Using NLP and Ontologies for Notary Document Management Systems

Flora Amato, Antonino Mazzeo, Antonio Penta and Antonio Picariello
Universitá di Napoli "Federico II", Italy
Dipartimento di Informatica e Sistemistica, via Claudio 21, 80125, Naples
{flora.amato, mazzeo, a.penta, picus}@unina.it

Abstract

In this paper we describe the use of NLP techniques and ontologies as the core for building novel e-gov based information systems, and in particular we define the main characteristics of a document management system in the legal domain, that manages a variety of paper documents, automatically transforming them into RDF statements, for suitable indexing, retrieval and long term preservation. Although we describe a general architecture that can be used for several application domains, our system is particularly suitable for the italian notary realm.

1 Introduction

In almost all legal traditional activities, most of the processes are characterized by the presence of hard paper documents that have to be properly managed: processed, archived and prepared for long term preservation. Despite the introduction of automatic tools, no significant reduction in the volume of paper documents created has been registered, and an intense and extensive dematerialization activity is still necessary: in particular, starting from a collection of documents, coming from a digitalization of paper works related to a particular bureaucratic process, (e.g., documents in public administration offices or legal notary documents), there is still the lack of efficient methodologies and tools in order to automatically *transform a legal digitalized document into a formally structured, machine readable records*.

In order to describe the peculiarities of our work, throughout the paper we will use a running example, as discussed in the following.

Example 1.1 (Notary Document Management Systems). *Let us suppose to analyze a* buying act. *In real estate market, in Italy and also in some other european countries, when someone has the intention of buying or selling a property, such as houses, pieces of lands and so on, a notary document, certifying the property transaction from an individual to another one, is signed. Such document is generally composed by an* introduction part *containing the caption, a part containing the* biographical data *of the individuals involved in the buying act, a section containing* data about the property *and a sequence containing several rules regulating the sales contract. We thus propose a system that: i) detects the several sections containing relevant information (segmentation), and ii) transforms the unstructured information within the retrieved section into a structured document, by means of iii) structural, lexical and domain ontologies.*

This process requires the use of different techniques from interdisciplinary fields: in particular, several efforts have been done regarding *legal ontologies*, both from a theoretical – in order to define legal lexical dictionaries – and for the application point of view, as for instance can be evidenced from the large number of e-gov initiatives in Europe – putting a great emphasis on the study of the *structure* and *properties* of legal information, as well on organization, storage, retrieval, and dissemination within the context of the legal environments. We notice that several works to represent legal knowledge has been proposed, such as: Valente's Functional Ontology of Law [8], Frame-based Ontology of Visser [9], McCarty's Language of Legal Discourse (LLD) [4] and Stamper's Norma [6]. As a consequence of such theories, several ontologies are now available, such as: ON-LINE (Ontology-based Legal Information Environment), DUBA (Dutch Unemployment Benefits Act), CLIME (Cooperative Legal Information Management and Explanation): Maritime Information and Legal Explanation (MILE) and Knowledge Desktop Environment (KDE) [4]. Several approaches based on the wordNet project have been also done: in particular in Italy, JurWordNet [7] is the first Italian legal semantic knowledge base [1].

It is worth noticing that, despite the vast amount of efforts, several challenging problems still remain opened, especially related to the *automatic ontology building process*. The use of Pattern Recognition techniques on the sentence level for the identification of concepts and document classi-

[1]We gratefully acknowledge ITTIG - CNR, Italy, and particularly dott.Tiscornia, for the use of JurWordNet in this work

fication for automatic document description is described in several works, as SCISOR[3] and FASTUS [1].

The paper is organized as follows: in section 2, the general system architecture is outlined; section 3 describes the theory underlying our work, in particular the ontological levels for legal information management; the RDF document building strategy is described in section 4; some implementative details are discussed in section 5 together with several conclusions.

2 System Architecture

Figure 1 shows at a glance the architecture of system. The main functionalities depicted are:.

Text Extractor: this module extracts the plain text from the source file, preserving the document format. The input of the module is a digitalized file, in various formats (pdf, tiff etc) and the output is formatted ASCII text. . *Structural Analysis*: this module performs the preprocessing of digital unstructured text. It identifies the textual macro-structures which allow the recognition of text sections, according to the information provided by the "structural ontology", that represents the organization of the documents in the legal domain. This module contains a "Text Segmentation" algorithm able to cut the document into a set of elements (i.e. paragraphs) on which further processing will be performed. The subdivision of the document into segments of text makes more accurate the further syntactic and semantic analysis.

Linguistic Analysis: this module performs a syntactic annotation of text, by means of a labeling strategy; in particular, each text element is associated to a grammatical category (*verb, noun, adjective)* and to a syntactic role (*subject, predicate, complement, etc*). In order to do that, several traditional NLP components are then used, i.e. a "Stop Word List", in order to eliminate un-relevant words in the sentence, such as pronoums, articles and so on; a "Stemmer", for removing the commoner morphological and inflexional forms from words in Italian language [10]; a "Part of Speach Tagger", for detecting the several grammatical part of a sentence; and a "Syntactic Analyzer", for recognizing the logic-syntactic relation existing between "sintagms". To these aims, we use ontologies based on the ItalWordnet[5] lexical database.

Semantic Analysis: This core module performs novel information extraction techniques. By means of structural, legal domain and lexical ontologies, this module detects concepts and relations among concepts. Our proposed semantic analyzer produces a proper semantic annotation, codified in RDF triples. In particular, it associates an appropriate concept to each discovered single entity.

3 The Knowledge Model

In the legal domain, almost all the documents is still written using natural languages. Even though, the unstructured form of documents follows a well determined sequence: in a notary act, for example, the notaries use a lot of words coming from a domain vocubulary that is a subset of the one used in natural language and in addition they use a certain pre-defined structures and patterns, codified by laws or normative rules. For these reasons, we say that our legal realm manages documents thta are *implicit-structured texts* written in a domain natural language and having to satisfy a number of constraints.

These simple considerations are at the basis of the following definitions.

Definition 3.1 (Structure-UnarySet)**.** *Let us give a domain* $\mathcal{D}^S$*; a* $Structure\text{-}UnarySet$ $(\mathcal{SU})$ *over* $\mathcal{D}^S$ *is the set of unary predicates, called* $structure\text{-}concepts$ *(sc),*

$$\mathcal{SU} = \{sc_1, ...sc_n\}$$

$sc_i \in \mathcal{D}^S$, $i \in \{1..n\}$

Definition 3.2 (Document-Structure-UnarySet)**.** *A* $Document\text{-}Structure\text{-}UnarySet$ $(\mathcal{DS})$ *is a non empty subset of* $\mathcal{SU}$ *containing all the necessary concepts for defining the structure of a given document according to a regular description.*

Definition 3.3 (Structure-BinarySet)**.** *Let us give a domain* $\mathcal{D}^S$*; a* $Structure\text{-}BinarySet$ $(\mathcal{SR})$ *over* $\mathcal{D}^S$ *is the set of binary predicates, called* $structure\text{-}relations$ (sr)*,*

$$\mathcal{SR} = \{sr_1, ...sr_m\}$$

$sr_i \in \mathcal{D}^S$, $i \in \{1..m\}$.

Example 3.1 (Structures example)**.** *Using example 1.1, according to definition 3.1, a possible* $\mathcal{SU}$ *is:* $\{person$, $component$, $date$, $location$, $organization$, $article$, $section$, $biographical\_section$, $notary\_section$, $buying\_act$, $parties\_section\}$*; according to definition 3.2,* $\mathcal{DS}$=$\{article$, $section$, $biographical\_section$, $notary\_section$, $buying\_act$, $parties\_section\}$ *, and according to definition 3.3,* $\mathcal{SR}$=$\{has\_number\_act$, is_part_of, is_kind_of, has_name, $has\_surname$, $has\_section$, $has\_article$ has_sold, is_born_at, $has\_SSN\}$

The following definition also stands.

Definition 3.4 (Base-Document)**.** *Let a* $Paragraphs\text{-}Sections$ (S^P) *be the set of textual line inside a document. A* $Base\text{-}Document$ $(\mathcal{D}^B)$ *is:*

$$\mathcal{D}^B = \{S_1^P, ..., S_m^P\}$$

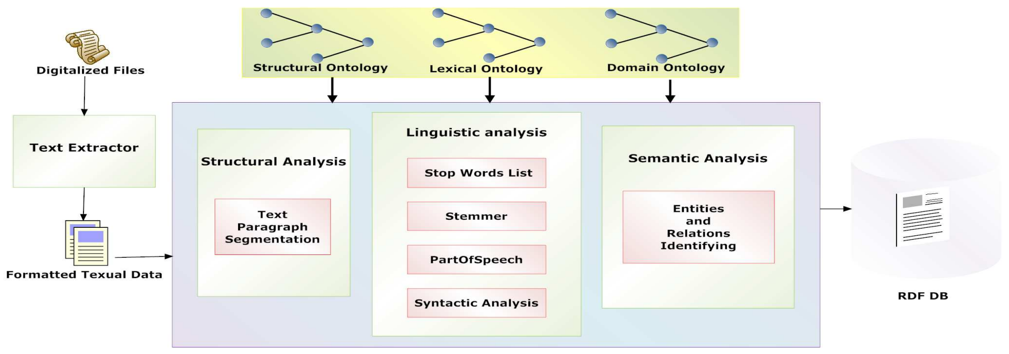

Figure 1. The proposed system

$S_i^P \cap S_j^P \supseteq \emptyset, i, j \in \{1..m\} \wedge i \neq j$.

In other word, a document is a set of text-areas that can be overlapped; note that we can have different $\mathcal{D}^B$, depending on the different set of partition criteria used. In order to capture the knowledge about the structure of the document, let us define a *Structure-TBox*

Definition 3.5 (Structure-TBox)**.** *A Structure-TBox ($\mathcal{ST}$) is a finite set of axioms made up by the elements of $\mathcal{SB}$ and $\mathcal{SU}$, expressed in form of $\mathcal{SHOIQ}(D_n)$ description logic.*

Note that $\mathcal{SHOIQ}(D_n)$ is a particular kind of Description Logic [2] on which several languages such as *OWL-DL* are based. In particular, for the concepts $sc \in \mathcal{DS}$ we define an interpretation $\mathcal{I}'$=$\langle \mathcal{D}^B, .^{\mathcal{I}'} \rangle$, being $.^{\mathcal{I}'}$ a mapping function assigning an element of $\mathcal{D}^B$ to each concept $sc \in \mathcal{DS}$. In this way, each *Paragraphs-Sections* appear as an instance of the concept belonging to $\mathcal{DS}$. We, also note that $sr \in \mathcal{D}^S$ could be an abstract role or a datatype role. This kind of knowledge takes into account the document's *implicit* structure: in fact, in example 3.2, we do not express in *Structure-TBox* the semantic of the roles of the people involved in a particular section. In order to characterize a fragment of TBox associated to a particular section, we introduce the following definitions and we describe them using suitable examples.

Definition 3.6 (TBox-Module)**.** *Let $\mathcal{T}$ be TBox, a TBox-Module $\mathcal{TM}$, is a set of axioms χ that are logically correct and complete.*

Example 3.2 (Structure-TBox)**.** *Considering example 1.1, a Structure-TBox, may be formed by several axioms selected by an expert for the "biographical-section", containing "name" and "surname" of "person", "address", "security social number", i.e.:*
$buying\_act \equiv = 4 has\_section.section$,
$biographical\_section \sqsubseteq section$,
$biographical\_section \equiv \geqslant 2 has.person$,
$person \equiv \exists hasName \sqcap \exists hasSurname \sqcap \exists hasSSN \sqcap \exists is\_born\_in.city$.
In other words, this is the set of axioms of the Structure-TBox that are the TBox-Module related to the biographical_section.

Each *TBox-Module* has to be characterized by means of a proper key, used to find what is the best fragment according to a given score.

Definition 3.7 (KnowledgeKey-Function)**.** *A KnowledgeKey-Function (ψ) is an invertible function:*

$$\psi \colon \mathcal{TM} \longrightarrow k \in \mathcal{K}$$

k being a unique key used to identify $\mathcal{TM}$ and $\mathcal{K}$ the set of these keys.

Example 3.3. *The $\mathcal{TM}$ in example 3.2, is identified by a key k=$\{CODICE \backslash s * FISCALE \backslash s * [A - Z0 - 9 \backslash s], nat[o, a]\}$; in this case, the key is a mixture of a regular expressions. The keys can be formed taking into account also the features extracted from standard natural language process on the text.*

We are now in a position to introduce others concepts related to document D.

Definition 3.8 (Structured-Document)**.** *A Structured-Document $\mathcal{SD}$ is a set of 2-tuples:*

$$\mathcal{SD}=\{\langle S_1^P, k_1 \rangle, ..\langle S_h^P, k_h \rangle\}.$$

S_i^P, and $k_i \in K$, $i \in \{1..h\}$ being Paragraphs-Sections and a knowledge key (obtained by applying the ψ function to a $\mathcal{TM}$) respectively.

Note that different $\mathcal{TM}$ (domain, structure, or lexical) may point to the same *Paragraphs-Sections*, then we could have in $\mathcal{SD}$ some tuples with the same *Paragraphs-Sections* having different keys.

In our vision, the knowledge related to the notary legal domain should be expressed in a *domain ontology* together with a *lexical ontology*. For example, in an italian notary act we could use a specific legal domain ontology built over the top of JurWordNet [7], in addition to a lexical ontology based on ItalWordNet [5].

Given these tree different kinds of knowledge, i.e. structural, domain and lexical knowledge, we use the first one for text segmentation aims, the second and third ones to infer more specific concepts related to the semantic content of the document: in particular, the individuals and the keywords extracted from a section are interpreted as concepts and the relative relations are then inferred using modules built on both domain and lexical ontology.

Eventually, we define a model in order to represent the knowledge contained in each section, in which the document is subdivided.

Definition 3.9 (Knowledge-Chunk). *A $Knowledge\text{-}Chunk$ (kc) is an RDF triple kc=$\langle r, p, a\rangle$, r being a resource name, p being a property name, a being a value.*

Definition 3.10 (KnowledgeChunk-Document). *Let D be a document; a $KnowledgeChunk\text{-}Document$ ($\mathcal{KC}^D$) is:*

$$\mathcal{KC}^D \in \{D, kc_1....kc_l\}$$

kc_i, $i \in \{1..l\}$ being the Knowledge-Chunk and D the related document.

Example 3.4 (Knowledge-Chunk). *For example for the "buyingAct", called $ID\text{-}Do\text{-}01$, we should have three Knowledge-Chunk:*
kc_1=$\langle myxmlns{:}ID\text{-}Do\text{-}01, buyingAct{:}asset$, "*Immobile*"$\rangle$,
kc_2=$\langle myxmlns{:}ID\text{-}Pe\text{-}01, foaf{:}name$," *Ludovico*"$\rangle$,
kc_3=$\langle myxmlns{:}ID\text{-}Pe\text{-}01, buyingAct{:}seller, myxmlns{:}ID\text{-}Do\text{-}01\rangle$, *and* $\mathcal{KC}^D$=$\{ID\text{-}Do\text{-}01, kc_1, kc_2, kc_3\}$

where $myxmlns$,$foaf$ and $buyingAct$ are predefined xml name space.

4 Methods

In this section we describe how the model we have proposed may be used to implement a system for managing notary document, and in particular the segmentation and the RDF extraction procedures.

In this context, text segmentation is the problem of assigning the several extracted fragments to a structured document, according to an *a-priori knowledge* characterizing a single legal document. The first step of our system is thus the extraction of simple fragments of the text, using some partition rules that are dependent from: i) normative prescriptions; ii) tradition of single notary schools; iii) common use of the single notary. A variety of rules may thus be detected, using several criteria.

Example 4.1 (Partition Criteria). *In the following we give an example of several possible criteria that we have formalized using real notaries expertises.*

1. *Starting from the beginning of the document, or from the word following the end of the previous section, every section ends before the keywords 'art.'or 'articolo'(law articles in english).*

2. *To identify each section, use particular tokens, such as "notaio", "vend", "acqui", "compravend", "rep", "repertorio", (in english: notary, sell, buy, article and son on): a section is a portion of text containing one of these tokens. To detect a section, we need to identify the starting and the ending word of it.*

Once extracted several partitions from a given text, text segmentation will be the problem of associating an element of $Base\text{-}Document$ to a $\mathcal{SD}$:

$$\rho : \mathcal{D}^B \longrightarrow \mathcal{SD}$$

This functionality can be implemented in a variety of way; in this paper, we propose an association between an S^P and a k according to a minimum score computed comparing the patterns extracted from text and those represented by the key. A possible implementation of ρ function is given by algorithm 1.

Algorithm 1: $Segm\text{-}Function$ algorithm

```
Input: $D$, $\mathcal{K}_{\mathcal{ST}}$, $\mathcal{K}_{\mathcal{DO}}$, $\mathcal{K}_{\mathcal{LO}}$, $N_C$
$D$ is the document,
$\mathcal{K}_{\mathcal{ST}}$,$\mathcal{K}_{\mathcal{DO}}$,$\mathcal{K}_{\mathcal{LO}}$ is the range of
$KnowledgeKey\text{-}Function$ for the structure Tbox
and domain, lexical ontology respectively,
$N_C$ is the enumeration of the partion criteria,
Output: $\mathcal{DS}$,
$\mathcal{DS}$ is the $Structured\text{-}Document$
begin
    $\mathcal{SD}^* = \{\emptyset\}$
    foreach $i \in N_C$ do
        $scoreVec[i] = 0$;
        $\mathcal{SD} = \{\emptyset\}$;
        $\mathcal{D}^B = getParagraphsSections(D, i)$;
        foreach $S_j^P \in \mathcal{D}^B$ do
            $\langle \mathcal{SD}, i\rangle$,$scoreVec$=$structuredFunction$
            ($S_j^P$, $\mathcal{K}_{\mathcal{ST}}$,$\mathcal{K}_{\mathcal{DO}}$,$\mathcal{K}_{\mathcal{LO}}$,$\mathcal{SD}$,$scoreVec$)
        end
        $\mathcal{SD}^* = \mathcal{SD}^* \cup \{\langle \mathcal{SD}, i\rangle\}$;
    end
    $\mathcal{SD} =$
    $getStructuredDocument(\mathcal{SD}^*, scoreVec)$;
end
```

Note that in algorithm 1, $scoreVec$ is an array of scores; $getParagraphsSections$ takes in input a given Document together with a partition criteria, and returns a $Base\text{-}Document$; $isStructure$ returns true if $Paragraphs\text{-}Section$ matches a key of $TBox\text{-}Module$; $getStructure$ returns the tuple having the best score together with the score itself; $getStructuredDocument$ computes the best $Structured\text{-}Document$ dinamycally built considering those sections having the best sum of the scores previously computed. The segmentation algorithm is followed by an RDF extraction, as described in algorithm 2.

Algorithm 2: $RDF\text{-}Extractor$ $(RDFex)$ algorithm

```
**Input**: $DS$
$DS$ is the $Structured\text{-}Document$.
**Output**: $\mathcal{KC}^D$,
$\mathcal{KC}^D$ is the $KnowledgeChunk\text{-}Document$
**begin**
    $\mathcal{KC}^D = \{D\}$
    **foreach** $\langle S_i^P, k_j \rangle \in \mathcal{SD}$ **do**
        $\mathcal{SM} = \psi^{-1}(k_j)$,
        $kc = InferenceProcedure(\mathcal{SM}, S_i^P)$
        $\mathcal{KC}^D = \mathcal{KC}^D \cup kc$
    **end**
**end**
```

In this algorithm, the $InferenceProcedure$ extracts *knowledge-chunks* from texts using a mix of inference mechanism, concepts and relations extraction. For example, we can use generic rules that are a combination of token patterns and/or syntactic patterns, in order to derive the instances of some concepts or relations, and eventually using subsumption on $TBox\text{-}Module$ for deriving more specific concepts.

Example 4.2 (Putting all together: RDF triples extraction)**.** *Starting from the running example document, the system extracts the relevant information and, the results are presented in a* RDF *triples containing the attributes identified into the buying-selling document.*

In particular the system extracts from one hand, several triples from notary act between the notary and the people involved into the buying-selling process with their generalities and in particular who is the seller and who is the buyer and, on the other hand, the related relationship property.

5 Discussions and Conclusions

We have implemented a prototype system in JAVA on top of the Oracle 10g back ends that is able to manage RDF technology. The system implements all the operations described in this paper and uses the Jena API and the cited ItalWordNet and JurWordNet for the italian and juridic languages respectively. Due to the complexity of the system and knowledge from specialized domains, we are planning a large experimentation using the very large italian notary documents collection.

References

[1] J. R. Hobbs, D. Appelt, M. Tyson, J. Bear, and D. Israel. Sri international: Description of the fastus system used for muc-4. *Fourth Message Understanding Conference, Morgan Kaufmann*, pages 143–147, 1992.

[2] Ian Horrocks. Owl: A description logic based ontology language. In Maurizio Gabbrielli and Gopal Gupta, editors, *ICLP*, volume 3668 of *Lecture Notes in Computer Science*, pages 1–4. Springer, 2005.

[3] P.S. Jacobs and L.F. Rau. Scisor: Extracting information from on-line news. *Comm ACM*, 33(11):88–97, 1990.

[4] L. T. McCarty. A language for legal discourse i. basic features. In *ICAIL '89: Proceedings of the 2nd international conference on Artificial intelligence and law*, pages 180–189, New York, NY, USA, 1989. ACM.

[5] A. Roventini. Italwordnet: Building a large semantic database for the automatic treatment of the italian language. *In Zampolli, A., Calzolari, N., Cignoni, L. (eds.), Computational Linguistics in Pisa, Special Issue of Linguistica Computazionale*, Vol. XVIII-XIX, 2003.

[6] R.K Stamper. The role of semantics in legal expert systems and legal reasoning. *Ratio Juris*, 4(2):219–244, 1991.

[7] D. Tiscornia. Some ontological tools to support legal regulatory compliance, with a case study. *Workshop on Regulatory Ontologies and the Modeling of Complaint Regulations (WORM CoRe 2003) Springer LNCS*, November 2003.

[8] A. Valente and J. Breuker. A functional ontology of law, 1994.

[9] P. Visser. The formal specification of a legal ontology, 1996.

[10] E. Zanchetta and M. Baroni. Morph-it! a free corpus-based morphological resource for the italian language. *Proceedings of Corpus Linguistics 2005*, pages 23–32, 2005.