



## Answering Complex Queries on Legal Networks : a Direct and a Structured IR Approaches

Nada Mimouni, Adeline Nazarenko, Sylvie Salotti

### ► To cite this version:

Nada Mimouni, Adeline Nazarenko, Sylvie Salotti. Answering Complex Queries on Legal Networks : a Direct and a Structured IR Approaches. Network Analysis in Law Workshop (held in connection with JURIX 2015), Dec 2015, Braga, Portugal. hal-01293032

**HAL Id: hal-01293032**

**<https://hal.science/hal-01293032>**

Submitted on 24 Mar 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Answering Complex Queries on Legal Networks : a Direct and a Structured IR Approaches

Nada MIMOUNI <sup>a,1</sup>, Adeline NAZARENKO <sup>a</sup> and Sylvie SALOTTI <sup>a</sup>

<sup>a</sup>*LIPN, Paris 13 University – Sorbonne Paris Cité & CNRS, France*

## Abstract.

This paper highlights the benefit of semantic information retrieval in legal networks. User queries get more complex when they combine constraints on semantic content and intertextual links between documents. Comparing two methods of search in legal collection networks, we present new functionalities of search and browsing. Relying on a structured representation of the collection graph, the first approach allows for approximate answers and knowledge discovery. The second one supports richer semantics and scalability but offers fewer search functionalities. We indicate how those approaches could be combined to get the best of both.

**Keywords.** Semantic information retrieval, Legal collection graphs, Exact answers, Approximate search, Knowledge discovery.

## 1. Introduction

Legal practitioners are permanently in the need to search legal collections in order to assess regulatory texts applicable to a specific case. Needs analysis with legal experts and partners of the Légitocal project [1] showed that users express their needs in the form of complex queries that address both the semantic content and intertextual links between documents. For instance, when drafting an order, municipal clerks typically have to identify the legislation to refer to as well as former orders published on the same topic, especially those that have been appealed. They would enter queries like *"Which local acts concerning rural roads have been appealed and were canceled by court decision?"*.

Legal information retrieval (IR) systems widely used by both citizens and practitioners are not able to handle such complex queries. They return documents based on the keyword they contain or are associated to as metadata and not on intertextual grounds. Answers are returned as a list of documents without taking into account the graphs to which those documents belong (esp. references to other texts). To build the context of a given document, a step of exploratory search – starting from one returned result and going through its hypertextual links – is required. In legal collections, where numerous and various types of links exist between legal sources, users get easily lost.

---

<sup>0</sup>This work has been partially funded by French Single Inter-Ministry Fund (FUI-9, 2010-2013 and is supported by the French National Research Agency (ANR-10-LABX-0083) in the context of the Labex EFL

<sup>1</sup>Corresponding Author: N. Mimouni; E-mail: nada.mimouni@lipn.univ-paris13.fr

Moreover, legal IR systems operate as logical search systems since a user needs to get all texts related to his query and not only the most relevant ones. Answers are returned when documents match exactly the sent queries. When no exact answer is found or when too few or too many answers are returned, users are not satisfied with the results and must continue to query the system until they get a relevant response.

In an earlier research, we have proposed an approach towards semantic and graph-based search in networks of legal documents [5] and demonstrated the potentialities that a graph-based semantic IR approach could offer to legal content management systems. In this approach, the intertextual information is taken into account at the querying level to that it is possible to answer complex queries like *"What are the legal decisions that cite Article 1382 of the Civil Code?"*, *"I am looking for decisions that were canceled by the Court of Cassation."* or *"What are the code articles that have been confirmed and which are cited by the municipal orders dealing with rural roads?"*.

In this work, we propose to extend that logical approach with advanced search and discovery functionalities, by returning approximate answers when no exact one is found, by allowing to restrict or enlarge the answers set when too many or too few are found and by discovering new knowledge not easily detected by hypertextual navigation from the list of results. Needs analysis showed that such functionalities are increasingly expected. For instance, legal practitioners may need to understand what laws cited by a given text are talking about or what articles citing a given law are dealing with.

We implemented graph-based search using two different approaches: a structured and a direct one. The first approach creates a semantic hierarchical and relational structure on the top of the collection thus allowing for advanced search functionalities but it is limited to sub-collections<sup>2</sup>. The second approach can handle larger collections but supports only exact querying.

We present our experimental data on section 2 and our methods of graph-based IR in Section 3. Exact search and browsing functionalities are discussed respectively in Sections 4 and 5. Section 6 introduces related works addressing the problem of modeling and querying graphs of legal collections.

## 2. Experimental Data

We test and evaluate our approaches on three collections of documents:

- The *Noise corpus* was used as an illustrative example in Légilocal. It is composed of 10 documents annotated with 10 attributes related to noise and a single reference relation. Documents are of different types: local acts (orders) making reference to legislative texts (decree, law, code, ordinance).
- The *Légilocal corpus* is a richer collection extracted from the Légilocal base for demonstration purposes: it is composed of 25 documents and 30 legal articles, various types of documents are represented (local decisions, legislative texts, judgements and editorial documents) – sometimes with different versions of the same document –, as well as various types of links (application, composition, decision either confirmation or cancelation, modification) [5].

---

<sup>2</sup>The target scale is a few hundred documents, but we must further optimize the code for it (solutions exist, it is an implementation issue).

- The *ILO corpus* is the largest collection with almost 400 documents collected from the International Labour Office (ILO)<sup>3</sup>. Documents are conventions and recommendations, linked by an implementation relation.

These collections have served us as a primary base to test the proposed approaches. The documents are analyzed to identify their types and extract their structure, they are annotated with semantic descriptors and the intertextual citations are themselves identified and semantically typed. Semantic descriptors annotating documents correspond to concepts extracted from semantic resources dealing with the texts domains. Real scale collections are to be created and documents analyzed by partners of the project.

Needs analysis realized with collaboration of Légilocal legal experts shows that practitioners' spontaneous queries naturally combine content and intertextual criteria. We built a list of queries which are formally different from each other but which covers a wide range of practitioners' queries. We created a set of queries on both ILO and Légilocal (LEGI) corpus along with their relevant answers to test and evaluate our search approaches, the quality of a method being related to the variety of query types that it supports.

- IL01 : "Which convention implements Recommendation 113 on collective bargaining?"
- IL02 : "Which convention implements the recommendation talking about seafarers occupational accidents?"
- IL03 : "What are the recommendations implemented by conventions dealing with air pollution?"
- LEGI1 : "What are the legal decisions that cite Article 1382 of the Civil Code?"
- LEGI2 : "What are the articles of the Environment Code that deal with motor vehicles and rural roads?"
- LEGI3 : "I am looking for the decision that is the subject of the judgment A of the Court of Cassation."
- LEGI4 : "I am looking for decisions that were canceled by the Court of Cassation."
- LEGI5 : "I wonder if this judgment of the Court of Appeal was itself the subject of an appeal."
- LEGI6 : "I would like to see municipal orders regarding rural roads that have been appealed and were canceled by decision of jurisprudence."
- LEGI7 : "What are the code articles which are cited by the municipal orders dealing with rural roads that have been confirmed?"
- LEGI8 : "What are the decisions prior to decision D?"
- LEGI9 : "I wonder if the texts referred by municipal orders dealing with rural roads are also cited by those concerning the motor vehicles."
- LEGI10 : "What are municipal orders that have been the subject of two appeals?"
- LEGI11 : "I wonder if the texts referred by the municipal order 97-17 of Champigné have been modified, and if so, what are the new versions of these texts and the source texts of the amendment."

---

<sup>3</sup>[www.ilo.org](http://www.ilo.org)

### 3. Modeling Legal Collections as graphs

The document collection is modeled as a graph of documents, which can be exploited directly using semantic technologies (direct approach) or which can be further structured into a hierarchical and relational conceptual structure, that itself supports search and browsing (structured approach).

The direct approach models the collection as an RDF graph where the objects (nodes) are document and descriptor identifiers and where properties (edges) are intertextual links (between documents) or annotations (between documents and descriptors). An OWL document ontology controls the semantics of the RDF graph. It provides for fine-grained descriptions of legal collections, taking any type of features into account: semantic metadata, document structure (sections, paragraphs, etc.), document types (law, decree, etc.) and intertextual relationships. User queries are expressed in SPARQL and are directly matched against the RDF graph.

Relying on FCA/RCA, the structured approach [6] models the collection as a set of interlinked hierarchical structures which organizes documents into classes on the basis of their semantic content and their interrelationships. Compared with the direct approach, the initial modeling step is simpler (the collection is modeled as a set of objects/attributes structures) but there is an additional structuring phase. Similarly querying the structure is straightforward, as any answer is already pre-computed as a class of documents, but the structure can be also exploited for browsing.

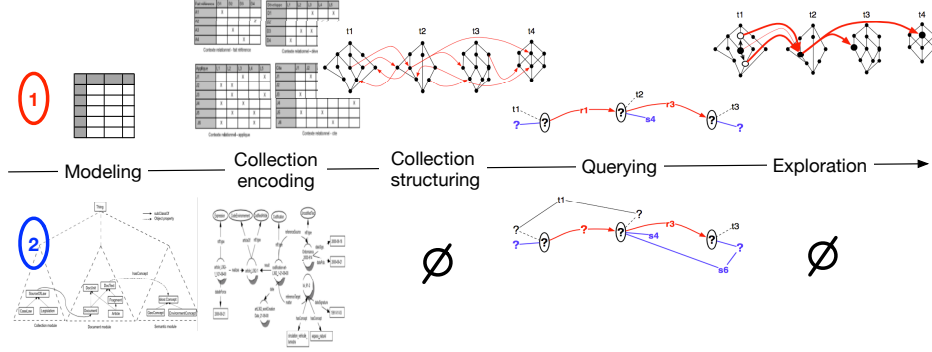
Figure 1 compares these approaches. We can notice that modeling and encoding the collection are more sophisticated with the direct approach (ontological model and RDF triples) than with the structured approach (binary tables). This allows the direct approach to express more complex query graphs. Conversely, the structured approach has the advantage of structuring the collection.

In information retrieval, structuring a document collection as a lattice pre-computes the answers to all queries that are satisfiable on that collection. In addition, browsing the structure provides approximate answers as it allows for generalizing or restricting queries. In the direct approach, answers are calculated on the fly when the SPARQL query is sent to the system. This approach is more flexible (when the model of the collection or the collection itself evolves), is not limited by the collection size and allows to express richer queries. On the contrary, the structured approach performs well on small collections or local perspectives over large collections and allows browsing and visualizing, but it cannot answer complex queries like LEGI11. We have developed a prototype to visualize the answer graphs from the structured approach displaying the objects of the answer along with all their attributes even if they are not mentioned in the initial query.

The choice to use one or the other approach depends tightly on the application requirements (user interfaces, number of documents, granularity of document description or the evolution of the collection).

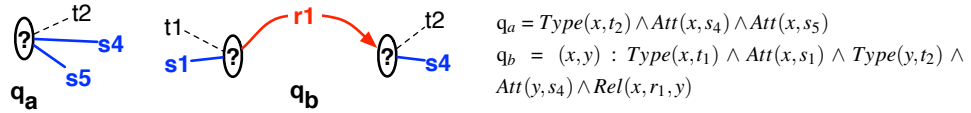
### 4. Exact Search in Collection Graphs

As presented in [5], document collections and queries are formalized as graphs. The proposed approaches enable answering different types of elementary and complex queries. We tested 14 types having or not a focus and with or without constraints. For instance,



**Figure 1.** Structured vs. direct approaches. The first approach (top) relies on FCA/RCA to build a semantic structure over the graph of documents. Queries are answered based on that structure. The second approach (bottom) exploits semantic technologies (OWL/RDF/SPARQL) to model the collection as a graph. Query answers are directly extracted from that graph. Compared with the structured approach, the direct one supports richer semantic graphs but offers no exploration facility.

Figure 2 shows two examples of query graphs :  $q_a$  corresponds to an elementary query and  $q_b$  to a relational one. The query  $q_a$  looks for documents of type  $t_2$  annotated with semantic attributes  $s_4$  and  $s_5$ . The query  $q_b$  looks for documents of type  $t_1$  annotated with  $s_1$  having the link  $r_1$  with documents of type  $t_2$  annotated with  $s_4$ .



**Figure 2.** Examples of query graphs:  $q_a$  represents an elementary query and  $q_b$  a relational query, which is represented with a focus in the formula.

On exact search and for elementary as well as for relational queries, both approaches perform well. All the selected queries are properly answered, with a list of documents or document graphs that exactly match the query or no answer if the query is not satisfiable. We show below some examples of queries, testing the direct and the structured approaches on our three test corpus : Noise, ILO and Légilocal corpus.

#### 4.1. Structured Approach

Once structured as a family of relational lattices, the collection can be exploited to answer elementary and relational queries. We tested this approach on both Noise and ILO corpus.

Let's consider an example of a relational query on the Noise corpus : "Which orders talk about sound level (*nvs*) and make reference (*rf*) to decrees about noisy activities (*ab*)?". Figure 3 shows the structure built for the collection. The query is modeled as an attributed graph, which nodes are virtual objects (variables) created for each unidentified document ( $Q_o$  for orders and  $Q_d$  for decrees), which attributes are semantic descriptors (*nvs* for node  $Q_o$  and *ab* for  $Q_d$ ) and the link is the intertextual relation (*rf*). An iterative search process calculates relevant answers [6], it instantiates  $Q_o$  and  $Q_d$  respec-

tively by documents *AS* (Strasbourg order) and *O45* (ordinance 45). It returns the following answer graph (the query has no focus, so answers are presented as graphs):  $\mathcal{G}_a = \text{Type}(\text{AS}, \text{order}) \wedge \text{Att}(\text{AS}, \text{nvs}) \wedge \text{Type}(\text{O45}, \text{decree}) \wedge \text{Att}(\text{O45}, \text{ab}) \wedge \text{Rel}(\text{AS}, \text{rf}, \text{O45})$ .

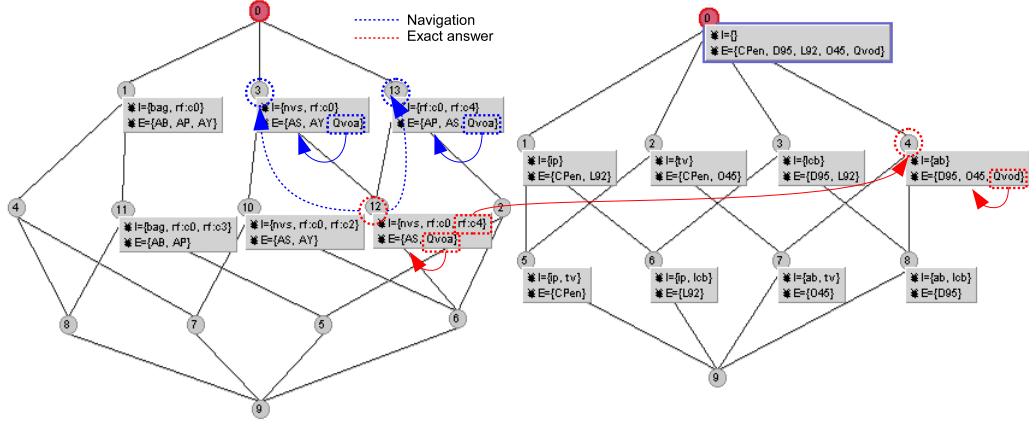


Figure 3. Exact and approximate search in a relational lattice family.

We tested the approach on the ILO corpus: all the collected queries returned the expected relevant answers. We detail hereafter two examples: IL01 and IL02.

**IL01** : this query contains an identified document *R113* (recommandation 113), we create the variable *QueryConv* corresponding to the searched one. We locate *R113*, check that it is actually associated with the attribute *collective bargaining* and search for the related convention. The query graph is:  $\mathcal{G}_q = \text{Type}(\text{QueryConv}, \text{convention}) \wedge \text{Type}(\text{R113}, \text{recommandation}) \wedge \text{Att}(\text{R113}, \text{collective\_bargaining}) \wedge \text{Rel}(\text{QueryConv}, \text{impl}, \text{R113})$ . The search returns convention *C144*. The answer graph is depicted on Figure 4, with the dashed line surrounding the exact answer.

**IL02** : the query graph contains two variables, two attributes and one relation :  $\mathcal{G}_q = \text{Type}(\text{QueryConv}, \text{convention}) \wedge \text{Type}(\text{QueryRec}, \text{recommandation}) \wedge \text{Att}(\text{QueryRec}, \text{occupational\_accidents}) \wedge \text{Att}(\text{QueryRec}, \text{seafarers}) \wedge \text{Rel}(\text{QueryConv}, \text{impl}, \text{QueryRec})$ . The search process instantiates variables *QueryConv* and *QueryRec* respectively with documents *C164* and *R142*. The visualisation of the answer graph is given in Figure 5. The central part of the graph (surrounded by the dashed line) corresponds to the exact answer having the attributes of the query.

#### 4.2. Direct Approach

As the direct approach allows to model the collection in a more fine-grained manner and is more naturally prepared to process large volume of data, it was more interesting to test it on the Légilocal collection. Nevertheless, we have tested it on the ILO corpus in the aim of comparing the approaches.

In the IL02 query, we search for objects of the class *Convention* having the relation *implement* with *Recommandation* objects described by semantic concepts *occupational accidents* and *seafarer*. The query is translated in the fol-

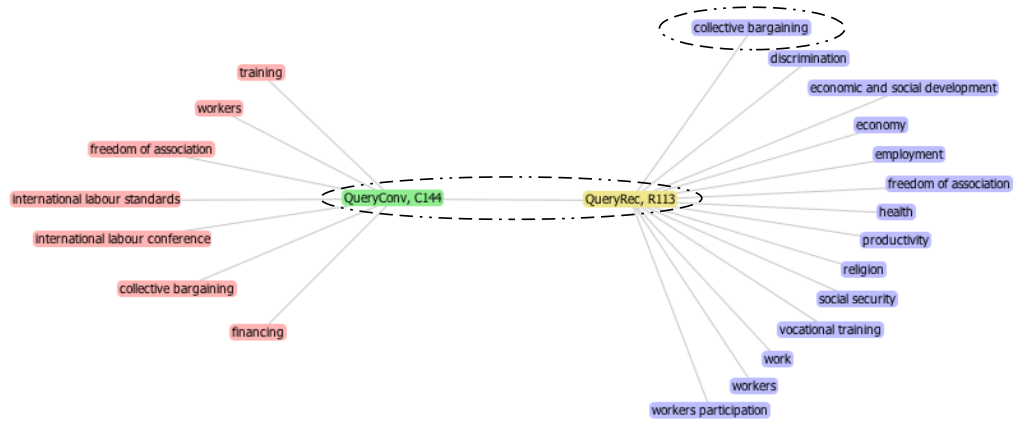


Figure 4. Answer graph for query ILO1.

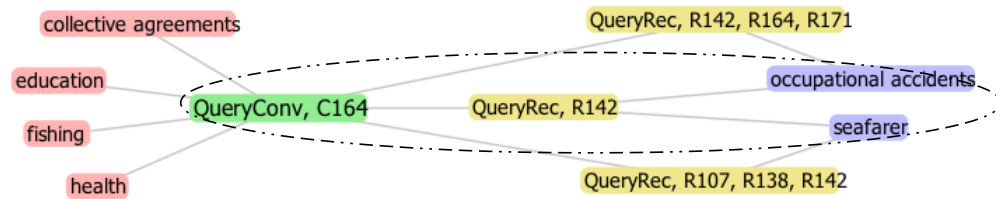


Figure 5. Exact and approximate answer graphs for query ILO2.

lowing SPARQL query (note that the relation `implement` is translated as an object property `ilo:implement` which domain is class `Convention` and range is class `Recommendation`, attributes are attached to documents via the object property `ilo:hasConcept`). It returns documents `C164` and `R142`.

```

1      SELECT ?conv ?recom
2      WHERE {
3        ?recom ilo:hasConcept ilo:occupationalaccidents , ilo:seafarer .
4        ?conv ilo:implement ?recom .
5      }

```

The expressivity of the query language enables composing more complex queries dealing with versions and fragments of documents, or with relations with more than two arguments. For instance, the query `LEGI11` is translated as follows:

```

1      SELECT ?text ?newversion ?oldversion ?source
2      WHERE {
3        ?visa rdf:type lido:VisaCitation .
4        ?visa lido:citationSource :ArreteChampagne97-17 .
5        ?visa lido:citationTarget ?text .
6        ?modification rdf:type lido:Modification .
7        ?modification lido:referenceTarget ?oldversion .
8        ?modification lido:referenceSource ?source .
9        ?modification metalex:result ?newversion .
10       ?text metalex:realizedBy ?oldversion .
11     }

```



## 5. Browsing for Approximate Search and Knowledge Discovery

### 5.1. Exploration Facilities

When a query doesn't have any exact answer, it is interesting to return an approximate answer to the user instead of an empty set of results. Also, when the user gets too few or too many answers to his query, he/she should be able to broaden or narrow his/her search by removing or adding constraints to the initial query.

This could be done by exploring the collection space and selecting documents that are more or less close to the position of the (possibly empty) exact answer variable, *i.e.* documents with less or more semantic or relational properties. The exploration process could be automated and remain transparent to the user (approximate results are calculated by the system and returned), or manual with suitable user interfaces to visualize the data space and navigate within it.

For the time being, this functionality is provided by the structured approach thanks to the structure it builds on the top of the collection, and not by the semantic approach [6]. Structuring the collection has the advantage of pre-computing the answers to all the satisfiable queries, either elementary or relational, which allows to find approximate answers to users' queries and discover new knowledge, relying on the strategy of [11] without extra calculation. Some exploration facilities are given in Figure 6.

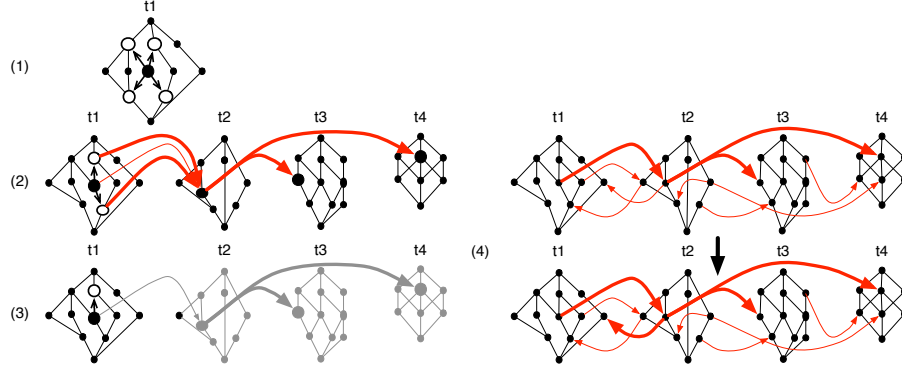
Such lattice navigation functionalities cannot be implemented on large collections, but is a track that we explore in the idea of simulating the relevant browsing strategies using SPARQL. We can actually propose some navigation scenarios, in the form of automatically generated sequences of SPARQL queries that would simulate the desired navigation strategies. For instance, the following method corresponds to a query relaxation procedure:

1. Starting from the initial query, relax one or more constraints (RDF triples) in the query graph.
2. Instantiate the relaxed SPARQL queries on the graphs of the collection.
3. Propose the obtained results to the user as an alternative or complement to the exact answer.

### 5.2. Approximate Search: examples

*Noise Corpus* Let's consider the example presented above (Section 4.1) on the noise corpus. It is possible to explore the order lattice by relaxing a semantic constraint or by dropping a relational constraint ending up with an elementary query (as described in Figure 6). For instance, on the structure of Figure 3, if we relax a semantic constraint by removing the descriptor *nvs*, we search for orders making reference to laws and decrees about noisy activities (*ab*), which are grouped in concept *c13*. A relaxed answer-graph  $\mathcal{G}_r$  is given by the document *AP* linked to the document *D95* :  $\mathcal{G}_r = \text{Type}(AP, \text{order}) \wedge \text{Type}(D95, \text{decree}) \wedge \text{Att}(D95, ab) \wedge \text{Rel}(AP, rf, D95)$ .

*ILO Corpus* On Figure 5, the right side of the graph contains, in addition to the exact answer, several approximate answers that have only one of the query attributes : seafarer for *R107*, *R138* and occupational accidents for *R164* and *R171*. They



**Figure 6.** Exploring facilities offered by the conceptual structure : query broadening and restricting for an elementary query (diagram 1) or a relational query (diagram 2), suppressing a relational attribute (diagrams 3), adding a relational attribute (diagram 4)

are obtained by upward navigating the conceptual structure which accounts for query generalization.

This can be simulated by the following sequence of relaxed SPARQL queries:

```

1      SELECT ?conv ?recom
2      WHERE {
3        ?recom ilo:hasConcept ilo:seafarer .
4        ?conv ilo:implement ?recom .
5      }

1      SELECT ?conv ?recom
2      WHERE {
3        ?recom ilo:hasConcept ilo:occupationalaccidents .
4        ?conv ilo:implement ?recom .
5      }

```

### 5.3. Knowledge Discovery: examples

On Figure 5, the left side of the graph represents the convention which implements the recommendation R142. The visualisation module presents all the attributes (semantic descriptors) of the convention C164 on the result graph. This enables the user to extend the search, looking for conventions that talk about similar topics, *i.e.* which are annotated with a subset of its attributes. This corresponds to the function of search by example [11] where the user has initially a sample, one document or a set of documents, and looks for similar ones (typically the case of a municipal clerk having to draft a local act and who starts by looking for similar ones in neighboring municipalities). We can at this stage (without navigation interface) present directly to the user the similar objects, by performing a simple search (elementary query with the attributes of C164) on the initial conceptual structure of conventions.

In addition, visualizing the conceptual structure built on the top of a given collection, allows for discovering clusters of objects that share the same properties (semantic or relational) without having to formulate queries. These clusters have served as a basis for the design and restructuring of domain ontologies [4].

## 6. Related Work

Many works have studied the intertextuality between legal sources and acknowledged that network analysis is a powerful way to model legal collections [2,7,10]. However, as in citation and social network analysis [8], focus has been put on the network level to identify the most strongly connected sub-collections or the most influential law sources. Less attention has been paid to the detailed analysis of intertextuality and the semantics of intertextual links.

In major legal access systems, intertextuality is presented as hypertextual links between texts or as a metadata that could be queried (*e.g.* Legifrance<sup>4</sup>). New approaches [9,3] try to enrich legal access systems with contextual networks based on link analysis methods. Their proposals consist mainly on the exploration and visualization of the citations network. They do not take intertextuality into account at the query level as we propose to do in exact and approximate search.

## 7. Conclusion

This paper shows that legal practitioners' complex queries could be handled through modeling and querying legal collections as semantic networks. Network nodes (documents) are associated with semantic attributes and connected to each other by various types of semantic links. Queries are themselves modeled as graphs, which are matched against the collection graph so as to return exact or approximate answers to the user. We proposed two methods – direct vs. structured methods –, which have different advantages and drawbacks. Using one model or the other depends tightly on the application: the granularity of the description required for the collection, the complexity of queries, the size of the collection, the relevance of restricted search perspective. Both proposed approaches are logical ones, they are not evaluated in terms of precision and recall. Most important criteria are the expressivity of the query language (which types of queries could be answered) and performance (collection size, processing time).

Future steps include the implementation of exploration scenarios as sequences of SPARQL queries and the design of adequate interfaces to help users create complex relational queries and analyze the returned results.

## References

- [1] F. Amardeilh, D. Bourcier, H. Cherfi, C. Dubail, A. Garnier, S. Guillemin-Lanne, N. Mimouni, A. Nazarenko, Ève Paul, S. Salotti, M. Seizou, S. Szulman, and H. Zargayouna. The légilocal project: the local law simply shared. In *Legal Knowledge and Information Systems - JURIX 2013: The Twenty-Sixth Annual Conference, December 11-13, 2013, University of Bologna, Italy*, pages 11–14, 2013.
- [2] J. H. Fowler, T. R. Johnson, J. F. Spriggs, S. Jeon, and P. J. Wahlbeck. Network analysis and the law : Measuring the legal importance of precedents at the u.s. supreme court. *Political Analysis*, 15:324–346, 2007.
- [3] D. Gultemen and T. van Engers. Graph-based linking and visualization for legislation documents (glvd). In *Network Analysis in Law Workshop (NAiL2013@ICAIL) associated with ICAIL 2013*, Rome, Italy,, June 2013.

---

<sup>4</sup><http://www.legifrance.gouv.fr/>

- [4] M. R. Hacene, P. Valtchev, and R. Nkambou. Supporting ontology design through large-scale fca-based ontology restructuring. In *Conceptual Structures for Discovering Knowledge - 19th International Conference on Conceptual Structures, ICCS 2011, Derby, UK, July 25-29*, volume 6828, pages 257–269, 2011.
- [5] N. Mimouni, A. Nazarenko, È. Paul, and S. Salotti. Towards graph-based and semantic search in legal information access systems. In *Legal Knowledge and Information Systems - JURIX 2014, Krakow, Poland*, volume 271 of *Frontiers in Artificial Intelligence and Applications*, pages 163–168. IOS Press, 2014.
- [6] N. Mimouni, A. Nazarenko, and S. Salotti. A conceptual approach for relational IR: application to legal collections. In *Formal Concept Analysis - 13th International Conference, ICFCA 2015, Nerja, Spain, June 23-26, 2015, Proceedings*, volume 9113, pages 303–318, 2015.
- [7] Romain, P. Mazzega, and D. Bourcier. A network approach to the french system of legal codes- part i: Analysis of a dense network. *Journal of Artificial Intelligence and Law*, 19:333–355, 2011.
- [8] R. Rubin. *Foundations of Library and Information Science*. Neal-Schuman Publishers, 2010.
- [9] R. Winkels, A. Boer, and I. Plantevin. Creating context networks in dutch legislation. In *Legal Knowledge and Information Systems - JURIX 2013, Italy*, volume 259 of *Frontiers in Artificial Intelligence and Applications*, pages 155–164. IOS Press, 2013.
- [10] R. Winkels and J. de Ruyter. Survival of the fittest: Network analysis of dutch supreme court cases. In *AI Approaches to the Complexity of Legal Systems. AICOL-III, Germany. Revised Selected Papers*, volume 7639, pages 106–115, 2011.
- [11] T. Wray and P. W. Eklund. Exploring the information space of cultural collections using formal concept analysis. In *Formal Concept Analysis - 9th Int. Conf., ICFCA*, pages 251–266, 2011.