Artificial Intelligence and Law (2024) 32:667–695
https://doi.org/10.1007/s10506-023-09364-9

ORIGINAL RESEARCH

A RDF-based graph to representing and searching parts of legal documents

Francisco de Oliveira[1] · Jose Maria Parente de Oliveira[2]

Accepted: 23 May 2023 / Published online: 1 July 2023

Abstract

Despite the public availability of legal documents, there is a need for finding specific information contained in them, such as paragraphs, clauses, items and so on. With such support, users could find more specific information than only finding whole legal documents. Some research efforts have been made in this area, but there is still a lot to be done to have legal information available more easily to be found. Thus, due to the large number of published legal documents and the high degree of connectivity, simple access to the document is not enough. It is necessary to recover the related legal framework for a specific need. In other words, the retrieval of the set of legal documents and their parts related to a specific subject is necessary. Therefore, in this work, we present a proposal of a RDF-based graph to represent and search parts of legal documents, as the output of a set of terms that represents the pursued legal information. Such a proposal is well-grounded on an ontological view, which makes possible to describe the general structure of a legal system and the structure of legal documents, providing this way the grounds for the implementation of the proposed RDF graph in terms of the meaning of their parts and relationships. We posed several queries to retrieve parts of legal documents related to sets of words and the results were significant.

Keywords Conceptual modeling · Legal ontologies · Legal knowledge graph · RDF graph · Semantic web · Linked data

✉ Francisco de Oliveira
francisco.oliveira@ifsp.edu.br

Jose Maria Parente de Oliveira
parente@ita.br

[1] Mathematics, IFSP - Federal Institute of Science and Technology of São Paulo, R. Primeiro de Maio, 500 - Estao, Itaquaquecetuba, SP 08571-050, Brazil

[2] Electric Engineering and Informatics, ITA - Aeronautics Institute of Technology, Pca. Mal. Eduardo Gomes, 50, 10587, So Jos dos Campos, SP 12228-900, Brazil

1 Introduction

For providing a better understanding of the context of this work, it is important to make clear the meaning of two important concepts: legal norm and legal document. A legal norm is an abstract entity that represents a pattern of “a body of rules of conduct of binding legal force and effect, prescribed, recognized, and enforced by controlling authority.",[1] while legal documents are concrete means for presenting such patterns.

Government agencies are the main providers of legal documents published on Internet portals in different formats, such as HTML and PDF. These agencies not only make a large number of legal documents available, but they also offer search means based on different criteria, such as number of the law, period of time, type of law, and keywords. So, as a result of applying such criteria, users usually get a set of legal documents and go through them to find the parts of interest. In other words, similarly to what happens when someone performs a search on the web, users have to read the returned pages to find what they are looking for.

Again, much in the same way as we do searches on the web, there is a need for finding specific parts of legal documents, such as sections, articles, items and so on. With such a support, users could find more specific information rather than having to scan the whole legal documents.

Some research efforts have been made in this area, but there is still a lot to be done to support users in finding some legal norms, specific parts of them, and all the related parts of other legal norms that would satisfy their needs. In addition,the importance of this view in a broader sense is justified by the ubiquity of the presence of legal information in the daily lives of citizens and public or private institutions.

Having the possibility of finding legal information is important once, as point out in BRASIL (1942), no nobody can claim ignorance of the legal norms. Thus, due to the large number of published legal documents and the high degree of inter-connectivity between them (Machado 2013), simple access to the whole legal documents is not enough. It is necessary to retrieve the related legal norms and their parts for a specific need. In other words, the retrieval of the set of legal norms and their parts related to a specific subject, which we call here a law framework, is necessary because it is closer to the user’s needs.

On the other hand, the tools and methods described in the literature meet this need partially, once they retrieve information at a level of granularity that loses important details. Here, we assume that the greater the complexity of the structure of the legal documents in terms of its constituent parts, the more difficult it becomes to represent granularity of parts of legal norms and their relations to other legal norms. Based on these assumptions, our research question is that legal documents are not properly represented to permit the retrieval of law frameworks for specific purposes.

In this work, we present a proposal of a RDF-based graph to represent parts of legal norms and their relations to other legal norms to permit the retrieval of law frameworks for specific needs based on a set of words.

Such a proposal is well-grounded on an ontological view. For that, we used the Basic Formal Ontology (BFO) as the starting point and extended it for our own purpose.

[1] https://legal-dictionary.thefreedictionary.com/law.

BFO is an upper-level ontology, aiming at consistently representing those upper level categories common to domain ontologies. In this work, BFO has a twofold use. First, it is used for defining the ontologies that describe the general structure of a law system and the structure of legal documents, providing not only a representation of constituent parts, but also their meaning and how they are related to each other. The second use of BFO is to provide the grounds for the implementation of the proposed RDF graph in terms of the meaning of their parts and relationships, including the meaning of how the whole and parts of legal documents are internally represented in a computer.

The implementation of the proposed graph encompasses some steps, such as downloading the HTML format of legal documents from official repositories, convert the legal documents in HTML format to RDF format, store the triples in a triple store, and once stored, the triples can be queried with SPARQL. We posed several queries to retrieve law frameworks related to sets of words and the results were significant.

The paper is organized as follows: In Sect. 2 we present the background underlying our proposal. In Sect. 3 we present the related works. In Sect. 4 we present our proposal of a RDF-based graph. In Sect. 5, we present experiments and results. Finally, in Sect. 6 we conclude the paper and provide an outlook for the continuation of this research.

2 Background

In this section, we present the main concepts in which our proposal is based on: Brazilian legal system, Basic Formal Ontology (BFO), Resource Description Framework (RDF), and SPARQL.

2.1 Brazilian legal system

In this work, we use the Brazilian legal system to contextualize the RDF graph being proposed and for the corresponding experiments carried out. Though the term legal system encompasses the set of legal norms and the ways in which they are interpreted and enforced,[2] our focus is on the set of legal documents.

Brazil is a country of continental proportions, composed of 27 states and more than five thousand municipalities. As a federal system, we have three levels of government (federal, state, and municipality), with each state and municipality having its own legislative chamber. While states and municipalities follow a unicameral system, the federal level has a bicameral system, with the National Congress divided into a Chamber of Deputies and the Federal Senate. These legislatures generate numerous legal documents. So the abundance and the entanglement of legal documents and their parts is significant.

The Brazilian legal system is based on the Brazilian Constitution, which was adopted in 1988 and provides the framework for the country's legal system. The Constitution establishes the basic principles of Brazilian legal norms, such as the separation of powers, federalism, and fundamental rights.

[2] https://www.collinsdictionary.com/dictionary/english/legal-system.

The Constitution has higher hierarchy than the other legal documents because in a certain way they derive from the constitution. The preeminence of the Constitution stems from the fact that it is the product of the original constituent power, while the other legal documents are the product of an institutional power.

The Brazilian legal system comprises the Constitution per se and other types of legal documents as described in Articles 59 and 84 of the Brazilian Constitution,[3] which are produced or approved by the National Congress, as well as others enacted by the Executive Power (Canotilho et al. 2018):

- Constitutional Amendments: These are changes to the Brazilian Constitution, which require a special legislative process and a supermajority vote in both houses of the National Congress.
- Complementary Laws: These are legal documents that are required by the Constitution to regulate specific matters, such as budgetary and fiscal policy, social security, and public debt.
- General Legal Documents: These are legal documents about general rules and regulations that are applicable throughout the country, which can cover a wide range of topics, such as civil, criminal, labor, tax, and environmental matters.
- Provisional Measures: These are temporary legal documents that can be enacted by the President of Brazil in emergency situations, such as to address a national crisis or to prevent harm to the public interest. These measures must be approved by the National Congress within a specified time frame to become permanent.
- Regulatory Decrees: These are acts from the Executive Power whose purpose is to regulate and enforce the general legal documents issued by the National Congress.

It is assumed that every legal document must either have a normative basis or a remission, i.e., there must be a basic legal document on which other legal documents rest. That is to say that any legal document not within this structure is seen as an illegitimate one.

2.2 Running example

In order to illustrate the intuition behind the research problem and the objective of this paper, we present here a running example taking into account for that a search need based on a set of words, and as a result the related laws and their parts in the Brazilian legal system, as well as how they are entangled.

The search need is expressed by the following question: “What are the legal documents and their parts related to copyrighted computer program", which leads to the following set of search words: $\{copyrighted, computer, program\}$. These words are searched through the legal documents to find all parts of legal documents where they are mentioned. Thus, assuming the existence of a mechanism for that, we get as a result a graph with the involved laws and their parts and the corresponding relationships between them, as shown in Fig. 1.

We can see that the Decree n. 2,556 is related to both Law n. 9,609/1998, called the Software Law, and Law n. 9,610/1998, which legislates on copyright. Decree n.

[3] https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm.

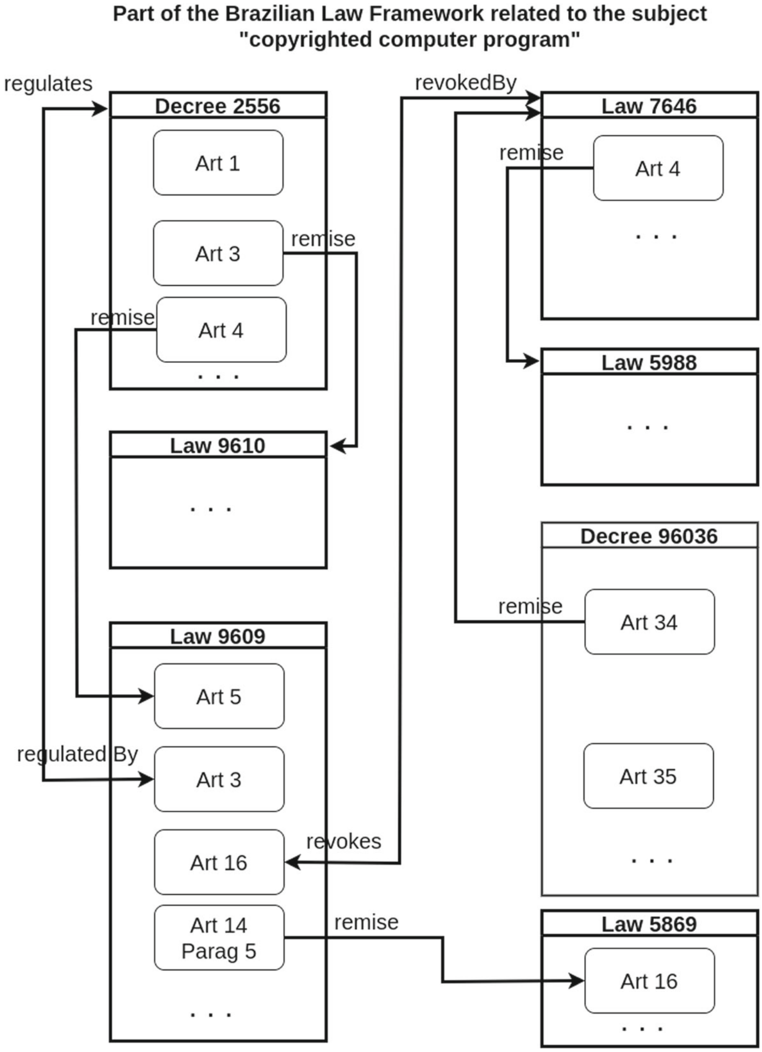

Fig. 1 Example of legal documents and their parts and the corresponding entanglement

2,556/1998 regulates article 3 of Law n. 9,609. On the other hand, Law n. 9,609/1998 revokes Law n. 7,646/1987, which deals with the intellectual property of computer programs and their commercialization. Law n. 9,609/1998 also refers to article 16 of Law n. 5,869/1973, which establishes the civil procedure code.

To summarize, Decree n. 2,556/1998 has only six articles and refers to two other legal documents and parts of them. Each of these referenced documents also refer to many others, and in this way the network of interconnections grows.

As can be seen from the example, even for a small number of legal documents, the network of interconnections can be large. In addition, the complexity and size of the interconnections gets much higher for a nationwide legal system with tens of

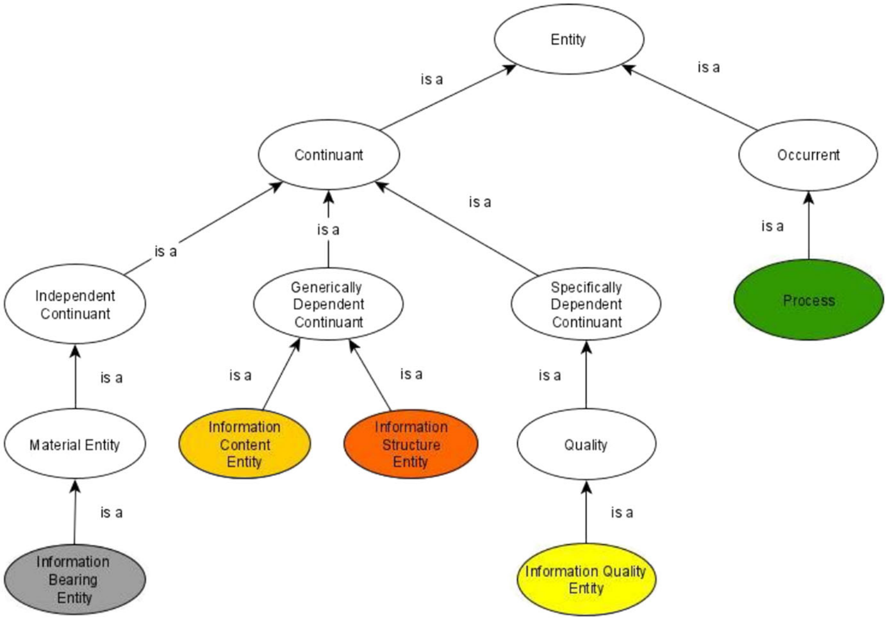
Entity
is a
is a
Continuant
Occurrent
is a
is a
is a
is a
Independent Continuant
Generically Dependent Continuant
Specifically Dependent Continuant
Process
is a
is a
is a
is a
Material Entity
Information Content Entity
Information Structure Entity
Quality
is a
is a
Information Bearing Entity
Information Quality Entity

Fig. 2 A fragment of BFO entities

thousands of legal documents that may include millions of specific parts with specific norms. As a consequence, to manually find the legal framework for some purpose becomes a hard work. This paper aims at providing an automatic way to facilitate finding such a legal framework.

2.3 Basic formal ontology

The Basic Formal Ontology (BFO) is grounded in the Aristotelian tradition. It is an upper-level ontology, aiming at consistently representing those upper level categories common to domain ontologies.

In addition to the possibility for representing many different domain entities, the main reason for using BFO in this work is due to the well-grounded capabilities for representing information content and structure in a coordinated way with other types of entities.

BFO is grounded in two broad categories of entities: continuant and occurrent. Figure 2 depicts the entities of BFO here considered. The colors used in the figure have the purpose of making easier to identify the entity types represented in BFO, specially in the figures presented in Sect. 4. In what follows, we describe such elements taking (Arp et al. 2015) as the main reference.

Continuant entities are those entities that continue or persist through time, preserving their identity through changes, and have no temporal parts.

An independent continuant is a Continuant entity that is the bearer of qualities. If a continuant entity *a* is the bearer of quality *b*, then we also say that *b* inheres in *a*.

Independent continuants are such that their identity and existence can be maintained through gain and loss of parts, and also through changes in their qualities. The entity type quality is defined on the next pages.

Material entities are Independent Continuant Entities that have some portion of matter as part. They are spatially extended in three dimensions and continue to exist through some time interval.

Information bearing entities (IBE). An IBE is a material entity that has been created to serve as a bearer of information. IBEs are either self-sufficient material wholes, or proper material parts of such wholes. Examples as self-sufficient material wholes are a hard drive, a paper printout (e.g., a report); examples as proper material parts of wholes are a specific sector on a hard drive, a single page of a paper printout (Smith et al. 2013) of legal document in PDF format.

Generically dependent continuants are Continuant entities that depend on one or more independent continuants that can serve as their bearer. We can think of the generically dependent continuants as complex continuant patterns of the sort created by authors or designers or, in the case of DNA, through the process of evolution. Examples include a company's trademark, the pattern of a signature, the words to be displayed on a computer screen, and a pattern of a body of rules. Each such pattern exists only if it is materialized in some counterpart specifically dependent continuant, more specifically a quality, which will be defined further. Such a relation in BFO is technically called "concretized by". In the case of legal document to be displayed on a computer screen they are concretized by means of binary codes inside a computer.

Information content entities (ICEs) are Generically Dependent Continuants that provide information about some portion of reality. An ICE is thus conceived as an entity that is about something in reality and which can migrate or be transmitted (for example through copying) from one entity to another (Ceusters and Smith 2015). In other words, ICEs have something as a subject; they represent, mention or describe this something; they inform us about this something (Smith et al. 2013). Typically, an Information Bearing Entity (IBE) such as a legal document will be associated with multiple ICEs at successive levels of granularity, including separate articles within the legal document, separate sentences within these articles, and so on.

Information content entities encompass some purposes such as the following (Smith et al. 2013):

- Descriptive, which is related to descriptions as we see in scientific papers, newspaper articles, and reports.
- Prescriptive, which is related to making or giving directions, rules, or injunctions.
- Directive, which is related to specifying a plan or method for achieving something.
- Designative, which is related to assigning a form of identification to someone, such as a registry of members of an organization, a phone book, a database linking proper names of persons with their social security numbers.

IBEs are physical entities that are created or modified to serve as bearer of certain patterned arrangements of ICE. Examples of such patterned arrangements include ink or other chemicals, electromagnetic excitations (Smith et al. 2013), and computer binary code.

Information structure entity (ISE). An ISE is a structural part of an ICE; it is a sort of an ICE with removed content: for example, an empty cell in a spread-sheet; a blank Microsoft Word file, the syntactic structure of a programming language, which governs the structure of instructions in that language in the process of elaborating a program, and the predefined structure of legal documents, as is the case of Brazilian legal system. ISEs thus capture part of what is involved when we talk about the 'format' of an ICE (Smith et al. 2013).

A specifically dependent continuant is a continuant entity that depends on one or more specific independent continuants for its existence. Specifically dependent continuants are said to inhere in other independent continuants, which are called their bearers, and they cannot migrate from one bearer to another. It is important to highlight that the specific dependence is a relation which obtains between one entity and another specific entity when the first one is intrinsically such that it cannot exist unless the second one also exists. Though there are more than one kind of Specifically Dependent Continuant, the one we are interested here is Quality.

Qualities are what things are by virtue of the way they are qualified (Aristotle 2016). Qualities inhere in independent continuants, which means that for a quality to exist some other independent continuants must also exist. Examples of qualities include the processing speed and the storage capacity of a computer.

An information quality entity (IQE), also referred to as "information carrier", is a Quality that is the concretization of some Information Content Entity (ICE) (Smith et al. 2013). In other words, an IQE is a quality of an information bearing entity which exists in virtue of such patterned arrangements and which is interpretable as an ICE or ISE. Such an IQE is created when an information bearing entity is deliberately created or modified to support it (patterned to serve as its bearer) The IQE used in this work is the binary code that concretize the RDF gaph of legal documents, in which such a code inheres in a computer. This is properly explained in Sect. 4.

Occurrents are those entities that occur, happen, unfold, or develop in time, usually referred to as events, processes or happenings. Occurrents are either processes that unfold in successive phases, or they are the instantaneous boundaries of processes, such as their beginnings or ends, or even the temporal and spatiotemporal regions that such entities occupy.

A process is an occurrent entity that exists in time by occurring or happening, has temporal parts and it always depends on some material entity. The dependence here is analogous to that between a specifically dependent continuant and its independent continuant bearers. Examples of processes include the life of a given organism, the process of cell division, the elaboration of an algorithm, the transformation of legal documents from a format to another, as from HTML into PDF.

2.4 RDF

The resource description framework (RDF) is a W3C standard for describing Web resources, such as the title, author, modification date, content, and copyright information of a Web page (Gandon et al. 2011). In other words, RDF was designed to

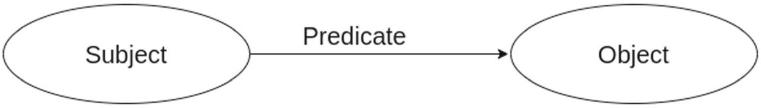

Fig. 3 RDF data model structure

provide a common way to describe information so that it can be read and processed by computer applications.

Despite not being designed to be displayed on the web as is HTML, RDF relies heavily on the infrastructure of the Web, using many of its familiar and proven features, while extending them to provide a foundation for a distributed network of data. RDF reuses the Web approach to identify resources (URI) and to allow one to explicitly represent any relationship between two resources in the form of statements. Such statements can come from any source on the Web and be merged with other statements supporting worldwide data integration (Gandon et al. 2011).

RDF is a graph data model in the form of triples. Each triple is composed of a subject, a predicate and an object. This can be illustrated graphically as in Fig. 3.

Subjects are known as the resources which we refer to, and objects can be resources as well or literals or even blank nodes. The predicate indicates either the relationship between resources or an resource's attribute, expressed as literals. Thus shared resources across subjects and objects are one of the fundamental principles that allow for constructing RDF graphs. For this reason, IRIs (Internationalized Resource Identifier) are used as the subjects and objects of triples, and as unique identifiers for predicates as well. Aiming at being machine-processable, the standardized serialization of RDF, published by W3C in the RDF/XML Syntax Specification in 2004, is the RDF/XML syntax. [4] Such a standard makes it easier to exchange documents.

To illustrate what RDF is and what it represents we can return to the running example, Fig. 1. We can see it as a RDF graph. One can refer to Fig. 4, which is its representation by means of RDF triples. The IRIs have been omitted in the figure to make it clearer to read.

We can extract a part of this figure. For example, Decree 2566 regulates Article 3 of Law 9609. Its RDF graph representation is in Fig. 5, while the corresponding RDF syntax representation can be read in Listing 1.

Listing 1 RDF Syntax for the 'Decree 2566 regulates Article 3 of Law 9609'

```
@prefix rdfs: <\url{http://www.w3.org/2000/01/rdf-schema#}> .
...
@prefix lex:  <\url{http://localhost:8080/fuseki/}> .
@prefix :     <\url{http://localhost:8080/fuseki/}> .

lex:D2556     lex:is_a          lex:legal_document .
...
lex:D2556     lex:regulates     lex:L9609/Article3 .
...
```

[4] www.w3.org/TR/rdf-syntax-grammar/.

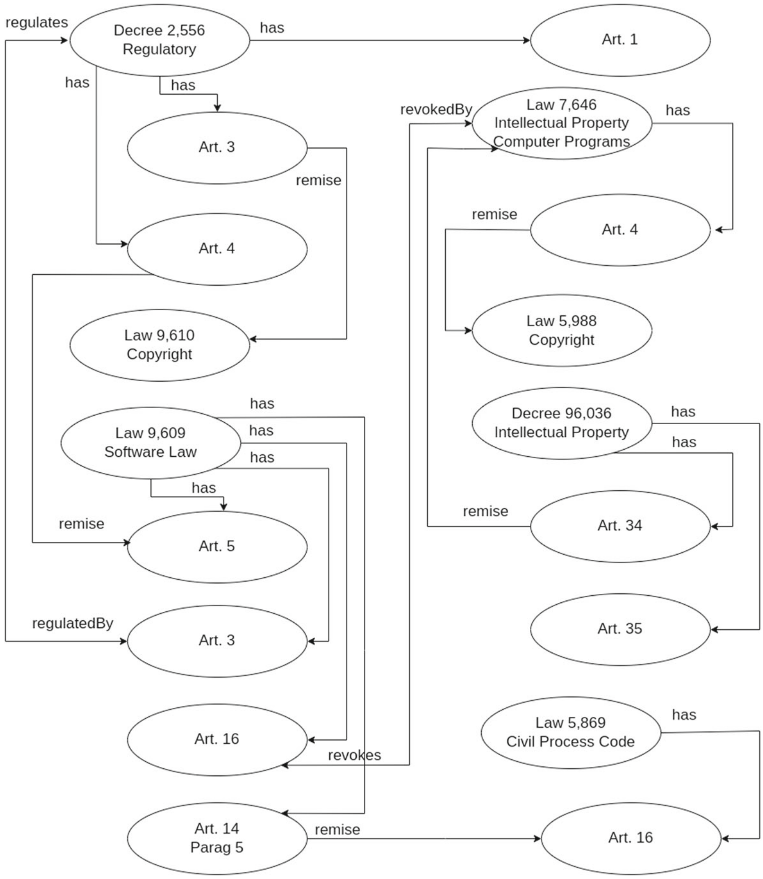

Fig. 4 Graph representation of running example

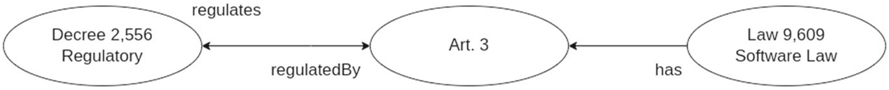

Fig. 5 Graph representation for ’Decree 2566 regulates Article 3 of Law 9609’

Remark: In the Listing 1 several rows of the real generated RDF, indicate by... (dot dot dot symbol), have been omitted for clarity and simplification. We’ve used the Notation 3 (N3),[5] a Turtle like RDF syntax.

As can be seen in the graph of Fig. 5, the relationship between Decree 2556 and Article 3 of Law 9609 is in both directions, that is, ’regulates’ has an inverse, ’regulatedBy’, and vice-versa. If the decree regulates the article then the article is regulated by the decree. The arrows indicate this feature. Therefore, it is to be expected that there is a declaration of the inverse of ’regulates’, which is ’regulatedBy’. The Listing 3 shows a RDF fragment representation of this.

[5] https://www.w3.org/TeamSubmission/n3/.

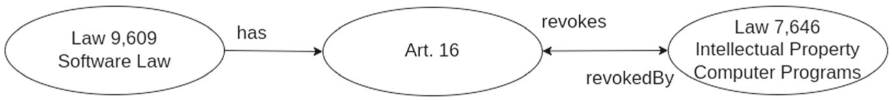

Fig. 6 Graph representation for 'Article 16 of Law 9609 revokes Law 7646'

Listing 2 RDF Syntax for the 'Article 3 of Law 9609 is regulated by Decree 2566'

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
...
@prefix lex:  <http://localhost:8080/fuseki/> .
@prefix :     <http://localhost:8080/fuseki/> .

lex:L9609           lex:is_a          lex:legal_document .
...
lex:L9609/Article3  lex:is_a          lex:article .
lex:L9609           lex:has           lex:L9609/Article3 .
...
lex:L9609/Article3  lex:regulatedBy   lex:D2556 .
...
```

Remark: Again, in the Listing 3 several rows of the real generated RDF, indicate by... (dot dot dot symbol), have been omitted for clarity and simplification. We've used the Notation 3 (N3), a Turtle like RDF syntax.

As a last example of graphical and syntactic representation through RDF, we can take another snippet from our running example. Look at Fig. 6.

The Fig. 6 shows the relationship between Article 16 of Law 9609 and Law 7646. The Article 16 of Law 9609 revokes Law 7646. We can see the representation of this through the RDF fragment according to Listing 3. This relationship is bidirectional too, that is, 'revokes' has an inverse, 'revokedBy', and vice-versa. If the article revokes the law then the law is revoked by the article. The arrows indicate this feature. Therefore, it is to be expected that there is a declaration of the inverse of 'revokes', which is 'revokedBy'.

Listing 3 RDF Syntax for the 'Article 16 of Law 9609 revokes Law 7646'

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
...
@prefix lex:  <http://localhost:8080/fuseki/> .
@prefix :     <http://localhost:8080/fuseki/> .

lex:L9609            lex:is_a    lex:legal_document .
...
lex:L9609/Article16  lex:is_a    lex:article .
lex:L9609            lex:has     lex:L9609/Article16 .
...
```

```
lex:L9609/Article16  lex:revokes   lex:L7646 .
...
```

Remark: Once more, in the Listing 3 several rows of the real generated RDF, indicate by... (dot dot dot symbol), have been omitted for clarity and simplification. We've used the Notation 3 (N3), a Turtle like RDF syntax. The inverse of 'revokes', that is 'revokedBy' isn't in the Law 9609 RDF File. This relationship was generated in the Law 7646 RDF file not showned here.

In short, RDF provides a simple way to represent distributed data, which is stored in a database known as a triple store. But such a representation of data is useless without some means of accessing that data. The standard way to access RDF data uses a query language called SPARQL (SPARQL stands for SPARQL Protocol And Rdf Query Language (Allemang and Hendler 2011).

For the purpose of this work, queries SparQL were built. Listings 4 and 5 detail two queries used to search for RDF triples in order to illustrate the objective of this paper.

3 Related work

By reviewing the literature, we arrived at a set of works that deal with the creation of ontologies and semantic web techniques.

An approach for transforming existing Austrian legal information into a legal knowledge graph was described in Filtz et al. (2021). With such an approach, it was possible to model intrinsic national aspects, population data and deal with the integration, through linked data, of legal information from other European Union countries. To that end, the technical specifications for identifiers on the web, supported by metadata standards, European Law Identifier (ELI) and the European Case Law Identifier (ECLI) were used. These vocabularies adhere to the RDF data format, the basis for linked data. Such a work is based on the premise that the current web search process is mainly based on keywords with the possibility to add filters to narrow the search. Similarly to our work presented here, Filtz et al. (2021) proposed to represent the links of a legal document with other legal documents. But, we go a little further because we deal not only with the interconnection between legal documents, but also with the connection between parts of the texts. Returning to Filtz et al. (2021), the authors also represent the link between information contained in legal documents such as entities mentioned in the text and external databases, for example Geonames or DBPedia. Another objective pursued by those authors is to support cross-jurisdictional research requests integrating legal data from other countries and the European Union.

MetaLex is an open standard proposal for encoding legal documents in XML as proposed by Boer et al. (2002). In that work, authors claim that there was a need for legal documents in a structured standard format. That is why the standardized and extensible Metalex format was developed. XML schemas can be used to validate the document. The initial focus of the work was the Dutch legislation. Then it was intended to work on Italian and Polish legislation. It also allows translation from XML to RDF through the use of XSL style sheets. The difference to other works lies in the independence of the language of legal documents and in the possibility of other applications besides search and presentation. The similarity to our work is the legislation representation in RDF format done. In the rest, the two works follow different lines, since here we focus on the interconnection of pieces of legislation through the use of semantic web technology.

It is worth mentioning that the present work had as a start point two thesis developed in the authors' research group, which dealt directly with the problem of ontological representation and the conversion of parts of legal documents into RDF triples.

Machado (2013) developed a doctoral thesis in which the objective was to create a Formal Conceptual Model of Positive Legal Ordering Relationships (MROJ). The result of such a research is a set of ontologies, among which the Ontology of the Legal Logical System, the Ontology of the Legal Norm Theory, the Ontology of the Legal Order, the Ontology of the External Legal System and the Ontology of the Trichotomous Division of Rigid Clauses. As some contributions of the work, Machado (2013) cites the expressiveness of the conceptual model. It also highlights the possibility of implementing the model in several ontological languages with different levels of expressiveness. Such implementations can be done according to computational requirements, interoperability and the possibility of publishing legal knowledge and its respective semantics. The External Legal Systematics Ontology inspired the development of ontologies defined here that support the conversion of parts of legal documents into RDF triples and their relationships.

In a master's thesis, de Oliveira (2017) assumed that accessing a single legal document is not enough to understand the regulation of a particular subject or theme. In other words, the understanding of the regulation related to a topic is impaired without access to the set of relevant legal documents that regulate the subject. Differently from what we have in the present work, de Oliveira (2017) calls the atomic parts of legal documents as legal provisions. That is, the heading of an article, a paragraph, a clause, an item, and so on are legal provisions. He also claims that legal documents have internal relationships between their own provisions. On the other hand, there are also external relationships with provisions of other legal texts.

de Oliveira and de Oliveira (2018) argues that the addition of internal and external relationships between the provisions of legal documents allows the recovery of the legal framework on a certain topic. This addition of relationships is accomplished by extracting the explicit links of the provisions of each legal document and converting them into a triple subject-predicate-object represented by RDF. To achieve this goal, the authors developed and implemented a method to convert legal documents to RDF triples, by extending the Ontology of External Legal Systematics elaborated by Machado (2013) and producing an oriented and labeled graph. For the purposes of queries in SPARQL language, de Oliveira and de Oliveira (2018) built a triplestore. The relevance of the data returned for the term-based searches was classified by legal experts. The metric of relevance results was the F-Measure.

Leone et al. (2019) brought a comparative inventory of the most recently published and reuse-oriented ontologies. The authors aim to give visibility to the state of the art in legal ontologies, showing and comparing existing alternatives. In the extensive comparative analysis carried out by the authors, they classified the analyzed ontologies and explained the differences between their features. The classification separated the ontologies into five domains (policies, licenses, tenders and procurements, privacy and cross-domains). For each domain, a set of ontologies was selected considering whether they were recently published or updated (second decade of the 21st century), the availability of ontology source files, and the modeling of the legal domain referring to some European or globally applicable legal framework. This work is an important contribution to the field, as it really brings into view the main ontologies of the legal domain, highlights the pros and cons, the reusability and availability of the inventoried and analyzed ontologies.

In the Lynx Project, Moreno Schneider et al. (2022) created a knowledge graph for the legal domain for the application of semantic processing as well as analysis and enrichment of documents in this domain. The main focus of the developed project is the treatment of compliance issues in multi-jurisdictional and multilingual contexts. Government institutions and private ones, in general, publish their regulations in their own national languages, which makes it challenging for decision makers to access, compare and even understand the documents corresponding to the regulations. The project created a legal knowledge graph (LKG) that is made available as a service platform (Lynx Service Platform). Access to documents is provided through a set of services based on natural language processing (NLP) and information retrieval (IR) that are combined by workflow managers (WM). In short, the platform allows multilingual searches and answers to questions within the legal domain. In this work, the authors present some use cases of Lynx that show the versatility and usefulness of the platform in several areas such as Geothermal Energy (GTE), analysis of contracts and labor legislation. The platform services share a set of rules and the legal knowledge graph was built on top of a data model making use of widely established ontologies and standards such as the European Legislation Identifier (ELI), open standards and recommendations of the W3C for open data publishing. The Lynx project has a document manager that works as a bridge between applications and external users and the project itself. The workflow manager is responsible for orchestrating the services. Lynx obviously makes use of linguistic processing resources through the integration of vocabularies that are dependent or not on the domain. As for semantic services, the platform uses named entity recognition (NER), entity linking (EL), temporal analysis of expressions (TimEx) and semantic similarity. The authors conclude that, despite the Lynx project being an important contribution in the area, the implementation and use of knowledge graphs for the legal domain is a frontier to be explored due to its current insufficient development. Comparing the work of Moreno Schneider et al. (2022) with what we are presenting here, there is a difference in objectives since the Lynx project has a broad proposal in the field of legal compliance and our graph aims to present the interconnections between parts of legal documents in the context of Brazilian federal legislation.

The work we present here is heavily based on de Oliveira (2017) and de Oliveira and de Oliveira (2018). Effectively, the present work is a substantial evolution in the sense that there is a redefinition of the original ontology for the creation of RDF triples, as well as providing a grounded ontological meaning for the proposed approach as a whole, and carrying out more and new experiments. The new ontologies are BFO compliant.

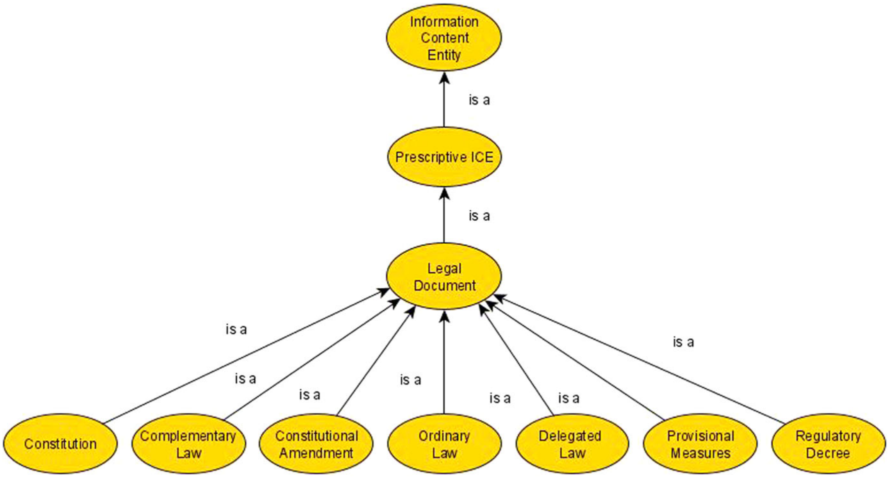

Fig. 7 Ontological conception of the Brazilian legal system

4 RDF-based graph for parts and whole of legal documents

As previously argued and particularly illustrated in Sect. 2.4, parts and the whole of legal documents can be represented by a RDF graph. Thus, this is what we present in this section, the approach for creating such a graph from a public repository of the Brazilian legal system. In such an approach, we assumed that legal documents have a basic structure. For this reason, to represent and search for specific parts of such texts, we needed to represent the general legal system we were interested in, having the basic structure of legal documents as the building block for identifying their parts.

Thus, in this section, firstly we present the ontological conception of the RDF-based graph, and then next present its implementation.

4.1 Ontological conception of the RDF-based graph

In order to provide a well-grounded ontological conception for the RDF-based graph, we started extending the leaves of BFO presented in Sect. 2.3, Fig. 2. Thus, the first aspect to represent was the part of the Brazilian Legal System of our interest here, as shown in Fig. 7, which is based on Information Content Entity.

Thus, as can be seen in Fig. 7 all types of legal documents in the Brazilian legal system have prescriptive purposes, and as said before the $Constitution$ is hierarchically higher than the other legal documents, though such a fact is not represented explicitly in the ontology.

In Sect. 2.1, we mentioned the $Constitution$ and described the legal documents $ComplementaryLaw$, $ConstitutionalAmendment$, $ProvisionalMeasures$, $RegulatoryDecrees$, and $GeneralLegalDocuments$. But now the type $General LegalDocuments$ is split into $Ordinary$ and $Delegated\ Laws$, which are described next.

OrdinaryLaws can address various topics, such as civil, criminal, commercial, labor, administrative, and many others. They establish rules and regulations that govern the rights and obligations of individuals, businesses, and the government. They can also modify or revoke existing legal documents.

Delegatedlaw refers to legislation that is created by the executive branch of the government under a delegation of legislative power from the National Congress. It is also important to say that Delegated laws in Brazil cover a wide range of areas, including administrative procedures, technical standards, licensing requirements, environmental regulations, and other specific regulations necessary for the implementation of the ordinary laws passed by the National Congress.

For the sake of clarity in Fig. 7, we do not show that *ComplementaryLaw* regulates matters in the *Constitution* and *ConstitutionalAmendment* amends the *Constitution*. In the same way, it is not represented in the figure that *OrdinaryLaw*, *Delegated Law* and *RegulatoryDecree* are interrelated through *remise*, *revokes*, *regulates* and *refersto* relationships. Such relationships are defined as follows[6]:

- *remise*: to give up something, sometimes used in quit-claim deeds; to give, grant, or release a claim to, waive, relinquish.
- *revokes*: to annul or make void by recalling or taking back; to cancel, rescind, repeal, or reverse.
- *regulates*: to control or direct according to rule, principle, or legal norm; to bring into conformity with a rule, principle, or usage; impose regulations.
- *refers to*: to direct to a source for help or information; to assign or attribute to; regard as originated by; to serve as a descriptor or have as a denotation: to relate or pertain.

The relationships remise, revokes and regulates have inverse relationships. It is worth highlighting that excepting the *Constitution* all other types of legal documents admit such relationships with other legal documents.

But the ontology of the Brazilian legal system is not enough for our purposes because we needed a representation of the structure of legal documents as well. Thus based on the Complementary Law n. 95/1998 (BRASIL 1998) and an analysis of the structure of some legal documents, we arrived at a description of such a structure. To represent it, we used Information Structure Entities because they provide a description of the structure of Information Content Entities. Figure 8 presents such a structure.

As can be seen in the figure, the structure of legal documents encompasses three parts: Primary Part, Normative Part and Final Part. The Primary Part has as parts Designative Name, Summary, and Preamble. The Designative Name is a string composed of the legal document title, a designate, number and date.

The Normative Part has different possibilities of structures. The most general one is when the Normative Part has one or more Books. As can be seen in the figure, there is the following bottom up sequence of "part of" relationships until a Book: Subsection is part of Section, which is part of Chapter, which is part of Subtitle, which is part of Title, which is part of Book. Then following the same line of reasoning, an Item is part of a Clause, which is part of an Enumerated Sentence List, which is part of a

[6] https://www.thefreedictionary.com/.

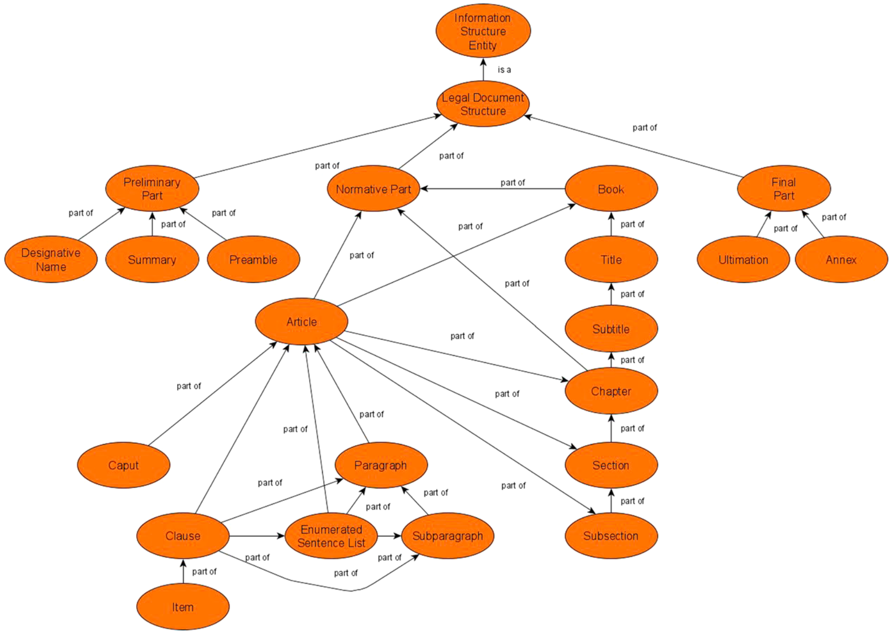

Fig. 8 General structure of legal documents

Subparagraph, which is part of a Paragraph, which is part of an Article, which is part of a Subsection. Besides this most general structure there are shortcuts in it, as is the case of an Item when it is part of a Clause, which is part of an Article, which is part of a Normative Part.

The Final Part concludes the Legal Document Structure. Such a part is composed of Ultimation and Annex.

To illustrate the notions of ICE and ISE of legal documents, we refer to Fig. 4. Thus we have the following correspondences:

- Laws 5869, 7646, 9609, and 9610 are ICEs of type Legal Document and ISE of type Legal Document Structure, which encompasses every component of the general structure of legal documents.
- Decree 2556 is an ICE of type Regulatory Decree and ISE of type Legal Document Structure, which encompasses every component of the general structure of legal documents, as well.
- Art. 3 and Art. 5, which are parts of Law 9609, and Art. 16, which is part of Law 5869, are ICEs contained in such Legal Documents and ISE of type Article.

Thus, it is important to say that with the use of Information Content Entity descriptions and Information Structure Entities depicting the general structure of legal documents, we can provide a wide representation of the Brazilian Legal System that is necessary for our purposes here.

Therefore, armed with the previous ontological conceptions of the Brazilian legal system and the general structure of legal documents, we conceived an ontological

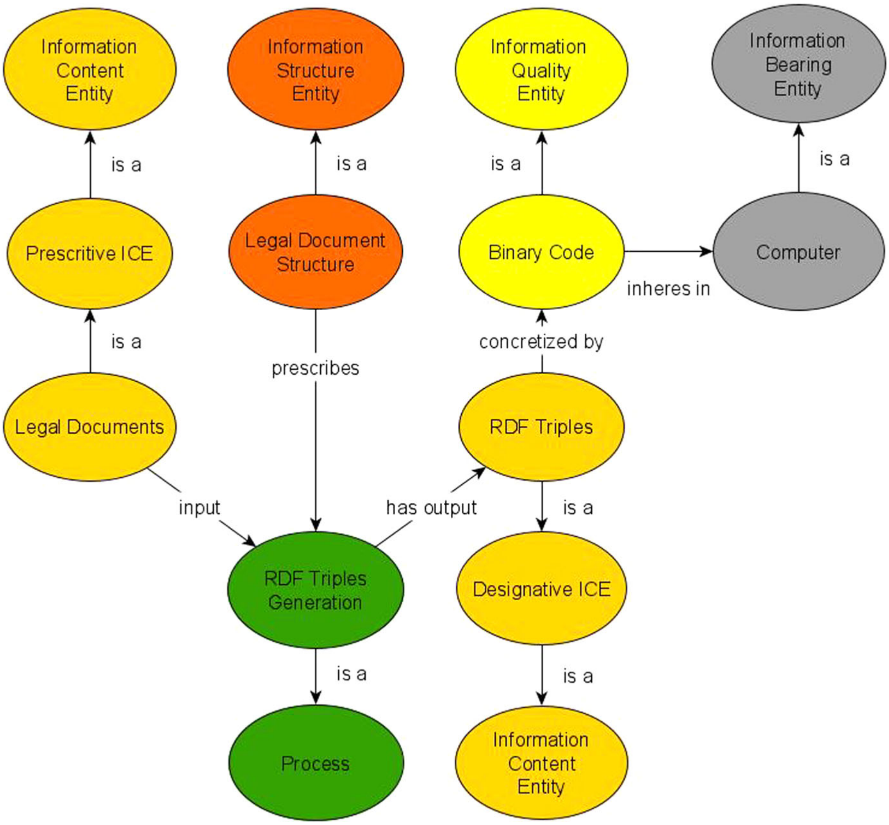

Fig. 9 Ontological view of the approach to create the pursued RDF-based graph. Each color corresponds to the top most entity type

view of the approach to create the pursued graph and the corresponding meaning of the involved entity types. Such an ontological view is depicted in Fig. 9.

The graph is made of RDF triples, which are the output of the Process RDF Triples Generation, as shown in Fig. 9. The Process RDF Triples Generation has as input instances of Legal Document and Legal Document Structure to process them and produce RDF Triples.

In ontological grounds, RDF Triples are of the type Designative Information Content Entity, which has the purpose of assigning a form of identification to someone or something. In the case of RDF Triples that follow the data framework Subject, Predicate and Object, we have that Subjects are assigned to Objects by means of Predicates, representing this way legal documents and their parts. For instance, in Fig. 4, we see that Decree 2556 (Subject), refers to (Predicate) Law 9610 (Object).

According to the Basic Formal Ontology, Generically Dependent Continuants, as is the case of RDF Triples, need to be concretized by patterned arrangements, which are inherited by an Information Bearing Entity. Here the patterned arrangement is a Binary Code, which is an Information Quality Entity that can be inherited by a

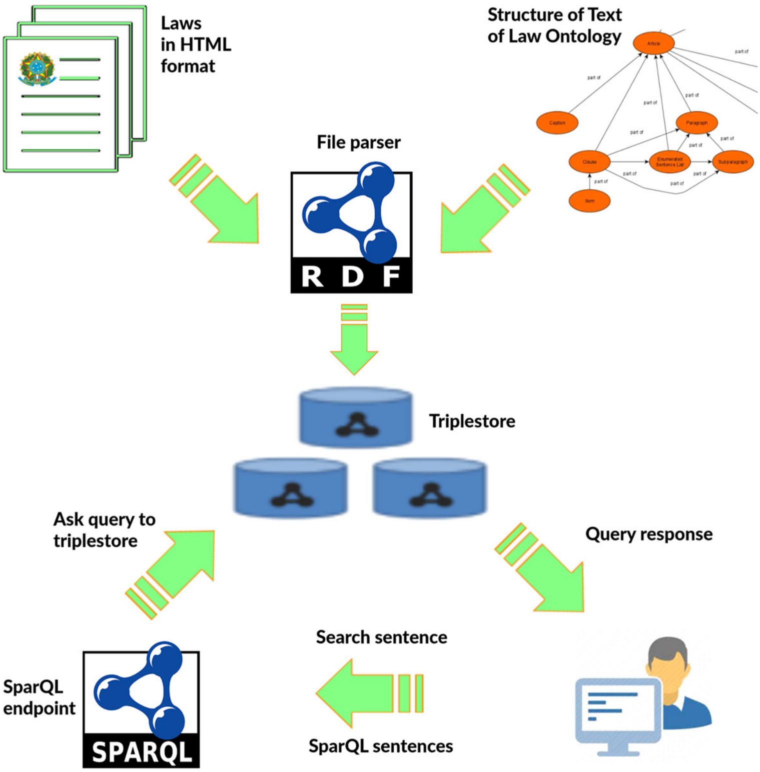

Fig. 10 Implementation structure of the RDF-based graph

Computer. That means to say that RDF triples are stored in computers in the form of files because computers are prepared to inherent this information quality. This way, we provide a complete ontological ground for the creation of the pursued RDF-based graph to representing and searching parts of legal documents.

4.2 Implementation of the RDF-based graph

From the ontological conception presented, we implemented the RDF-based graph, whose steps are illustrated in Fig. 10 and described next:

1. Download of the HTML files containing the legal documents from the Brazilian official repositories.
2. Convert the legal documents from HTML format to RDF using the file parser program we developed and the ontology of the general structure of legal documents.
3. Store the generated RDF triples into a triplestore.
4. Once stored, the RDF triples can be queried with SPARQL.

Table 1 Number of legal norm documents per type

Type	Number of files
Constitution	19
Decrees	29.091
Laws	15.725
Supplementary laws	201
Delegated laws	21
Empire laws	94
Others	1.754
Total	46.905

4.2.1 Download of the HTML files containing the legal documents from the official Brazilian repositories

Though there is no publicly available legal information system, the legal documents are made available to the general public in the form of documents in HTML format, aka web pages. For this work, we used the legal documents available on a portal of the Federal Government.[7]

A total of 46,905 HTML documents were collected in June 2022. The portal separates legal documents by specific types. Table 1 shows the quantity of documents per type. The downloaded files were stored in local directories of a desktop computer. Although there are no additional statistics, all the downloaded HTML files are 2.2 Gb in size, and each file is 49 Kb in size on average.

4.2.2 Convert the legal documents from HTML format to RDF using the File Parser program and the ontology of the general structure of legal documents

The conversion of documents from HTML to RDF format was done through a Java language computer program, here called File Parser, which we created specifically for this purpose. The conversion process is activated through a script that scans the directories containing the HTML files. Each HTML file is passed as an argument to the File Parser which generates the RDF files as described in Algorithm 1.

According to Algorithm 1, each of the HTML files containing the legal documents is converted into RDF triples. The text of each HTML paragraph is handled individually. First, it is inferred what kind of part of legal document is in the paragraph. The types that are detected are those depicted in the ontology of the general structure of legal documents.

As previously mentioned, parts of legal documents can be an article, the header of an article, a paragraph, an item, a clause, etc. The process of associating the entities of the ontology with the generated RDF triples is not automatic. The converter program identifies types as it works through each HTML paragraph. Next, we analyze the existence of explicit links to generate relationships with other parts of the same legal text, called internal links, or relationships with parts of other legal documents, called

[7] https://www.planalto.gov.br/ccivil_03/.

Algorithm 1 File Parser Algorithm

```
Require: $htmlDir$, the directory with downloaded html files
for each $htmlFile$ in $htmlDir$ do
    $triples \leftarrow \emptyset$
    for $htmlParagraph$ in $htmlFile$ do
        extract Ontology Part Kind
        generate RDF $triple$
        $triples \leftarrow triples \cup \{triple\}$
        for each $htmlLink$ in $htmlParagraph$ do
            extract Relationship Type
            generate RDF $triple$
            $triples \leftarrow triples \cup \{triple\}$
        end for
    end for
    if $triples \neq \emptyset$ then
        add prefixes to $triples$
        save $triples$ as a RDF N3 Turtle like formatted file
    end if
end for
```

external links. For each link found, a triple is generated. The predicates of these relationships found come from the ontology defined in Sect. 4.1 (Fig. 4.1. Finally, after processing the entire HTML file as described above, the triple file is saved in Turtle like syntax which is compatible with Notation 3 (N3).

4.2.3 Store the generated RDF triples into a triplestore

The triplestore used was the TDB, which is part of the Apache Jena software project. It is a repository that can be accessed directly through the Jena API or as a SPARQL endpoint. Our File Parser takes each of the serialized RDFs and sends it to the triplestore via the Jena API. After sending the RDFs to the server, 2,128,400 triples were created into the triplestore.

4.2.4 Once stored, the RDF triples can be queried with SPARQL

Access to the triples can be made through a SPARQL endpoint. In this work, we chose the Fuseki server, which is also part of Apache Jena. For the specific experiments of this work, SPARQL queries can be sent directly through the Jena API, or using APIs of other languages, for example, the SPARQLWrapper for Python or directly in the web interface provided by the Fuseki server.

Thus, in order to search for parts of legal documents, we developed the sparqlQuery algorithm, presented in Algorithm 2, which is detailed next, as a way of facilitating the understanding of its operation.

In general terms, Algorithm 2 receives the search terms and returns a set of triples that match the given terms. For each triple returned in a such way, the algorithm searches for neighbouring triples, those triples that are at distance 1. With these new set of neighbours, the algorithm proceed to find for their own neighbours, i.e., it searches for triples up to a depth of 2.

Algorithm 2 sparqlQuery Algorithm

```
Require: $searchSentence$                                  ▷ search terms provided by user
Ensure: $result$                                           ▷ set of returned triples
1: $result \leftarrow \emptyset$
2: $triplesQuery \leftarrow queryFromSentence(searchSentence)$
3: $triples \leftarrow getTriples(triplesQuery)$
4: $result \leftarrow result \cup triples$
5: for $triple$ in $triples$ do               ▷ each triple equals (subject, predicate, object)
6:     $neighboursQuery \leftarrow queryFromURI(subject)$
7:     $neighbours \leftarrow getTriples(neighboursQuery)$
8:     $result \leftarrow result \cup neighbours$
9:     for $neighbour$ in $neighbours$ do
10:        for $uri$ in $\{ subject, object \}$ do
11:            $nextNeighboursQuery \leftarrow queryFromURI(uri)$
12:            $nextNeighbours \leftarrow getTriples(nextNeighboursQuery)$
13:            $result \leftarrow result \cup nextNeighbours$
14:        end for
15:    end for
16: end for
```

To observe how the above algorithm works, we can use the search terms “programa computador registrado” (copyrighted computer program) as an example.

Firstly, an empty set is created to store the result for the search terms at line 1. At line 2, a SparQL query is assembled and assigned to *triplesQuery*. Such a query is presented in Listing 4.

Listing 4 Query for search terms

```
PREFIX lex: <http://localhost:8080/fuseki/>
SELECT ?s ?p ?o
WHERE {
?s lex:hasText ?o .
BIND(str('lex:hasText) AS ?p) .
FILTER ( CONTAINS(?o, 'programa') ) .
FILTER ( CONTAINS(?o, 'computador') ) .
FILTER ( CONTAINS(?o, 'registrado') ) .
}
ORDER BY ?s ?p ?o
```

Then, this query is sent to the triplestore and the result set is updated (lines 3 and 4). From this first set of triples returned, the algorithm proceed to fetch its neighbours. To get this done, for each triple (line 5) a new query is built using its subject as search parameter (line 6). The constructed SparQL query, for lex:L9609/Article3 subject, is shown in Listing 5.

Listing 5 Query for neighbors at distance 1 in the graph

```
PREFIX lex: <http://localhost:8080/fuseki/>
SELECT ?s ?p ?o
{
{
```

```
<lex:L9609/Article3> ?p ?o .
BIND('<lex:L9609/Article3>' AS ?s)
}
UNION
{
?s ?p <lex:L9609/Article3>
BIND('<lex:L9609/Article3>' AS ?o)
}
FILTER(
CONTAINS(STR(?p), 'refersTo')
|| CONTAINS(STR(?p), 'remise')
|| CONTAINS(STR(?p), 'revokes')
|| CONTAINS(STR(?p), 'revokedBy')
|| CONTAINS(STR(?p), 'regulates')
|| CONTAINS(STR(?p), 'regulatedBy')
) .
}
ORDER BY ?s ?p ?o
```

Again, the query is sent to the triplestore and the result set is updated once more (lines 7 and 8). The last steps consist in iterating over the returned neighbours (line 9). Then the subject and object of each triple (line 10) are taken to build a new SparQL query for these elements (line 11) which then is sent to the triplestore (line 12). The triples returned from this step are added to the result set (line 13). After the iterate cycle, the execution of the algorithm is ended.

As can be seen from the above description, the authors opted for a non-trivial approach. A search is made for parts of legal documents that contain a set of search terms, then the neighbours of the returned triples (distance 1) and their close neighbours (distance 2) are searched. The pros of this approach are primarily that it makes it possible to automate the triple search process from search terms. This process could, in the future, be encapsulated in a web service or micro service. In addition, the method provides a cleaner return, that is, triples inherently related to the search terms are searched and this results in a cleaner response graph.

5 Experiments and results

For the purpose of the experiments performed, we used only a part of the ontology shown in Fig. 8. Particularly, we used only articles and their component sections, in such a way that articles are directly linked to legal documents. However, the full use of the ontology of Fig. 8 requires implementing, in the parser program, the generation of triples corresponding to Article's upper structure. This was not done in this work and did not impact the proposed objective. Of course, this improvement can be done in future works.

Table 2 Result set for the search query by terms "programa computador registrado"

Subject	Predicate	Object
lex:D2556/Artigo1	lex:hasText	Art. 1º Os programas de computador poderão a critério do titular dos respectivos direitos, ser registrados no Instituto Nacional da Propriedade Industrial - INPI
lex:D96036/Artigo34	lex:hasText	Art. 34. Os programas de computador já registrados na SEI poderão ser incluídos à vista de requerimento do interessado no cadastro de programas de computador nas categorias correspondentes, observado o disposto na Lei n° 7.646, de 18 de dezembro de 1987, e neste regulamento no prazo de 180 dias, a partir da data de publicação deste Decreto, de acordo com roteiro apropriado, fornecido pela SEI
lex:D96036/Artigo35	lex:hasText	Art. 35. É concedido o prazo de 180 dias para que os programas de computador não registrados na SEI, e que estejam em comercialização no País, se enquadrem neste regulamento. A data de publicação deste Decreto constitui o termo inicial desse prazo
lex:L7646/Artigo4	lex:hasText	Art. 4º Os programas de computador poderão a critério do autor, ser registrados em órgão a ser designado pelo Conselho Nacional de Direito Autoral - CNDA, regido pela Lei nº 5.988, de 14 de dezembro de 1973 e reorganizado pelo Decreto nº 84.252 de 28 de julho de 1979
lex:L9609/Artigo3	lex:hasText	Art. 3º Os programas de computador poderão a critério do titular, ser registrados em órgão ou entidade a ser designado por ato do Poder Executivo, por iniciativa do Ministério responsável pela política de ciência e tecnologia. (Regulamento)

The experiment carried out had the purpose to search for parts of legal documents and their relationships in a dataset of 46,905 legal documents, which generated 2,128,400 RDF triples that were stored on a triplestore.

Table 3 The result of the query to fetch the neighbors of lex:L9609/Article3

Subject	Predicate	Object
lex:D2556	lex:regulates	lex:L9609/Artigo3
lex:D2556/Texto	lex:remise	lex:L9609/Artigo3
lex:L9609/Artigo3	lex:regulatedBy	lex:D2556

For carrying out searches on the triple store, we used sets of terms related to very important issues being discussed at the present time. The terms are in Portuguese because the triples are in Portuguese, but they are followed by their English version.

1. “programa computador registrado” (copyrighted computer program)
2. “propriedade intelectual” (intellectual property)
3. “notícia falsa” (fake news)
4. “fraude financeira” (finance fraude)
5. “acesso informação” (information access)
6. “corrupção” (corruption)
7. “direitos humanos” (human rights)
8. “imigração” (immigration)
9. “droga ilícita” (illicit drugs)
10. “direito consumidor” (consumer rights)
11. “trabalho escravo” (slave work)

We tried all of the above search sentences using Algorithm 2. In particular, the triples returned from the query “programa computador registrado” (copyrighted computer program), as the result of lines 3 and 4 of the algorithm are shown in Table 2.

After that, the algorithm iterates over each triple returned, builds a Sparql query to look for the neighbors of the iterated triple, that is, the neighbors at distance 1 in the graph. As an example, we use the triple whose object is lex:L9609/Article3. For this object, the query of its neighbours is executed (lines 7 and 8). The triples returned from this action are shown in Table 3.

The next loop, starting at line 9, is an analogous process. It iterates over the returned neighbours. For each neighbour, it gets the subject’s uri and the object’s uri of such a triple to get the neighbours at a distance 2. A query for these neighbours is built and executed (lines 11, 12 and 13). The returned triples are added to the *result* set.

In the end, the complete result set for searching “programa computador registrado” are the triples showned in Table 4.

The Algorithm 2 was performed for all search terms listed at the beginning of this section. In a universe of 2,128,400 triples generated from 46,905 legal documents, the amount of triples for parts of legal documents returned for each experiment is shown in Table 5.

It is worth stating that we used only a subset of the General Structure of the Legal Documents Ontology depicted in Fig. 8. This adopted simplification in the generation of triples allowed the representation of legal documents in the form of a graph without losses to recover parts of them, since the part of the ontology omitted refers only to the textual hierarchy of the legal documents.

Table 4 Result set for searching "programa computador registrado"

Subject	Predicate	Object
lex:D2556	lex:regulates	lex:L9609/Artigo3
lex:D2556/Artigo1	lex:hasText	Art. 1º Os programas de computador poderão
		a critério do titular dos respectivos direitos
		ser registrados no Instituto Nacional da Pro-
		priedade Industrial - INPI
lex:D2556/Texto	lex:remise	lex:L9609/Artigo3
lex:D96036/Artigo34	lex:remise	lex:L9609/Artigo3
lex:D96036/Artigo34	lex:hasText	Art. 34. Os programas de computador já re-
		gistrados na SEI poderão ser incluídos, à vista
		de requerimento do interessado, no cadastro de
		programas de computador, nas categorias corres-
		pondentes, observado o disposto na Lei n° 7.646
		de 18 de dezembro de 1987, e neste regulamento
		no prazo de 180 dias, a partir da data de publi-
		cação deste Decreto, de acordo com roteiro
		apropriado, fornecido pela SEI
lex:D96036/Artigo35	lex:hasText	Art. 35. É concedido o prazo de 180 dias para
		que os programas de computador não registrados
		na SEI, e que estejam em comercialização no País
		se enquadrem neste regulamento. A data de publi-
		cação deste Decreto constituio termo inicial
		desse prazo
lex:L7646	lex:revokedBy	lex:L9609
lex:L7646/Artigo4	lex:remise	lex:L5988

Table 4 continued

Subject	Predicate	Object
lex:L7646/Artigo4	lex:hasText	Art. 4º Os programas de computador poderão a critério do autor, ser registrados em órgão a ser designado pelo Conselho Nacional de Direito Autoral - CNDA, regido pela Lei nº 5.988, de 14 de dezembro de 1973, e reorganizado pelo Decreto nº 84.252, de 28 de julho de 1979
lex:L9609	lex:revokes	lex:L7646
lex:L9609/Artigo3	lex:regulatedBy	lex:D2556
lex:L9609/Artigo3	lex:hasText	Art. 3º Os programas de computador poderão, a critério dotitular, ser registrados em órgão ou entidade a ser designado por ato do Poder Executivo, por iniciativa do Ministério responsável pela política de ciência e tecnologia. (Regulamento)

Table 5 Size of the result sets for each search experiment

Exp	Search terms	Number of triples
1	Programa de computador registrado (copyrighted computer program)	12
2	Propriedade intelectual (intellectual property)	251
3	Notícia falsa (fake news)	10
4	Fraude financeira (finance fraude)	17
5	Acesso informação (information access)	273
6	Corrupção (corruption)	102
7	Direitos humanos	362
8	Imigração (human rights) (immigration)	85
9	Droga ilícita (illicit drugs)	24
10	Direito consumidor (consumer rights)	224
11	Trabalho escravo (slave work)	48

6 Conclusion

In this work, we presented a proposal of a RDF-based graph to represent and search for parts of legal documents that are the output for a set of terms that represents the pursued legal information. Such a proposal is well-grounded on an ontological view.

For that, we used the Basic Formal Ontology (BFO) as the starting point and extended it for our own purposes. BFO was used for defining the ontologies that describe the structure of the Brazilian legal system and the general structure of legal documents, as well as to provide the grounds for the implementation of such a graph.

The objective of representing and searching for parts of legal documents through the RDF graph was achieved given that the graph was implemented and experimented, providing more granular retrieval of parts of legal documents than other similar works.

The impact in the area is that the method offers an effective way for organizing legal documents, structuring them through ontologies and using semantic web techniques, as well as enabling the search for useful and objective legal information, since the relevant parts of legal documents, as well as the corresponding legal documents as whole, are recovered.

Based on what has been exposed here, some advances and improvements can be attempted. Far from being an exhaustive list, some of these possibilities are the following:

- Improve and refine the ontology referring to the general structure of federal legal documents in Brazil, see Fig. 8.
- Improve the parser, either to eliminate programming errors of the prototype developed for this work, or to improve its scope.
- Use similarity and natural language processing techniques to improve search results by terms.
- Given that revocations in whole or in part of legal documents exclude the revoked text from the current legal set, it is necessary to deal with this properly in the implemented system. We are currently returning revoked texts and this may generate some noise in the investigation of the legal framework related to a given topic.
- Identify implicit links in the legal documents and the generate the corresponding triples to such relationships.

Apart from that, due to the work of capturing the representative HTML files of legal documents on the official websites and transforming them into RDF as we did here, it could be an initiative of the legislature to make these legal texts available in an open RDF-type format. That seems quite feasible since all legislation is created and published by themselves, which implies that they have both structural and relational knowledge of the various legal documents as whole or of their parts.

All triples generated and stored in the repository for the purpose of this work can be made available through RDF files serialized in N3 Turtle like format. These files can be published under the open linked data principles.

Declarations

Conflict of interest Authors are required to disclose financial or non-financial interests that are directly or indirectly related to the work submitted for publication. Please refer to “Competing Interests and Funding” below for more information on how to complete this section.

References

Allemang D, Hendler J (2011) Semantic web for the working ontologist. Morgan Kaufmann, Boston

Aristotle. 2016. Órganon. Ed. , Translation: Edson Bini (3rd ed.). São Paulo: Edipro

Arp R, Smith B, Spear AD (2015) Building ontologies with basic formal ontology. The MIT Press, Cambridge

Boer A, Hoekstra R, Winkels R (2002) Metalex: legislation in xml. In: Proceedings of JURIX 2002, legal knowledge and information system, pp. 1–10

BRASIL (1942) Decreto-lei no. 4,657, de 4 de setembro de 1942. https://www2.camara.leg.br/legin/fed/declei/1940-1949/decreto-lei-4657-4-setembro-1942-414605-normaatualizada-pe.html. Accessed: 2023-03-08

BRASIL (1998) Lei complementar no 95. de 26 de fevereiro de 1998. https://www2.camara.leg.br/legin/fed/leicom/1998/leicomplementar-95-26-fevereiro-1998-363948-publicacaooriginal-1-pl.html. Accessed: 2022-09-23

Canotilho JG, Mendes GF, Sarlet IW, Streck LL, Leoncy LF (2018) Comentários à constituição do Brasil. Saraiva Educação, São Paulo

Ceusters W Smith B (2015). Aboutness: towards foundations for the information artifact ontology. In: Proceedings of the sixth international conference on biomedical ontology (ICBO), Vol 1515, pp. 1–5. CEUR

de Oliveira F (2017), Obtenção do arcabouço legal por meio da adição das remissões externas aos documentos legais. Master's thesis, ITA - Technological Institute of Aeronautics, São José dos Campos, SP

de Oliveira F de Oliveira J.M.P(2018). Obtenção do arcabouço legal por meio da adição das remissões externas aos documentos legais. In: Proceedings of the XI seminar on ontology research in Brazil and II Doctoral and Masters Consortium on Ontologies, São Paulo, SP, pp. 128–139

Filtz E, Kirrane S, Polleres A (2021) The linked legal data landscape: linking legal data across different countries. Artif Intell Law 29(4):465–539

Gandon FL, Krummenacher R, Han SK, Toma I (2011) Semantic annotation and retrieval: RDF. Springer, Berlin, pp 117–155

Leone V, Caro LD, Erena V (2019) Taking stock of legal ontologies: a feature-based comparative analysis. Artif Intell Law 28:207–235

Machado AL (2013) Modelo Conceitual Formal de Relacionamentos do Ordenamento Jurídico Positivo. Ph. D. thesis, ITA - Technological Institute of Aeronautics, São José dos Campos, SP

Moreno Schneider J, Rehm G, Montiel-Ponsoda E, Rodríguez-Doncel V, Martín-Chozas P, Navas-Loro M, Kaltenböck M, Revenko A, Karampatakis S, Sageder C, Gracia J, Maganza F, Kernerman I, Lonke D, Lagzdins A, Bosque Gil J, Verhoeven P, Gomez Diaz E, Boil Ballesteros P (2022) Lynx: a knowledge-based ai service platform for content processing, enrichment and analysis for the legal domain. Inf Syst 106:101966. https://doi.org/10.1016/j.is.2021.101966

Smith B, et al. (2013) Iao-intel - an ontology of information artifacts in the intelligence domain. In: Proceedings of the eighth international conference on semantic technologies for intelligence, defense, and security, Vol 1097, Fairfax, VA, pp. 33–40. CEUR

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.