Using dialogues to access semantic knowledge in a web legal IR system

Paulo Quaresma and Irene Rodrigues
pq|ipr@di.uevora.pt,
Departamento de Informática, Universidade de Évora, Portugal

Abstract.
We present a dialogue system that enables the access in natural language to a web legal information retrieval system. The documents in the IR system are composed by the set of documents produced by the Portuguese Attorney General. These documents were analyzed and an ontology describing their structure was defined. Then, they were automatically parsed and a (partial) semantic structure was created. The ontology and the semantic content was represented in the OWL language.

The proposed system has the capability of inferring user attitudes and is able to reason about them using the documents semantic content.

An example of a user interaction session is presented.

1 Introduction

There is a growing need for tools that enable citizens to access information in documents using natural language sentences queries. One of the main problems to build such a system is to obtain the knowledge necessary to perform the different analysis stages of natural language sentences.

We propose to use a semantic web language to model the document knowledge and to define an ontology representing the main classes of domain objects, their properties and their relations. Then, the documents can be analyzed and their semantic content can be represented. At the moment it is only possible to partially represent the documents semantic content because there is a need for more complete ontologies and more powerful natural language analyzers.

As basic semantic language we are using the OWL (Ontology Web Language) language, based on the previous DAML+OIL (Darpa Agent Markup Language - [10]) language, which was defined using the RDF (Resource Description Framework - [6, 2]) language. Using OWL it is possible to represent the documents structure and some of its semantic content. Moreover, the user's natural language queries can be semantically analyzed accordingly with the base ontology.

Using this approach, the dialogue system that we propose is able to supply adequate answers to the Portuguese Attorney General's Office documents database (PGR).

For instance, in the context of an user interrogation searching for information on state pensions for relevant services to the country, the following question could be posed:

Who has pensions for relevant services?

The user is expecting to have as an answer some characteristics of the individuals that have that kind of pensions. He does not intend to have a list of those individuals or the documents that refer to the act of attributing such pension.

In order to obtain the adequate answer our system must collect all individuals referred in the documents database that have those pension and then it must supply the common characteristics to the user in a dialogue.

As it was referred, our system must have an adequate representation of pensions and individuals in a semantic web ontology and all documents must have the adequate labels in the semantic web language representing the knowledge on 'pensions' and 'Individuals' conveyed by the document.

The answer to the above question could be:

'Individuals that were agents of an action putting their lives at risk'

This information can be obtained by extracting what those individuals have in common or by reading the Portuguese law. The possibility of extracting what are the common characteristics of a set of objects is a powerful tool for a question answering system. This behavior can be achieved by choosing an adequate ontology to represent the objects including the events referred by the documents.

The remainder of this article is structured as follows: in section 2, the Semantic Web language is described. In section 3 the overall structure of the system is presented; section 4 deal with the semantic/ pragmatic interpretation. In section 6 a more extensive example is presented and, finally, in section 7 we discuss some current limitations of the system and lay out possible lines of future work.

2 Web semantics

The documents domain knowledge was represented using a semantic web language. The first step was to define an ontology adequated for the domain and to represent it in the OWL language.

In a previous work [9] we have proposed a methodology to automatically create an ontology from a set of documents. In this methodology, documents are analyzed using NLP techniques and all name entities and verbs are extracted and related by *action* objects. In this work we use the results obtained by this methodology but the ontology was manually improved by juridical experts for some sub-domains.

In fact, we have selected a domain of the Portuguese Attorney General documents – pensions (granted or refused) – and, in this domain, we have selected smaller sub-domains, such as, pensions for firemen, and militaries.

As an example, the $Individual$ class is presented below (only some of the class attributes are shown – code, name, profession):

```
<owl:Class rdf:ID="Individual">
        <owl:label>Individual</owl:label>
</owl:Class>
<owl:DatatypeProperty rdf:ID="individualCode">
        <owl:domain rdf:resource="#Individual"/>
        <owl:type rdf:resource="&owl;FunctionalProperty"/>
        <owl:range rdf:resource="xsd;integer"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="individualName">
        <owl:domain rdf:resource="#Individual"/>
        <owl:type rdf:resource="&owl;FunctionalProperty"/>
        <owl:range rdf:resource="xsd;string"/>
</owl:DatatypeProperty>
```

```
<owl:ObjectProperty rdf:ID="individualProfession">
        <owl:domain rdf:resource="#Individual"/>
        <owl:range rdf:resource="#Profession"/>
</owl:ObjectProperty>
```

After defining the ontology, the documents needed to be analyzed and their semantic content represented in OWL. This step was also done is a semi-manual approach: first, documents were parsed and some semantic information was extracted (as referred in [9]); then, the results were manually improved for the chosen sub-domains.

Complete automatic document semantic representation is a quite complex open problem and, as a consequence and in the scope of this work, we have decided to apply a semi-automatic approach.

These two steps (ontology + document semantic representation) are the basis of the proposed question answering system and they allow the implementation of other steps, such as, the semantic/pragmatic interpretation of queries and the dialogue management.

3 Natural Language Dialogue System

In order to answer user queries the system has to analyze the sentence, to access the documents database(s) and, finally, it has to build a comprehensive answer.

The analysis of a natural language query is split in four subprocesses: Syntax, Semantics, Pragmatics, Dialogue manager.

Syntax Analysis: our syntactic interpreter was built the PALAVRAS parser from E. Bick [1]. This parser was developed in the scope of a large project and it is able to handle 22 different languages.

The parser uses a set of syntactic rules that identify the Portuguese sentence structures and tries to match these rules with the input sentence(s).

As an example, the following sentence:

“Who has a pension for relevant services?”

Has the following structure:

```
phrase([np([det(who,_+_+_), n('individual',_+s+m)]),
        vp(v('have',3+p+_)),
           args_v([np([det(a,_+p+_), n('pension',_+s+_),
             pp(for,np([name('relevant services',_+s+m)]))])]]).
```

Semantic Interpretation: each syntactic structure is rewritten into a First-Order Logic expression. The technique used for this analysis is based on DRS’s (Discourse Representation Structures)[5].

This technique identifies triggering syntactic configurations on the global sentence structure, which activates the rewriting rules. We always rewrite the pp’s by the relation ’rel(A,B)’ postponing its interpretation to the semantic pragmatic module.

The semantic representation of a sentence is a DRS that is built with two lists, one with the new sentence rewritten and the other with the sentence discourse referents.

For instance, the semantic representation of the sentence above is the following expression:

individual(A), pension(B), name(C,’relevant services’), rel(B,C), have(A,B).

and the following discourse referents list:

[ref(A,p+_+_,who),ref(B,s+_+_,undef),ref(C,p+_+_,undef)]

4 Semantic/Pragmatic Interpretation

The semantic/pragmatic module receives the sentence rewritten (into a First Order Logic form) and tries to interpret it in the context of the document database information (ontology).

In order to achieve this behavior the system tries to find the best explanations for the sentence logic form to be true in the knowledge base for the semantic/pragmatic interpretation. This strategy for interpretation is known as "interpretation as abduction" [4].

The knowledge base for the semantic/pragmatic interpretation is built from the Semantic Web description of the document database. The inference in this knowledge base uses abduction, restrictions (GNU Prolog Finite Domain (FD) constraint solver) and accesses to the document databases through an Information Retrieval Agent. This process was also described in more detail in [8].

From the description of the class *pension*, the KB has rules for the interpretation of the predicates: *pension(A)* and *rel(A,B)*. Suppose there exists the following description of the class $Pension$ and of two subclasses[1]:

```
<owl:Class rdf:ID="Pension">
     <owl:label>Pension</owl:label>
</owl:Class>
<owl:DatatypeProperty rdf:ID="pensionCode">
     <owl:domain rdf:resource="#Pension"/>
     <owl:type rdf:resource="&owl;FunctionalProperty"/>
     <owl:range rdf:resource="xsd;integer"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="individual">
     <owl:domain rdf:resource="#Pension"/>
     <owl:type rdf:resource="&owl;FunctionalProperty"/>
     <owl:range rdf:resource="#Individual"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="event">
      <owl:domain rdf:resource="#Pension"/>
      <owl:range rdf:resource="#Event"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="supportDocuments">
      <owl:domain rdf:resource="#Pension"/>
      <owl:range rdf:resource="#DocumentList"/>
</owl:DatatypeProperty>

<owl:Class rdf:ID="Pension_relevant_services">
      <owl:label>Pension relevant services</owl:label>
      <rdfs:subClassOf rdf:resource="#Pension"/>
</owl:Class>
<owl:Class rdf:ID="Retiring_pension">
       <owl:label>Retiring Pension</owl:label>
       <rdfs:subClassOf rdf:resource="#Pension"/>
</owl:Class>
```

This description means that "pension" is a class which has some properties: the individual that gets the pension, events describing the actions supporting the pension, and a list of supporting documents. Moreover, "pension" has two sub-classes: retiring pensions and relevant services pensions.

[1] Due to its complexity, in this paper we only present a small subset of the complete ontology.

Using the ontology, a set of rules was automatically produced enabling the semantic/ pragmatic interpretation of a sentence like " pension" as the Predicate Logic expression *pension(A,_,_,_,_)*.

This description also gives rise to rules allowing for the interpretation of noun phrases such as "retiring pension".

```
rel(A,B) <-
  pension(A),
  name_is(B, [retiring,pension]),
  abduct(pension_retiring(A,_,_,_,_)).
```

From the previous description of the class Individual (in section 2) the KB has rules that will allow the interpretation of the noun "Person" and of noun phrases such as: "Individual name", "Individual Profession". One of the generated rules is:

```
rel(B,A) <-
  individual(B),
  profession(A)
  abduct(individual(B,_,A,_)).
```

This rule enables us to obtain the expression *individual(B,_, A,_)* as the interpretation of the noun phrase "profession of individual".

During the semantic/pragmatic interpretation the evaluation of a predicate like "Individual(A)" is done by an access to the Semantic Web documents. The result of such an evaluation is the constraint of variable A to database identifiers of objects from class individual.

The interpretation of names, such as, *name(A,pension)*, is done by accessing the documents database in order to collect in (constraint) A all entities identifiers that have in their name the word 'pension'.

The result of interpreting the sentence represented by [2]:
individual(A),pension(B),name(C,'relevant services'),rel(B,C),rel(A,B)
[ref(A,s+_+_,what),ref(B,s+_+_,undef),ref(C,s+_+_,undef)]
is the following expression:
pension_relevant_services(B,A,_,_,_), individual(A,_,_,_). Where:
- $B =_{\#} (1046..1049 : 1345 : 1456..1457)$ – B constrained to all pension for relevant services.
- $A =_{\#} (7001...7852)$ – A is constraint to individuals

The above LP expression contain the possible interpretations of the sentence in the context of our documents database.

The dialogue manager is responsible to interact with the user by supplying him an answer or by posing him pertinent questions.

5 Dialogue Manager

The dialogue manager must recognize the speech act associated with the sentence (in this domain it can be an $inform$, a $request$, or a $askif$ speech act), to model the user attitudes (intentions and beliefs), and to represent and to make inferences over the dialogue domain.

In order to achieve this goal the system needs to model the speech acts, the user attitudes (intentions and beliefs) and the connection between attitudes and actions. This task is achieved through the use of logic programming framework rules (see [7] for a more detailed description of these rules).

[2] The interpretation of $A\ have\ B$ is the same of $B\ of\ A$, so $have(A, B)$ is equivalent to $rel(A, B)$

In this framework, after having accessed the textbase, the system may have a multiple solution and it may need to start a clarification sub-dialogue. In the clarification sub-dialogue, the systems asks the user to select one of the possible solutions. In order to collaborate with the user we have defined a cluster predicate that tries to aggregate the solutions into coherent sets. The strategy behind this predicate is to aggregate the solutions accordingly with the range of property values of the selected objects. For instance, in the presented example the selected individuals might be clustered by their profession, or by their support documents, or by the events in which they are actors. In the next section, this strategy will be described in more detail.

6 Example

Considering the already presented question:

Who has pensions for relevant services?

The dialogue manager receives this sentence semantic/pragmatic interpretation, as we presented in the previous sections it will be the following expression:

```
pension_relevant_services(B,A,_,_,_), individual(A,_,_,_).
```

with the following restrictions:

- $B =_{\#} (1046..1049 : 1345 : 1456..1457)$ – B is constraint to all pension for relevant services.
- $A =_{\#} (7001...7852)$ – A is constraint to individuals

After having the sentence re-written into its semantic representation form, the speech act is recognized and we'll have:

```
request(user, system, inform(user, system,
        [pension_relevant_services(pensionCode=B,individual=A)]))
```

Using the "request" and the transference of intentions rules (described in [7]) we'll have:

```
intention(system,inform(system, user,
        [pension_relevant_services(pensionCode=B,individual=A)]))
```

Now, the system will access the databases and it will retrieve a set of solutions. Suppose there are several possible solutions. We'll have A and B constrained to related individuals and pensions for relevant services:

- $B =_{\#} (1046..1049 : 1345 : 1456..1457)$
- $A =_{\#} (7030...7842 : 7850)$ – A is constrained to individuals that have pension for relevant services.

As a consequence of having several solutions, a sub-dialogue will be started and the solutions will be aggregated in clusters.

```
cluster([pension_relevant_services(pensionCode=B,individual=A)],C).
```

The $cluster$ rule identifies the variable which is the focus of the query (obtained in the syntactical analysis) and aggregates the property values for the associated objects. For instance, in this example it will detect that the query is about individuals (variable A) and it tries to cluster its constrained values accordingly with their professions, events, and documents relation. After having clustered the property values, the system uses an heuristic to choose the property that better divides the objects (by better we mean that the cardinality of the obtained sets has the same magnitude order) and it performs the ask_select action.

In this example the answer might be:

'Individuals that are firemen, and militaries'.

Or, using another property (event list):

'Individuals that were agents of an action putting their lives at risk'

7 Conclusions and Future Work

The dialogue system described in this paper is still a prototype but it will be made available to all users in the context of the Portuguese Attorney General's web information retrieval system (http://www.pgr.pt).

Note that, whenever the system is unable to find an interpretation of the user query in the context of the ontology knowledge, then it acts as a traditional information retrieval system using a word based matching algorithm.

Clearly, and due to its complexity, many modules have aspects that may be improved:

- The coverage of the semantic analyzer, such as, plurals, quantifiers, co-references and anaphoric relations;
- The ontology coverage, namely the capacity to automatically create more complex ontology relations;
- The semantic representation of the documents content;
- The capability of the dialogue manager to take into account previous interactions and the user models.

At present, some work is already being done trying to solve some kind of coreferences in texts [3], namely using the centering theory to deal with pronominal anaphora.

We are also applying the system to build a dialogue question-answering system to access the University web pages.

References

[1] Eckhard Bick. *The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, 2000.

[2] D. Brickley and R. Guha. *Resource Description Framework (RDF) - Schema Specification*. W3C, 1999.

[3] Caroline Gasperin, Renata Vieira, Rodrigo Goulart, and Paulo Quaresma. Extracting xml syntactic chunks from portuguese corpora. In *TALN'2003 - Workshop on Natural Language Processing of Minority Languages and Small Languages of the Conference on "Traitement Automatique des Langues Naturelles"*, Batz-sur-Mer, France, June 2003.

[4] Jerry Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. Interpretation as abduction. Technical Report SRI Technical Note 499, 333 Ravenswood Ave., Menlo Park, CA 94025, 1990.

[5] H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer, Dordrecht, 1993.

[6] O. Lassila and R. Swick. *Resource Description Framework (RDF) - Model and Syntax Specification*. W3C, 1999.

[7] P. Quaresma and J. G. Lopes. Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. *Journal of Logic Programming*, 54, 1995.

[8] Paulo Quaresma and Irene Pimenta Rodrigues. Using dialogues to access semantic knowledge in a web ir system. In N. Mamede, J. Baptista, and M. Volpe Nunes, editors, *Computational Processing of the Portuguese Language: 6th International Workshop, PROPOR'2003*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2721, pages 201–205. Springer-Verlag, June 2003.

[9] José Saias and Paulo Quaresma. Using nlp techniques to create legal ontologies in a logic programming based web information retrieval system. In J. Breuker, A. Gangemi, D. Tiscornia, and R. Winkels, editors, *Workshop on Legal Ontologies and Web based Legal Information Management of the International Conference on Artificial Intelligence and Law, ICAIL'03*, June 2003.

[10] www.daml.org. *DAML+OIL – DARPA Agent Markup Language*, 2000.