Knowledge Representation and Management for E-Government Documents

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 the last few years bureaucratic procedures didn't show a significant reduction in the volume of paper documents created. In order to reduce the huge amount of space for archiving and preserving documents and to speed up the secarh process, a semantic-based dematerialization process should be performed. In this paper we describe a novel system that manages several kind of bureaucratic documents in the e-gov domain, automatically extracts several interesting information and produces a suitable semantic representation that may be considered as the first step towards a full automated document management system.

1 Introduction

The presence of a great amount of information is typical for bureaucratic processes, such as the ones related to public administrations. Such information is often recorded on papers, or in a digital but not unique format, and the related management process is not well-structured and very expensive, both in terms of space used for storing documents, and in terms of time spent for searching documents in archives.

In addition, the manual management of these documents is absolutely not error-free. The aims of this paper is the definition and the design of methodologies and techniques for synctatic-semantic documents management, and in particular, for information retrieval aims. Text processing is very interesting for e-government related activities: public or private government structures, in general, might be very interested in this kind of processes.

The dematerialization activity uses 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 struc-

ture 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 [1], Frame-based Ontology of Visser [2], McCarty's Language of Legal Discourse (LLD) [3] and Stamper's Norma [4]. 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) [3]. Several approaches based on the wordNet project have been also done: in particular in Italy, JurWordNet [5] 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 classification for automatic document description is described in several works, as SCISOR[6] and FASTUS [7]. In the system BREVIDOC, documents are automatically structured and important sentences are extracted. These sentences are classified according to their relative importance [8]. From the Natural Language Processing (NLP) point of views, legal research concentrates on the automatic description of documents. In particular, the main focuses are: development of thesauri, machine learning for features recognition, disambiguation of polysems, automatic clustering and neural networks. The most important systems are FLEXICON, KONTERM, ILAM, RUBRIC, SPIRE, the HYPO extension [9] and SALOMON. 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 *(Notary Documents)*. Let us consider the Italian juridic domain, and in particular the notary one: a notary is someone legally empowered to certify the legal validity of a document. 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. Consider for example the Italian sales contract fragment, proposed in figure 1; an Italian reader can easily detect the areas concerning the caption, the personal data and the property attributes. In a similar

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

way, we 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.

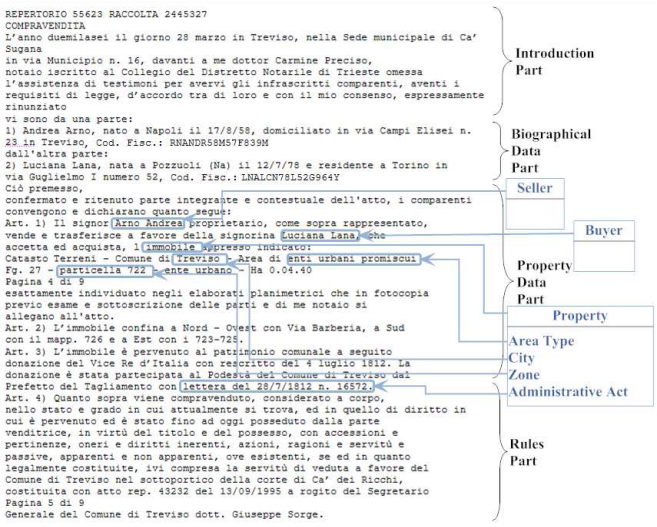

Fig. 1 An example of Notary Documents

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 and, eventually, some conclusions are discussed in section 5.

2 System Overview

In order to describe the main functionalities and characteristic of the proposed work, figure 2 shows at a glance the architecture of system. In the following we will briefly discuss the main parts of the system. *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, such as a pdf file, and the output is formatted textual data[2]
Structural Analysis: this module performs the preprocessing of digital semi-structured 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

[2] in ASCII format

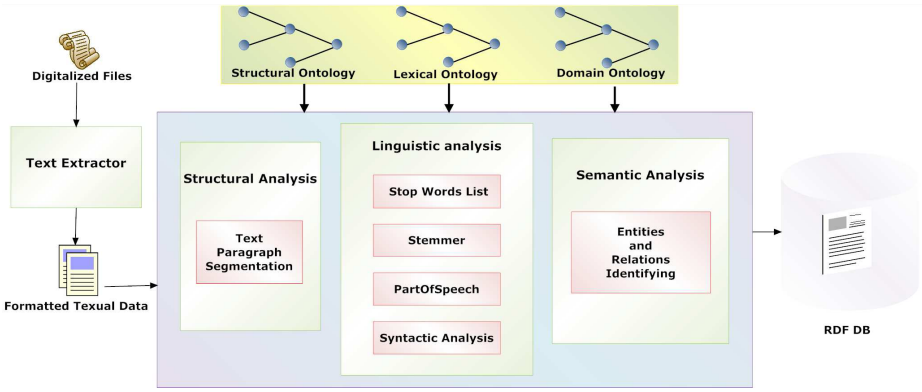

Fig. 2 The proposed system

a *Text Segmentation*, 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.

Lexical Analysis: this module performs a syntactic-semantic annotation of text, by means of a labeling strategy; in particular, each text element is associated to a grammatical category (*verb, noun, adjective, and so on*) 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 Ital-Wordnet[11] 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 Document Representation

In the legal domain, almost all the documents is still written using natural languages on hard papers. Even though, the unstructured form of document follows a well determined sequence: in a notary act, for example, the notaries use a use a certain pre-defined structure, that is codified by laws or normative rules.

For these reasons, we say that our legal realm manages semi-structured documents written in a simplified natural language.

Let us introduce some preliminary symbol: a *Structure-UnarySet* ($\mathscr{SU}$) over a domain $\mathscr{D}^S$ is the set of unary predicates, called *structure-concepts* (*sc*), a *Document-Structure-UnarySet* ($\mathscr{DS}$) is a non empty subset of $\mathscr{SU}$ containing all the necessary concepts for defining the structure of a given document according to a regular description; the *Structure-BinarySet* ($\mathscr{SR}$) over domain $\mathscr{D}^S$ is the set of binary predicates, called *structure-relations* (*sr*).

According with the introduced notation, we can describe the legal document at different levels, such as the *Base-Document* ($\mathscr{D}^B$) that is the set of textual lines inside a document, called *Paragraphs-Sections* (S^P) : these lines specify a text-areas that can be overlapped; note that we can have different $\mathscr{D}^B$, depending on the different set of partition criteria used. We use a *Tbox* defined as a *Structure-TBox* composed by a finite set of axioms, made up by the elements of $\mathscr{AB}$ and $\mathscr{SU}$, expressed in form of $\mathscr{SHOIQ}$ (D_n) description logic, for capturing the knowledge about the structure of the documents. In order to characterize a fragment of our TBox $\mathscr{T}$ associated to a particular section, we use *TBox-Module* ($\mathscr{T}.\mathscr{M}$) defined as restriction of the initial set of axioms χ.

Example 2 (Structure-TBox). Considering example 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.:
$buying\_act \equiv = 4 has\_section.section$,
$biographical\_section \subseteq section$,
$biographical\_section \equiv 2 has.person$,
$person \equiv \exists\ hasName \cap \exists\ hasSurname \cap \exists\ hasSSN \cap \exists\ is\ born\ in.city$.
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* is characterized by means of a proper key, used to find what is the best fragment according to a given score; we thus use the following invertible function, *KnowledgeKey-Function* (ψ):

$$\psi: \mathscr{T}.\mathscr{M} \rightarrow k \in \mathscr{K}$$

k being a unique key used to identify $\mathscr{T}.\mathscr{M}$ and $\mathscr{K}$ the set of these keys.The $\mathscr{T}.\mathscr{M}$ in example 2, is identified by a key k={$CODICE\backslash s * FISCALE\backslash s *$[A– Z0 – 9\ s], nat [o, a]}; in this case, the key is a mixture of a regular expressions. The patterns in the keys can be selected taking into account also the features extracted from standard natural language process on the text.

We are now in a position to introduce what we mean for a structured document related to the document *D*. A *Structured-Document* $\mathscr{SD}$ is a set of 2-tuples:

$$\mathscr{SD} = \{?S_1^P, k_1?, .. ?S_h^P, k_h?\}.$$

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 $\mathscr{T}.\mathscr{M}$) respectively. Note that different $\mathscr{T}.\mathscr{M}$ (domains, structure, or lexical) may point to the same *Paragraphs-Sections*,then we could have in $\mathscr{SD}$ some tuples with the same Paragraphs -Sections but differnt keys.

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 are also used 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 both domain and lexical ontology modules.

Eventually, we define the knowledge associated to the documents, in terms of Knowledge-Chunk,kc.(kc) is an triple defined according with the *Model and Syntax of Resource Description Framework (RDF) Specification*. The final description of the legal document, *KnowledgeChunk-Document*, is :

$$\mathscr{KC}^{D} \in \{D, kc_1 .... kc_1\}$$

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

Example 3 (*Knowledge-Chunk*). For example for the "buyingAct", called *ID-Do*-01, we should have three Knowledge-Chunk:
kc_1=$\langle$*myxmlns*:*ID-Do*-01,*buyingAct*:*asset*,"*Immobile*"$\rangle$,
kc_2=$\langle$*myxmlns*:*ID-Pe*-01, *foaf*:*name*,*"Ludovico"*$\rangle$,
kc_3= $\langle$*myxmlnsID-Pe*-01, *buyingAct*:*seller*,*myxmlns*:*ID-Do*-01$\rangle$, and
$\mathscr{KC}^{D}$={*ID-Do*-01, kc_1, kc_2, kc_3}
where myxmlns, foaf and buyingAct are predefined xml name space.

4 Information Extraction from burocratic document

In this section we describe the several algorithms that are used in our system. The first algorithm we discuss is the text segmentation algorithm.

In our model, text segmentation is the problem of assigning the several extracted fragments to a structured document, according to a the knowledge characterizing the legal document itself.

The first step we propose is that of extracting 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. In the following we give an example of several possible criteria that we have retrieved by real notaries expertise. In particular, we use the following criteria

1. starting from the beginning of the document, or from the word following the end of the previous section, every section is ended by a punctuation character;

2. 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).
3. to identify each section, we use particular tokens, 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; we thus use the following procedure: let us give three tokens in the document: $T_{i-1}, T_{\mathrm{i}}, T_{\mathrm{i}+1}$, in order to identify the starting word of the section relative to $\mathrm{T_i}$, we consider the interval $[T_{i-1}, T_i]$ built using the sequence of words appearing in the document between $T_{\mathrm{i}-1}$ and T_i; we individuate the word w_{middle} located in the middle of this interval. Now we try to find punctuation mark ‘:’closer to w_{middle}; if such mark doesn’t appear in the interval, we look for ‘.’, else for ‘;’or,even, ‘,’, and consider the first word after this. If the interval doesn’t contain any punctuation mark, we simply use as the $\mathrm{w_{middle}}$ word for the section related to $\mathrm{T_i}$. Similar reasoning, on the interval $[T_{\mathrm{i}}, T_{\mathrm{i}+1}]$ is done to determinate the ending word of the section.

In figure 3 we show an example of applying three initial partition criteria on the same act fragment. Once extracted several partitions from a given text, the following definition describe a suitable general function for text segmentation purposes.

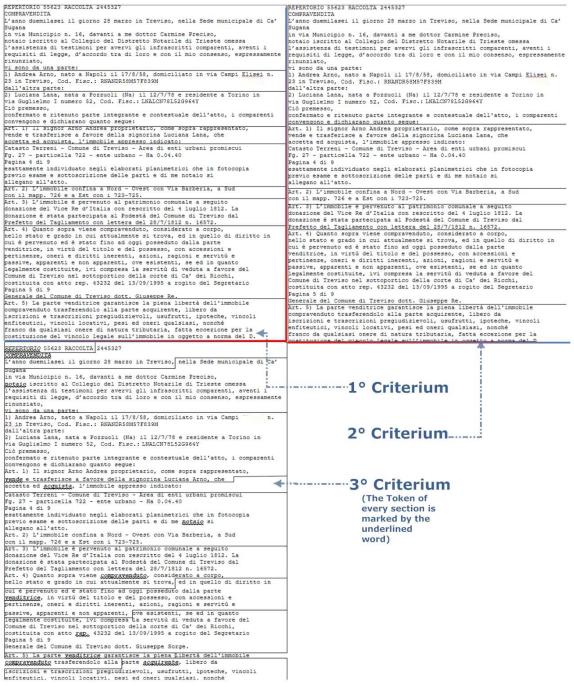

Fig. 3 Application of tree on the same Act fragment

A Segm-Function (ϱ) is a function that associates an element of Base-Document to a $\mathscr{SD}$:

$$\varrho : \mathscr{D}^B \rightarrow \mathscr{SD}$$

Note that a Segm-Function may 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.

A possible implementation of ϱ function is given by algorithm 1.

Algorithm 1 : Segm-Function algorithm

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

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-Document*; *structureFunction* is a function that has the role of matching a *Paragraphs-Section* with one of *TBox-Module* in input and and of retrieving a the tuple having the best score together with the score itself; *getStructuredDocument* computes the best *Structured-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.

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-Module* for deriving more specific concepts.

Example 4 (Putting all together: RDF triples extraction). In this subsection we present an example in order to show how the system works. Starting from the run-

Algorithm 2 : *RDF-Extractor* (*RDFex*) algorithm

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

ning 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. As shown in the figure 4, the system has extracted from one hand,

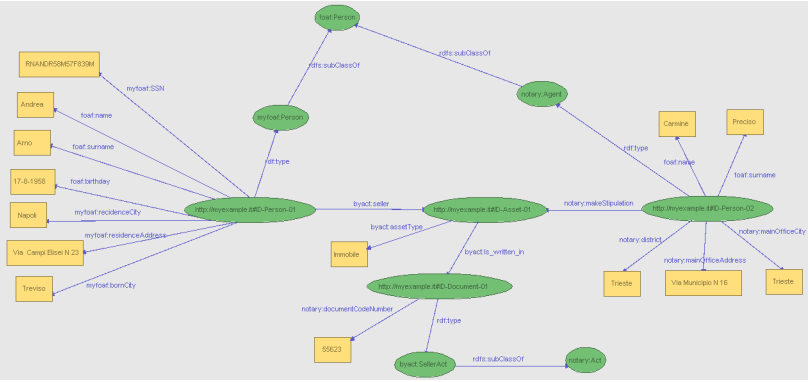

Fig. 4 A section of *RDF* graph extracted from a Notary Act in fig 1

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 Conclusion and future works

In this paper we have presented a general system as the core of an e-gov information system in the notary domain. In particular, we propose the use of different levels of ontology and of NLP techniques in order to transform semi-strucutured documents into structured *RDF* triples and we described suitable algorithms that are able to

segmentate and to extract relevant information from notary documents. The experimental section has shown very encouraging results. Future works will be devoted in developing index methodologies for semantic retrieval purposes.

References

1. Valente, A., Breuker, J.: A functional ontology of law (1994)
2. Visser, P.: The formal specification of a legal ontology (1996)
3. McCarty, L.T.: A language for legal discourse i. basic features. In: ICAIL '89: Proceedings of the 2nd international conference on Artificial intelligence and law, New York, NY, USA, ACM (1989) 180–189
4. Stamper, R.: The role of semantics in legal expert systems and legal reasoning. Ratio Juris 4(2) (1991) 219–244
5. Tiscornia, D.: 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)
6. Jacobs P S, R.L.F.: Scisor: Extracting information from on-line news. Comm ACM 33(11) (1990) 88–97
7. et al, H.J.R.: Sri international: Description of the fastus system used for muc-4. Fourth Message Understanding Conference, Morgan Kaufmann (1992) 143–147
8. et all, M.S.: A full-text retrieval system with a dynamic abstract generation function. in Proc SIGIR 94 (1994) 152–161
9. Bruninghaus St, A.K.D.: Finding factors: Learning to classify case opinions under abstract fact categories. in Proc ICAIL'97 (1997) 123–131
10. Zanchetta, E., Baroni, M.: Morph-it! a free corpus-based morphological resource for the italian language. Proceedings of Corpus Linguistics 2005 (2005) 23–32
11. Roventini, A.: 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)