referred to by more than one global name, but each global name must always identify the same entity, in every context.

A *resolution layer* completes the picture. Its role is to link the URIs used in the documents to concrete entities (when available). Two types of mechanisms can be used: a redirection service or a mapping dataset. The redirection service maintains internally a set of mappings from the published URIs to the latest or most refined versions of the entities being referenced, and uses chains of HTTP 3xx redirects to resolve URIs. The other mechanism consists in the publication of the mapping database in an open format: clients are informed of all the mappings between the published URIs and the URIs where the entity data can be found (for instance through the OWL `owl:sameAs` property) and can directly retrieve the right resources.

4 ALLOT: A Light Legal Ontology On TLC

ALLOT is a lightweight ontology we developed based on the Akoma Ntoso non-ontology following the associated guidelines and best practices. The ALLOT ontology, available at http://akn.web.cs.unibo.it/allot/, is meant to be used to describe in detail the references present in Akoma Ntoso documents, both documental and non-documental references. ALLOT can also used to bridge KBs extracted from Akoma Ntoso documents to KBs that use other ontologies such as Metalex or PARL (data.gov.uk). As expected, it uses the naming conventions discussed in the previous section and it is strongly tied up with the TLC.

ALLOT is composed of three layers, depicted in figure 1: *core* (where the TLC are declared and documented), *implementation* (where the TLC are implemented in terms

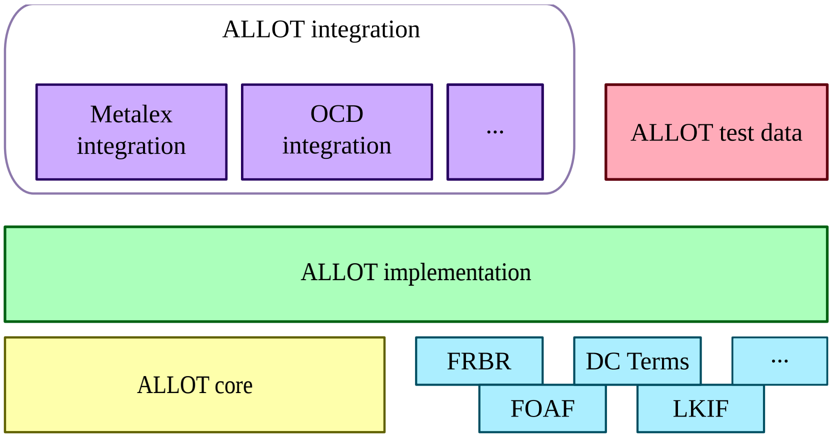

Figure 1: ALLOT layers

of well known ontologies such as FOAF or SKOS) and *external integration* (small ontologies used to align the ALLOT implementation to existing datasets based on other ontologies).

In addition to these layers, there is also a *test dataset* that contains test data and examples. This dataset contains many entities described using the ALLOT ontology. We used this dataset to check the consistency of our ontology during its development. This dataset also constitutes a set of examples that can be used by implementers to see how the various pieces of the ALLOT implementation fit together and how to create similar datasets from internal databases.

4.1 The core layer

The core layer is a basic transposition of the TLCs in OWL classes. It contains one OWL class for each TLC and few other properties. It also documents what is the intended use of each TLC. The classes and properties included in this layer form the basic classification made available by any ALLOT-based dataset; query writers can be sure that every knowledge base based on ALLOT contains at least this "skeleton" of organization.

From a conceptual point of view, this layer is equivalent to the non-ontology described in section §3.1.

From a more technical point of view, the ontology that realizes this layer exploits at least one of the advanced features of OWL2: name punning. All the top level classes are both OWL classes and individuals of the class `allot:Concept`. The rationale behind this is to allow designers to distinguish between, for example, the concept of what is a process (represented by the class `allot:Process` itself) from the concrete instances of these processes, i.e. the approval of act 345 (represented by individuals of type `allot:Process`).

4.2 The implementation layer

The implementation layer is the main component of ALLOT. Its role is to give a more precise definition to the top level classes originally defined in the core layer. This lay-

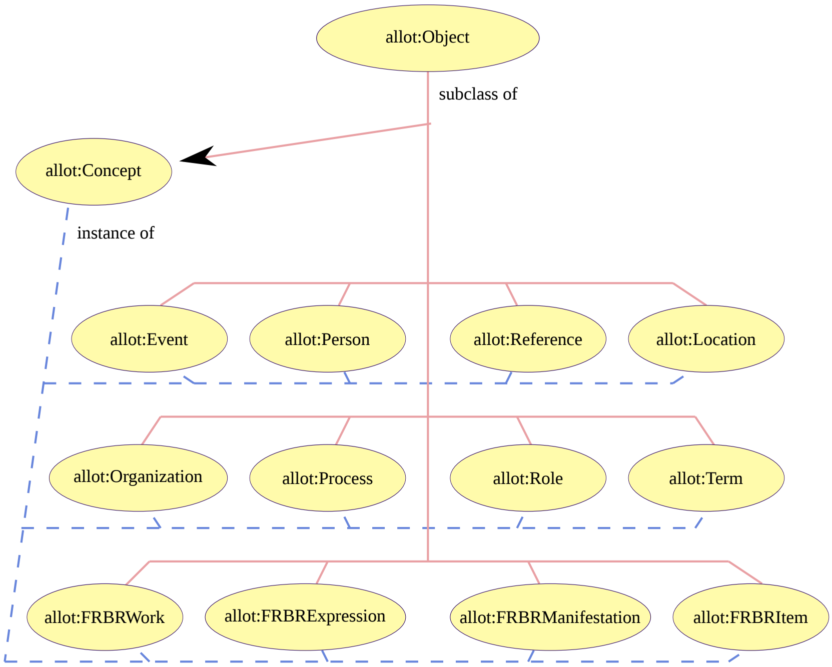

Figure 2: ALLOT core layer

ers makes ALLOT-based datasets interoperable with current semantic web and Linked Data datasets.

In practice, each top level class has been linked to similar classes from other ontologies, for example the TLC Person has been linked to FOAF Person and the birth of a person has been described in terms of BIO Events [17], as can be seen in figure 4.

Various kinds of "links" have been used: in some cases a TLC has been made subclass of other external classes; in other cases class or property equivalences have been used; in yet other cases external classes have been used as ranges or domains of newly defined properties. Various ontologies and vocabularies have been used: FOAF to describe people, BIO for biological events (i.e. birth and death), LKIF-core for intervals that can interoperate with other legal knowledge bases, PRO [33] for transient roles, FRBR [15] for documents, DC Terms [30] for various accessory datatypes, etc. In addition to these external ontologies, some definitions has been developed from scratch as we have found no published ontology that could satisfy our criteria for fitness.

The development of this layer has been quite challenging: most of the ontologies we wanted to use to increase the interoperability of ALLOT-based datasets cannot be used together without making the datasets inconsistent, as we will show later in this section. These incompatibilities and issues made impossible the direct use of these external ontologies. Some of the notable adjustments we made are related to the use context objects (modeled after the situations design pattern [21]) to keep track of properties whose value can change in the future, for instance names or political affiliations; an example of such context objects can be found in the ALLOT characterization of persons' names, as show in the example in figure 5. People in ALLOT are individuals of the FOAF:Person class. In the FOAF model each person can have more than one family name, this is accomplished attaching multiple `foaf:familyName` properties to

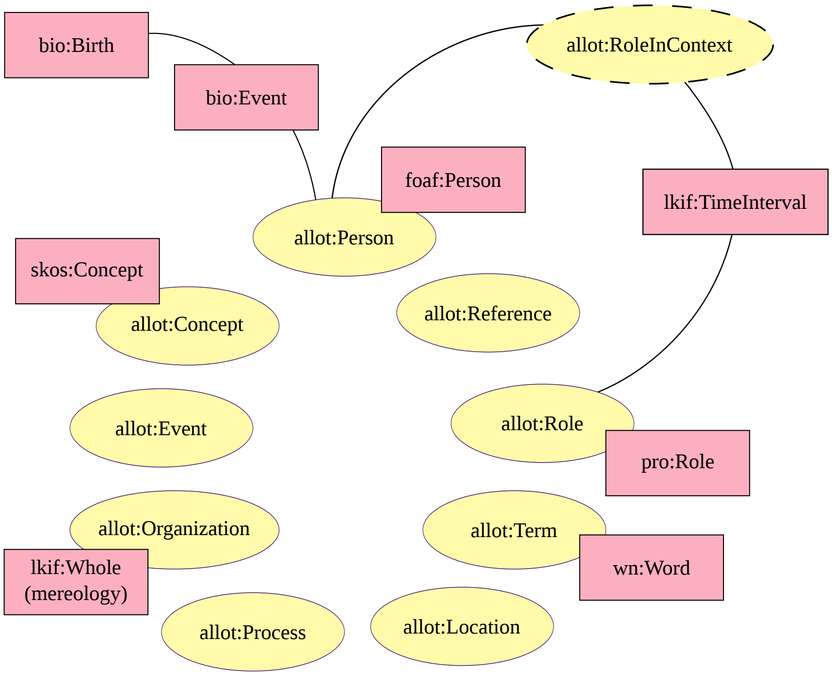

Figure 3: ALLOT implementation layer

```
Individual: rdfs:ke.susan-doe-i28i19k59382f
  Types:
    allot:Person
  Facts:
    bio:birth  test:birth-ke.susan-doe-i28i19k59382f,
    allot:name  rdfs:name-2-ke.susan-doe-i28i19k59382f,
    allot:name  rdfs:name-ke.susan-doe-i28i19k59382f,
    allot:role  rdfs:member-of-parliament-ke.susan-doe-↩
        ↪ i28i19k59382f
```

Figure 4: Example of an ALLOT Person individual

a FOAF:Person individual. The problem with this approach is that one does not know when each of these name is or had been valid. On the other hand, in the ALLOT model each person (that is still a FOAF:Person) has one or more contextualized name objects that hold all the name information (the context of such objects can be an interval of time, a place or a social environment). This creates a problem: if we were to reuse the `foaf:familyName` property to link the contextualized names to the string with the actual surname, the reasoners could conclude that a contextualized name object, being the subject of a `foaf:familyName` assertion, is also a FOAF:Person leading to an inconsistency. In this case we did not reuse the `foaf:familyName` property and implemented our own set of name properties. We could have created a property chain that stated the equivalence between `foaf:familyName` and `allot:name + allot:familyName` but OWL does not allow the use of data properties in property chains.

In some cases we were able to use existing ontologies without too many problems

```
Individual: test:ke.susan-doe-i28i19k59382f
  Types:
    allot:Person
  Facts:
    allot:name  test:name-ke.susan-doe-i28i19k59382f,
    allot:name  test:name-2-ke.susan-doe-i28i19k59382f

Individual: test:name-ke.susan-doe-i28i19k59382f
  Types:
    allot:NameInContext
  Facts:
    allot:interval  test:interval-name-ke.susan-doe-i28i19k59382f↩
        ↪ ,
    allot:givenName  "Susan",
    allot:familyName  "Doe"

Individual: test:name-2-ke.susan-doe-i28i19k59382f
  Annotations:
    rdfs:comment "The name Susan Doe got after her first marriage↩
        ↪ ."
  Types:
    allot:NameInContext
  Facts:
    allot:interval  test:interval-name-2-ke.susan-doe-↩
        ↪ i28i19k59382f,
    allot:givenName  "Susan",
    allot:familyName  "Smith"
```

Figure 5: Record for Susan Doe in ALLOT

because their underlying conceptual model was similar to that of ALLOT. For example, roles are described using a mixture of PRO roles and a pool of ad-hoc roles created for the legal domain. Similarly, we described the various TLC FRBR classes as equivalent to their respective classes in the SPAR FRBR ontology, without the need for any additional modification.

4.3 The external integration layer

The external integration layer of ALLOT is the piece that makes it possible to link knowledge bases modeled on ALLOT with knowledge bases modeled on other ontologies. It also allows the use of ALLOT as a single conceptual model to be used to query different knowledge bases.

Without links to other external ontologies, knowledge bases based on ALLOT would be hard to use in conjunction with other knowledge bases, exactly as it happens now when trying to use together data from a Metalex KB and a PARL KB. However, it is of paramount importance to have a mechanism in place that can make all these datasets talk to each other. Interoperability is especially important when dealing with current international law: many of the entities and documents referenced in local law are often derived from international treaties or directives, or they refer to the same supranational entities. All these common entities could act as joining points for datasets, making it possible to query not only national or regional knowledge bases but also foreign datasets. However, this kind of interoperability requires that the datasets share at least part of their underlying ontologies or that their ontologies have been aligned somehow. The external integration layer of ALLOT solves this problem acting as a N-to-1 converter, making it possible to query these datasets as if they where originally based on ALLOT.

The integration layer of ALLOT is composed of independent modules, i.e. small ontologies or SPARQL queries that describe how to align ALLOT to other existing ontologies, connecting classes and properties of an external ontology to their equivalent in ALLOT. Each of these modules is specialized in linking ALLOT to a precise ontology, using the most appropriate techniques. There are various techniques that can be used to link knowledge bases based on different ontologies: `sameAs` and `equivalentClass` relations, alignments made using specialized alignment ontologies, the generation of new statements using SPARQL CONSTRUCT, etc. We describe more in depth in section 5 the techniques we used to align ALLOT to the other existing ontologies; for each alignment technique we analyze its pros and cons and suggest what are the cases in which each should be used or avoided. More concrete examples of alignment and queries are showed later, in section 6 where we illustrate the alignment modules we developed for various legal datasets.

The main reason to have this integration layer separated from the main parts of ALLOT (i.e., the core and implementation layers) is the will to keep ALLOT small and understandable. In order to use ALLOT together with other knowledge bases, it is necessary to deal with many other ontologies; to support all these ontologies directly inside ALLOT would greatly reduce the simplicity of the ontology and make it hard to maintain. A similar approach has been adopted by other ontologies too. For instance, YAGO 2 [24]is an independent ontology that defines links and rules to refer to DBPedia, and LEXVO [18] ontology. Another reason for this separation is that it allows the development of independent modules that can be taken into account only as necessary, avoiding the possible incoherence that may arise when dealing with incompatible ontologies.

5 Linking ALLOT to existing ontologies

As briefly described in section 4.3, one of the the design goals of ALLOT is to make it possible to use it as a pivot ontology to connect heterogeneous legal datasets. In this section we expand on this point and discuss some issues and solutions we have found while aligning ALLOT to existing legal ontologies.

The alignment of ALLOT to other ontologies is not a trivial task, as in general are ontology alignments. As usual, the main difficulty arises from the fact that the ontologies already in use in publicly available KBs use very different concepts. Not only the concepts are different, often the use of certain concepts also implies the creation of intermediary entities that have no equivalent in other ontologies. The heterogeneity of ontologies required us to use different alignment techniques. In particular, we identified three classes of alignments to ALLOT: the use of alignment ontologies (e.g., SKOS), alignment via creation of new resources (i.e., using SPARQL CONSTRUCT) and integration of external datasets (e.g., using DBPedia to fill in the missing data).

5.1 Alignment via SKOS

There are some legal ontologies that include core concepts and properties close to the model of ALLOT. This is the case, for instance, of Metalex. Metalex focuses on encoding structure, editing process and metadata of legal sources and documents.

It gives a lot of importance to events and actions in order to model the activities of creation, publishing and revision of those documents.

On the other hand, Metalex covers only part of the TLCs and, thus, of the ALLOT ontology since it deals primarily with documents as bibliographic things and events. Moreover, the conceptual model that underlies the Metalex ontology is different from that of ALLOT: none of its classes, properties and implied relations have exact equivalents in ALLOT. The problem here (common with other ontologies too) is that, although there is certain degree of overlap between the concepts expressed in Metalex and those of ALLOT, none of these similarities match the technical and semantic constraints imposed by OWL relationships such as `equivalentClass` or `sameAs`.

In this case, it is useful to express in broad terms what are the relations between the classes of Metalex and those of ALLOT. That allows us to document and preserve some similarities between the ontologies. To do so, we follow the recommendations of 'Ontology matching' [20] and use SKOS properties. For example we state that `metalex:Event skos:narrowMatch allot:Event`. In other words, we state that, even if it is not possible to create a mathematically precise relation between these two classes, tools should know that the definition of what is considered a `allot:Event` is similar but narrower than that of what constitutes a `metalex:Event`.

While these SKOS annotations are not directly usable by DL reasoners, there are additional tools and API (such as [16]) that can help automated analysis system in the use of datasets that contain heterogeneous data, partly based on Metalex and partly based on ALLOT, allowing fuzzy matching between entities based on these two different ontologies. The same alignment techniques can also be used with other ontologies, for instance, to map the class `lkif:Role` (that indicates a legal role in the LKIF ontology) into the class `allot:RoleInContext` in ALLOT (that is narrower, as indicates a role possibly limited in its duration or in the space in which it is valid) or, similarly, to state that `lkif:Name` is broader (`skos:broader`) than `allot:RoleInContext`, since the latter limits the context, temporal interval or place a name is valid.

5.2 Alignment via construction

There are many other cases where the alignment to ALLOT is not straightforward, due to the huge differences in the models behind the datasets being aligned. A glaring example is OCD [4], the reference ontology used by the Italian lower chamber of the Parliament to made their data available. Many intrinsic differences between OCD and ALLOT make a SKOS-based translation between them practically impossible.

First of all, the ontology mixes the concept of person with that of role, a flaw that is quite common in legal datasets. For example, there are 10 different deputies whose name is "Giorgio La Malfa", all with different titles, descriptions and duties, yet they all have the same birth date. An example of such record is shown in figure figure 6, in this case his mandate for the 15th Italian legislature. Obviously the real meaning behind all these records is that there is a person called "Giorgio La Malfa" that has been elected 10 times in the Camera.

Ontologies structured like OCD suffer extreme redundancy of data and the lack of connection between related records. For instance, the record in figure 6 shows that the mandate of "Giorgio La Malfa" in the 15th Italian legislature (whose URI is `deputato.rdf/d3240_15`) is not linked in any way to the person "Giorgio La Malfa" (whose URI is `persona.rdf/p3240`), elected several times as Deputy. At the same time, information about the mandate itself are kept in a separate entity, identified by the URI `mandatoCamera.rdf/mc8_3240_19790617`.

```
Individual: ocd:deputato.rdf/d3240_15
  Types:
    ocd:deputato
  Annotations:
    foaf:gender "male",
    foaf:firstName "GIORGIO",
    foaf:surname "LA MALFA",
    dc:title "GIORGIO LA MALFA, XV Legislatura della Repubblica",
    rdfs:label "GIORGIO LA MALFA, XV Legislatura della Repubblica↩
       ↪ ",
    ocd:rif_leg <http://dati.camera.it/ocd/legislatura.rdf/↩
       ↪ repubblica_15>,
    ocd:rif_incarico <http://dati.camera.it/ocd/incarico.rdf/↩
       ↪ i332_3240_28_20070419>,
    ocd:aderisce _:http://dati.camera.it/ocd/deputato.rdf#ader1

Individual: _:http://dati.camera.it/ocd/deputato.rdf#ader1
  Annotations:
  ocd:componente _:http://dati.camera.it/ocd/deputato.rdf#mem2,
  ocd:rif_gruppoParlamentare <http://dati.camera.it/ocd/↩
      ↪ gruppoParlamentare.rdf/gr332>,
  rdfs:label "MISTO (03.05.2006-28.04.2008)"
```

Figure 6: OCD: a member of parliament record

The design of such an ontology shows clearly that it has been extracted directly from a DB whose design has, in turn, been developed "organically" over the years. This fact how some positive effects (it is probably very easy for the IT department of the Italian parliament to create and update these datasets) but also some grave drawbacks. For instance, the table with the list of Presidents of the Italian Republic has been turned into a class (direct subclass of `owl:Thing`) and their rows made entities of that class, confusing the concept of role with that of person, but also creating a very flat class structure that does not reflect the relation between the various classes of roles and events.

In such a scenario, we use a different alignment technique: linking the two ontologies using mini-datasets generated on the fly through SPARQL CONSTRUCTs. Basically, a SPARQL query is used to interrogate the OCD repository. Once the data is returned it is converted into a set of ALLOT-based assertions, that model correctly events and roles, and fixes the above mentioned issues.

To illustrate this technique, we show how to construct information about a member of parliament. As already pointed out, the crucial point is that the OCD dataset contains multiple instances, one for each time that person has been elected. This information is actually redundant and not completely matching the ALLOT model: one should rather define a role "Deputy" and indicate the context (legislature and time period) a person was given that role. ALLOT assertions stating that a person was given a role (of Deputy) in a specific time frame can be generated straightforwardly. The TLC classes involved in such statements would be: `allot:Person`, directly mapped into `ocd:person`, and `allot:Role`, to indicate the role of Deputy, linked by the property `allot:RoleInContext` of the class `allot:Person`. The snippet in figure 7 shows the SPARQL query to achieve this goal with the OCD dataset.

```
PREFIX  ocd:    <http://dati.camera.it/ocd/> PREFIX ↩
    ↪ ocdlegisl:  <http://dati.camera.it/ocd/↩
    ↪ legislatura.rdf/> PREFIX allot:    <http://akn.↩
    ↪ web.cs.unibo.it/allot/> PREFIX  lkiftime: <http↩
    ↪ ://www.estrellaproject.org/lkif-core/time.owl#>

CONSTRUCT {
  _:br_member a allot:Person;
    allot:givenname ?it_firstName ;
    allot:familyname ?it_surname ;
    allot:name ?it_name ;
    bio:event [
      a bio:Birth;
      bio:date ?birthDate
    ] ;
    allot:role [
      a allot:RoleInContext ;
      allot:roleType allot:member_of_parliament ;
      allot:interval [
        a lkiftime:Interval
      ] ;
    allot:legislature rdfs:it.legislature-16
  ] .
  rdfs:it.legislature-16 a allot:Legislature ;
    rdfs:comment ?leg_label .
} WHERE {
  ocdlegisl:repubblica_16 rdfs:label ?leg_label .
  ?it_dep a ocd:deputato;
    ocd:rif_leg ocdlegisl:repubblica_16 ;
    foaf:firstName  ?it_firstName ;
    foaf:surname ?it_surname .
  ?it_person a foaf:Person ;
    foaf:firstName  ?it_firstName ;
    foaf:surname ?it_surname ;
    bio:event ?it_bioev .
  ?it_bioev bio:Birth ?it_birth .
  ?it_birth bio:date ?it_birthDate .
  BIND (SUBSTR(?it_birthDate,1,4) AS ?year) .
  BIND (SUBSTR(?it_birthDate,5,2) AS ?month) .
  BIND (SUBSTR(?it_birthDate,7,2) AS ?day) .
  BIND ( xsd:date(CONCAT(?year, "-", ?month, "-", ?day))↩
      ↪  AS ?birthDate ) .
  BIND ( CONCAT(?it_firstName, " ", ?it_surname) AS ?↩
      ↪ it_name) }
```

Figure 7: Example of SPARQL CONSTRUCT query to align OCD to ALLOT

The same conversion, for instance, can be applied to model the government assignments (processing the property `ocd:rif_incaricoGoverno` connected to the class `ocd:incaricoGoverno`) or the membership of a Deputy to a Parliamentary group and/or to a Parliamentary committee. Space limits prevent us to go into details of such alignment. Some interesting issues, for instance, have also been raised by the translation of places and biographic information, as well as the normalization of missing data. Further details can be found in the online version of this ontology module.

We experimented similar alignment techniques, based on SPARQL CONSTRUCT, to work with the dataset published by the Brazilian Parliament [2]. This dataset contains personal information on the members of the Parliament and Senate in RDF format, and information on their political activity such as committee memberships, speeches and propositions in raw format (.csv). The first step consisted of expressing all these data in RDF. To do so, we defined a schema mapping between the CSV and RDF and used OpenRefine [5]. Then, we run SPARQL queries to extract data and express them in ALLOT.

The snippet in figure 8, for instance, shows the SPARQL query to construct triples about a member of Parliament, corresponding to the one shown in figure 7 and working on OCD. Prefixes declarations are not shown for the sake of synthesis.

```
PREFIX  allot:    <http://akn.web.cs.unibo.it/allot/>
PREFIX  dbpedia:  <http://dbpedia.org/ontology/>
PREFIX  lkiftime: <http://www.estrellaproject.org/lkif-↩
    ↪ core/time.owl#>

CONSTRUCT {
  _:br_member a allot:Person;
    allot:name ?br_name ;
    bio:event [
      a bio:Birth;
      bio:date ?br_birthdayParsed       ];
    allot:role [
      a allot:RoleInContext ;
      allot:roleType allot:member_of_parliament ;
      allot:interval [
        a lkiftime:Interval ] ;
      allot:legislature rdfs:br.legislature-54
    ] .
  rdfs:br.legislature-54 a allot:Legislature ;
  rdfs:comment "54th legislature of brazilian parliament↩
      ↪ " .
} WHERE {
  ?br_person a dbpedia:Person;
    foaf:name ?br_name;
    foaf:birthday ?br_birthday
  BIND (SUBSTR(?br_birthday,7,4) AS ?br_year) .
  BIND (SUBSTR(?br_birthday,4,2) AS ?br_month) .
  BIND (SUBSTR(?br_birthday,1,2) AS ?br_day) .
  BIND ( STRDT(CONCAT(?br_year, "-", ?br_month, "-", ?↩
      ↪ br_day), xsd:date) AS ?br_birthdayParsed )
}
```

Figure 8: Example of SPARQL CONSTRUCT query to align the Brazilian Parliament dataset to ALLOT

This dataset allows us to also point out another possible issue when aligning legal ontologies to ALLOT. Some data, in fact, are published in the dataset but in a format that cannot be directly translated into RDF. A glaring example is shown in figure 9. The data about the election a member of Parliament started her/his mandate, in fact, are encoded as a single XML frament within a literal.

```
<polbr:election
  rdf:parseType='Literal'>
  <timeline:atYear>2010</timeline:atYear>
  <foaf:name>LUPION</foaf:name>
  <biblio:number>2505</biblio:number>
  <pol:party>DEM</pol:party>
  <pol:Office>Deputado Federal</pol:Office>
  <geospecies:State>PR</geospecies:State>
  <earl:outcome>Média</earl:outcome>
  <spinrdf:Union> PSDB - PP - DEM - PPS - PRB</↩
      ↪ spinrdf:Union>
  <polbr:unionParties> PRB / PP / PPS / DEM / PSDB↩
      ↪ </polbr:unionParties>   <polbr:situation>↩
      ↪ APTO(Deferido)</polbr:situation>
  <polbr:protocolNumber>142462010</polbr:↩
      ↪ protocolNumber>
  <polbr:processNumber>797-74.2010.6.16.0000</↩
      ↪ polbr:processNumber>
  <polbr:CNPJ>12.172.387/0001-42</polbr:CNPJ>
</polbr:election>
```

Figure 9: Structured information embedded within RDF literals

This is a poor design choice, although it is not a problem for the current usages and goals of the dataset. On the other hand, a pre-processing phase can be implemented in charge of extracting these fragments, parsing data and express the same information in RDF. That is not optimal but it makes it possible to collect all data and to align the dataset to ALLOT, without resorting to other sources.

5.3 Alignment via construction and external integration

In the previous cases all the information needed for alignment was already in the datasets and SPARQL queries could synthesize new assertions, apart from specific issues that could be solved through pre-processing as just discussed.

In other cases, the data contained in original dataset is not enough: some datasets miss part of the required information, for example the dates of the mandates or the biographic data of the member of parliaments. This lack of information may be caused by various factors, sometimes it is just the effect of bad design, in other cases it may be connected to privacy issues (for example, in the case of biographic data) or the fact that some datasets are published while still in a prototypical stage.

This is the case of the data about the parliament and its members published on *data.gov.uk*. As already discussed, this is an initiative of the UK government for the publication of various pieces of data about government, parliament, civil issue, transportation and so on. The data about the parliament are published based on the PARL ontology [6].

For each member of the parliament *data.gov.uk* publishes only the name and the party membership. Other information, such as birth date or the duration of the mandates, are not available. The idea is to retrieve missing information from external sources.

In fact, we integrated the personal information such as the date of birth using DB-Pedia [11], and the information on their parliamentary activity such as the debates and the committee memberships using TheyWorkForYou [7]. The main problem we had to face was that these the datasets were not natively linked to each other, so we had to join them using some common information as keys (such as the given name and surname).

After adding these data described in PARL, it is possible to use SPARQL CONSTRUCT and to proceed with the integration to ALLOT in a manner similar to what we did with the OCD alignment. For example, the SPARQL excerpt in figure 10 shows the alignment of the some biographic information such as name, surname and date of birth.

```
PREFIX  parl:   <http://reference.data.gov.uk/def/parliament/> ↩
    ↪ PREFIX dbpprop:  <http://dbpedia.org/property/> PREFIX ↩
    ↪ dbpedia:  <http://dbpedia.org/resource/> PREFIX allot:    <↩
    ↪ http://akn.web.cs.unibo.it/allot/> PREFIX  lkiftime: <http↩
    ↪ ://www.estrellaproject.org/lkif-core/time.owl#>

CONSTRUCT {
  _:uk_member a allot:Person;
    allot:givenname ?firstname ;
    allot:familyname ?lastname ;
    allot:name ?nameConcat ;
    bio:event [
      a bio:Birth;
      bio:date ?birthdate ] ;
    allot:role [
      a allot:RoleInContext ;
      allot:roleType allot:member_of_parliament ;
      allot:interval [
        a lkiftime:Interval ] ;
      allot:legislature rdfs:uk.legislature-55
    ] .
  rdfs:uk.legislature-55 a allot:Legislature ;
    rdfs:comment "55th Parliament of the United Kingdom" .

} WHERE {
  ?uk_member foaf:name name ;
    foaf:lastName ?lastname ;
    foaf:firstName ?firstname ;
    parl:partyMemberOf ?party .
  ?uk_dbmember foaf:name name ;
      dbpprop:birthDate ?birthdate .
    BIND ( (CONCAT(?lastname, " ", ?firstname)) AS ?nameConcat ) ↩
        ↪ }
```

Figure 10: Excerpt of the integration module for data.gov.uk

6 Using ALLOT to query heterogeneous datasets

To test the feasibility of using ALLOT as an ontology for querying multiple databases, we developed three alignment modules, available at http://akn.web.cs.unibo.it/allot-ext/. These modules allow the use of ALLOT to query the datasets released by the Italian Chamber of Deputies, the Brazilian Parliament and the UK Parliament. These alignment modules are based on the SPARQL CONSTRUCT alignment technique, together with the addition of external data sources for the missing data.

Writing queries that deal with legal data sourced from multiple datasets is easy, thanks to the minimality and simplicity of ALLOT. An example of query is shown in figure 11. That query returns the list of members of Parliament under 40 years of age. The same query can return entities from both the UK parliamentary data (thanks to the module excerpt shown in figure 10) and from the IT parliamentary data (thanks to the module shown in the figure 7).

```
SELECT DISTINCT ?person ?name ?birthDate
WHERE {
  ?person a allot:Person;
  allot:name ?name ;
  bio:event [
    a bio:Birth ;
    bio:date ?birthDate ] ;
  allot:role [
    allot:roleType allot:member_of_parliament ;
  ]
  FILTER ( ?birthDate >= "1973-01-01"^^xsd:date )
}
```

Figure 11: ALLOT query for members of parliaments under-40

Figures 12 and 13 show how to get the same result by querying each dataset separately, a solution that is clearly verbose and error-prone.

```
# prefixes omitted

SELECT ?person ?name
       (COALESCE(?birthdate_it, ?birthdate_uk, ?birthdate_br))
WHERE {
  {
    ocdlegisl:repubblica_16 rdfs:label ?leg_label .
    ?person a ocd:deputato;
      ocd:rif_leg ocdlegisl:repubblica_16 ;
      foaf:firstName  ?it_firstName ;
      foaf:surname ?it_surname .
    ?it_person a foaf:Person ;
      foaf:firstName  ?it_firstName ;
      foaf:surname ?it_surname ;
        bio:event ?it_bioev .
      ?it_bioev bio:Birth ?it_birth .
      ?it_birth bio:date ?it_birthdate .
    BIND (SUBSTR(?it_birthdate,1,4) AS ?it_year) .
    BIND (SUBSTR(?it_birthdate,5,2) AS ?it_month) .
    BIND (SUBSTR(?it_birthdate,7,2) AS ?it_day) .
    BIND ( xsd:date(CONCAT(?it_year, "-", ?it_month, "-", ?it_day↩
       ↪ )) AS ?birthdate_it ) .
    FILTER ( ?birthdate_it <= "1973-01-01"^^xsd:date )
    BIND (CONCAT (?it_firstName, " ",  ?it_surname) AS ?name)
  } UNION {
    ?person foaf:name ?name ; parl:partyMemberOf ?party .
    ?uk_dbmember foaf:name ?name ;
      dbpprop:birthDate ?birthdate_uk .
    FILTER ( ?birthdate_uk >= "1973-01-01"^^xsd:date )
  }
...
```

Figure 12: Separate queries for members of parliaments under-40 (first part)

```
...
  UNION {
    ?person a dbpont:Person;
      foaf:name ?name;
      foaf:birthday ?birthday .
    BIND (SUBSTR(?birthday,7,4) AS ?br_year) .
    BIND (SUBSTR(?birthday,4,2) AS ?br_month) .
    BIND (SUBSTR(?birthday,1,2) AS ?br_day) .
    BIND (xsd:date(CONCAT(?br_year, "-", ?br_month, "-", ?br_day)↩
        ↪ ) AS ?birthdate_br ) .
    FILTER ( ?birthdate_br >= "1973-01-01"^^xsd:date )
  }
}
```

Figure 13: Separate queries for members of parliaments under-40 (second part)

Another example is the query in figure 14, that returns the list of members of Parliament that hold a seat on a given date. Compared to the query shown in figure 11, this query is slightly more complicated. The main source of complexity, however, comes from the use of the sophisticated LKIF time module, used by ALLOT to describe time intervals. The representation of moments and periods of in time in LKIF is very precise but it also requires the use of various intermediary entities and concepts that makes it hard to write very concise queries. In this case, the query asks for all the persons whose role as member of parliament is an LKIF interval that is between two moments that delimit a certain `Pair_Of_Periods`.

```
PREFIX  allot:    <http://akn.web.cs.unibo.it/allot/>
PREFIX  lkiftime:  <http://www.estrellaproject.org/lkif-core/time↩
    ↪ .owl#>
PREFIX  lkifmod:    <http://www.estrellaproject.org/lkif-core/↩
    ↪ time-modification.owl#>
PREFIX  lkifmereo:  <http://www.estrellaproject.org/lkif-core/↩
    ↪ mereology.owl#>

SELECT DISTINCT ?person ?name
WHERE {
  ?person a allot:Person;
      allot:name ?name ;
      allot:role [
        allot:roleType allot:member_of_parliament ;
        allot:interval [
          a lkiftime:Interval ;
          lkiftime:between ?pp ] ;
      ] .

  ?pp a lkiftime:Pair_Of_Periods .
  _:m1 a lkiftime:Moment ; lkifmereo:component ?pp ;
    lkifmod:date ?mandate_start .
  _:m2 a lkiftime:Moment ; lkifmereo:component ?pp ;
    lkifmod:date ?mandate_end .
  FILTER ( ?mandate_start <= "2011-03-16T00:00:00+01:00"^^xsd:↩
      ↪ dateTime &&
     ?mandate_end >= "2011-03-16T00:00:00+01:00"^^xsd:dateTime )
}
```

Figure 14: ALLOT query for members of parliaments in charge on 2011-03-16.

One could argue that the simplicity brought by the use of ALLOT can be attained using any other ontology as a pivot ontology, as long as that ontology is comprehensive enough. However, these ontologies have often goals different from that of ALLOT. Take for examples the widespread LKIF ontology. Although they share many concepts, LKIF and ALLOT show different abilities and deficiencies. For example, a deficiency of LKIF is the fact that it does not define a way to represent information about people, not even basic data such as birth dates or names; instead, LKIF delegates this problem to external ontologies, without giving guidance on which other ontologies should be used. ALLOT, instead, provides simple yet complete methods to record such information. On the other hand, LKIF shines when it comes to representing very detailed information about more abstract concepts. For instance, LKIF is based on a rich theory of how time works and allows events to be described as happening after or before other periods of time, a useful feature that can map with precision the legal content of many laws. ALLOT uses a much simpler model of time and lacks the ability to describe these finer points.

To summarize, ALLOT should be seen as a lightweight and easy ontology that can be used to write simple queries that focus on the most common entities and properties can be found in legal datasets. This is different from other ontologies such as Metalex or LKIF that can be used to define very precise queries over a set of advanced legal concepts at the expenses of simplicity of these queries.

7 Conclusions

The main obstacle to a more effective integration of legal KBs still remains the heterogeneity of these resources. It is not only a matter of formats but also a problem of intrinsic incompatibility: in several cases radically different views of the world are employed, making it very hard to link entities in a dataset with entities in other datasets.

In this paper we presented ALLOT, a lightweight ontology designed on top of Akoma Ntoso guidelines and shown how it can be used to define simple queries to interrogate multiple datasets based on different ontologies. The key idea of ALLOT, and the overall “non-ontology” approach, is to heavily simplify the model and to define a few disjoint and very abstract classes. Such a simple ontology has been successfully used as pivot to align and query heterogeneous datasets. In particular, we experimented ALLOT to bridge the ontologies modeling the datasets of the Italian Chamber of Deputies, the Brazilian Parliament and the UK Parliament.

On the other hand, we experienced a lot of difficulties to align some ontologies, because of their intrinsic model and differences with ALLOT. We eventually managed to complete the task but we had to write complex CONSTRUCT queries to capture subtle differences between classes. This is a quite common scenario that hinders the integration of KBs in practice.

Our experience with ALLOT also highlighted some weaknesses of current technologies that have great impact in this context. First of all, the fact that there is no standardized way to record and transmit information about the provenance of legal statements. In fact, most of the assertions that can be extracted from legal documents are not universal truths. These assertions are assertions made by a particular actor (say, the author of the document) in a precise context (a particular version of the document) at a precise time (the date and time of the document). Semantic technologies like RDF or OWL lack the features needed to express in a simple way the kind of statements these details require. In general, to express such information, ad-hoc descriptions have to be devised and used, possibly following some of the already existing design patterns

[25]. The problem with these workarounds is that they are harder to use compared to their simpler counterparts that cannot preserve all the information contained in the original legal text. However, the understandable designers' desire to keep the ontologies "simple" for their users, make the designers stick to the simplest features of the current technologies. In the case of ALLOT, the effects of the lack of contextualization capabilities become evident as we we included constructs to express contexts and time-bound information as suggested by Akoma Ntoso guidelines. To allow them we faced a remarkable drawback, that also applies to all ontologies that follow similar approaches: the explosion of intermediate entities and relations. Yet, design patterns define guidelines and do not require designers to invent new conceptual schemas but the actual creation of classes, relations and tuples is still a tedious and error-prone process.

Our hope is that future technologies will have more expressive features to express in simple ways this kind of information and all the accurate data that can be extracted from legal texts. This, in turn will simplify how KBs are created, linked and queried.

References

[1] CGOV: Central government ontology, an ontology of UK central government. http://reference.data.gov.uk/def/central-government.

[2] Ligado nos polÃticos. http://ligadonospoliticos.com.br.

[3] OASIS LegalDocumentML technical committee charter. https://www.oasis-open.org/committees/legaldocml/charter.php.

[4] Ontologia Camera dei Deputati, an ontology for the Italian Chamber of Deputies. http://data.camera.it/data/en/.

[5] OpenRefine: A free, open source, power tool for working with messy data. http://openrefine.org.

[6] PARL: Parliament ontology, an ontology of UK parliament. http://reference.data.gov.uk/def/parliament/.

[7] TheyWorkForYou: Hansard and official reports for the uk parliament, scottish parliament, and northern ireland assembly - done right. http://theyworkforyou.com/.

[8] Joost Breuker Abdullatif, Joost Breuker, Abdullatif Elhag, Emil Petkov, and Radboud Winkels. Ontologies for legal information serving and knowledge management. In *Proceedings of JURIX 2002*, pages 73–82. IOS Press, 2002.

[9] Gioele Barabucci, Luca Cervone, Monica Palmirani, Silvio Peroni, and Fabio Vitali. Multi-layer markup and ontological structures in Akoma Ntoso. In *AI Approaches to the Complexity of Legal Systems: Proceedings of AICOL 2009*, volume 6237 of *LNCS*, pages 133–149. Springer-Verlag, 2010.

[10] Sean Bechhofer and Alistair Miles. SKOS Simple Knowledge Organization System Reference. Recommendation, W3C, August 2009. http://www.w3.org/TR/2009/REC-skos-reference-20090818/. Latest version available at http://www.w3.org/TR/skos-reference.

[11] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *J. Web Sem.*, 7(3):154–165, 2009.

[12] Alexander Boer, R. J. Hoekstra, E. De Maat, Fabio Vitali, Monica Palmirani, and B. Ratai. Metalex: Open XML interchange format for legal and legislative resources. CWA 15710:2007 Metalex 1, 2010.

[13] Alexander Boer, W. Radboud, and Fabio Vitali. Metalex XML and the Legal Knowledge Interchange Format. In *Computable Models of the Law*, volume 4884 of *LNCS*, pages 21–41. 2008.

[14] Dan Brickley and Libby Miller. FOAF vocabulary specification 0.98. Technical report, August 2010. http://xmlns.com/foaf/spec/20100809.html.

[15] Paolo Ciccarese and Silvio Peroni. Essential FRBR in OWL2 DL, 2011. http://purl.org/spar/frbr.

[16] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic Web*, 2(1):3–10, 2011.

[17] Ian Davis and David Galbraith. BIO: A vocabulary for biographical information. http://vocab.org/bio/0.1/.

[18] Gerard de Melo and Gerhard Weikum. Language as a foundation of the semantic web. In Christian Bizer and Anupam Joshi, editors, *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany, October 28, 2008*, volume 401 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[19] Sylvie Despress and Sylvie Szulman. Merging of legal micro-ontologies from europen directives. *Artif. Intell. Law*, 15(2):187–200, June 2007.

[20] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[21] Aldo Gangemi. *Law, Ontologies and the Semantic Web*, chapter Introducing pattern-based design for legal ontologies, pages 53–71. IOS Press, 2009.

[22] Aldo Gangemi, Maria-Teresa Sagri, and Daniela Tiscornia. A constructive framework for legal ontologies. In *Law and the Semantic Web*, pages 97–124, 2003.

[23] Rinke Hoekstra, Joost Breuker, Marcello Di Bello, and Alexander Boer. *Law, Ontologies and the Semantic Web*, chapter LKIF Core: Principled Ontology Development for the Legal Domain, pages 21–52. IOS Press, 2009.

[24] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61, 2013.

[25] C. Huitfeldt, S. Peroni, and F. Vitali. Documents as timed abstract objects. *to appear in the Proceedings of Balisage Series on Markup Technologies, The Markup Conference 2012*, 2012.

[26] Daniel Lathrop and Laurel Ruma. *Open Government: Collaboration, Transparency, and Participation in Practice*. O'Reilly Media, Inc., 2010.

[27] Olivia M.A. Madison. The IFLA Functional Requirements for Bibliographic Records: International standards for universal bibliographic control. *Library Resources & Technical Services*, 44(3):153–159, July 2000.

[28] Claudio Masolo, Laure Vieu, Emanuele Bottazzi, Carola Catenacci, Roberta Ferrario, Aldo Gangemi, and Nicola Guarino. Social roles and their descriptions. pages 267–277. AAAI Press, 2004.

[29] Monica Palmirani and Fabio Vitali. Akoma Ntoso users. `http://www.akomantoso.org/akoma-ntoso-in-detail/users`.

[30] A. Powell, M. Nilsson, A. Naeve, and P. Johnston. Dublin Core metadata initiative: Abstract model, 2005. `http://dublincore.org/documents/abstract-model`.

[31] Maria-Teresa Sagri and Daniela Tiscornia. Metadata for content description in legal information. In *Proceedings of DEXA '03*, pages 745–752. IEEE Computer Society, 2003.

[32] Ioannis Savvas and Nick Bassiliades. A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration. *Expert Syst. Appl.*, 36(3):4467–4478, April 2009.

[33] David Shotton and Silvio Peroni. PRO, the publishing roles ontology, 2012. `http://purl.org/spar/pro`.