A Logic Programming Based Approach To QA@CLEF05 Track

Paulo Quaresma and Irene Rodrigues

Departamento de Informática, Universidade de Évora, Portugal
{pq,ipr}@di.uevora.pt

Abstract. In this paper the methodology followed to build a question-answering system for the Portuguese language is described. The system modules are built using computational linguistic tools such as: a Portuguese parser based on constraint grammars for the syntactic analysis of the documents sentences and the user questions; a semantic interpreter that rewrites sentences syntactic analysis into discourse representation structures in order to obtain the corpus documents and user questions semantic representation; and finally, a semantic/pragmatic interpreter in order to obtain a knowledge base with facts extracted from the documents using ontologies (general and domain specific) and logic inference. This article includes the system evaluation under the CLEF'05 question and answering track.

1 Introduction

This paper describes some aspects of a dialogue system that has been developed at the Informatics Department of the University of Évora, Portugal. Namely, the system's ability of answering Portuguese questions supported by the information conveyed by collection of documents.

First, the system processes the documents in order to extract the information conveyed by the documents sentences. This task is done by the *information extraction* module.

Then, using the knowledge base built by the first module, the system is able to answer the user queries. This is done by the *query processing* module.

We use models from the computational linguistic theories for the analysis of the sentences from the document collection and queries. The analysis of the sentence includes the following processes: syntactical analysis uses the Portuguese parser *Palavras* [1] using the constraint grammars framework [2] ; semantical analysis interpreter uses discourse representation theory [3] in order to rewrite sentences parser into a Discourse Representation Structure (DRS); and, finally, semantic/pragmatic interpretation uses ontologies and logical inference in the extraction and retrieval modules.

For the documents collection (Publico and Folha de S. Paulo) used in CLEF05 we obtained over 10 million discourse entities that we had to keep in a Database. In order to integrate the logical inference and the external databases we use ISCO[4, 5], a language that extends logic programming.

The QA system, in order to satisfy CLEF requirements, has to answer queries in Portuguese, supported on information conveyed by a given collection of documents. The answer to a specific question is: a set of words and the identification of the document that contained the answer.

For instance, for the following question: "Who was Emiliano Zapata?"

Our system answers:

"Mexican revolutionary 1877-1919 - document: PUBLICO-19940103-32"

At the moment, the system is able to answer:

— Definition questions

"Quem é Joe Striani? – FSP940126-160 norte-americano guitarrista"

— Temporally restricted factoid questions

"Onde é que caiu um meteorito em 1908 – PUBLICO-19951103-46 sibéria"

— Factoid questions

"Onde é a sede da OMC – PUBLICO-19940506-28 genebra"

This system is an evolution of a previous system evaluated at CLEF 2004 [6]. Some of the existing problems were solved, namely, the need to use a pre-processing information retrieval engine to decrease the complexity of the problem. In this CLEF edition, we were able to solve this major scalability problem via the use of ISCO and its power to connect PROLOG and relational databases.

However, the pre-processing of the collection of documents took more time than we expected and we were not able to answer all the questions to the Folha de S. Paulo newspaper. As we will point out in the evaluation section this was our major problem and it is the reason why our results didn't improve from CLEF04 to CLEF05.

In section 2 the architecture of the system is described. In the following sections 3 and 4 the syntactical analysis and the semantical interpretation modules are detailed. The knowledge representation approach is presented in section 5. Section 6 describes the semantic-pragmatic interpretation of the documents. Section 7 presents the module for query processing and answer generation. In section 8 the evaluation results are presented. Finally, in section 9 some conclusions and future work are discussed.

2 System Architecture

The QA system has two operating modes:

Information extraction: the documents in the collection are processed and as a result a knowledge base is created. The phases of information extraction include (figure 1 present this module processes):

- Syntactical analysis: sentences are processed with the Palavras[1] parser. The result of this process is a new collection of documents with the parsing result of each sentence.
- Semantic analysis: the new collection of sentences is rewritten [3] creating a collection documents with the documents semantic representation, where each document has a DRS (structure for the discourse representation), a list of discourse referents and a set of conditions.

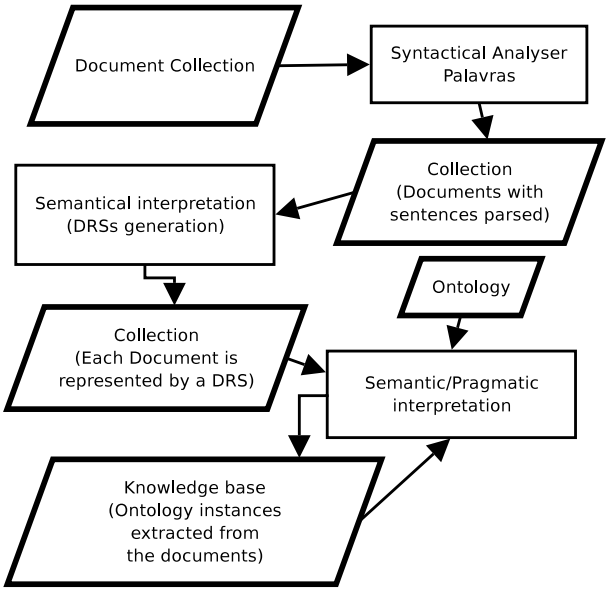

Fig. 1. Document Processing

– Semantic and pragmatic interpretation: the previous collection of documents is processed, using the ontology and, as a result, a knowledge base is built. This knowledge base contains instances of the ontology.

Query processing: this module processes the query and generates the answer, i.e. a set of words and the identification of the document where the answer was found. Figure 2 presents this module diagram. It is composed by the following phases:

– Syntactical analysis: using the parser Palavras[1].
– Semantic analysis: from the parser output, a discourse structure is built, a DRS[3] with the correspondent referents.
– Semantic/Pragmatic interpretation: in this phase some conditions are rewritten taking into account the ontology and generating a new DRS.
– Query Processing: the final query representation is interpreted in the knowledge base through the unification of the discourse entities of the query with documents discourse entities (see section 7).

These processes are described in more detail in the next sections.

3 Syntactical Analysis

Syntactical analysis is done using the PALAVRAS parser from Eckhard Bick[1], This parser gives good morpho-syntactical information and it has a good coverage of the Portuguese language.

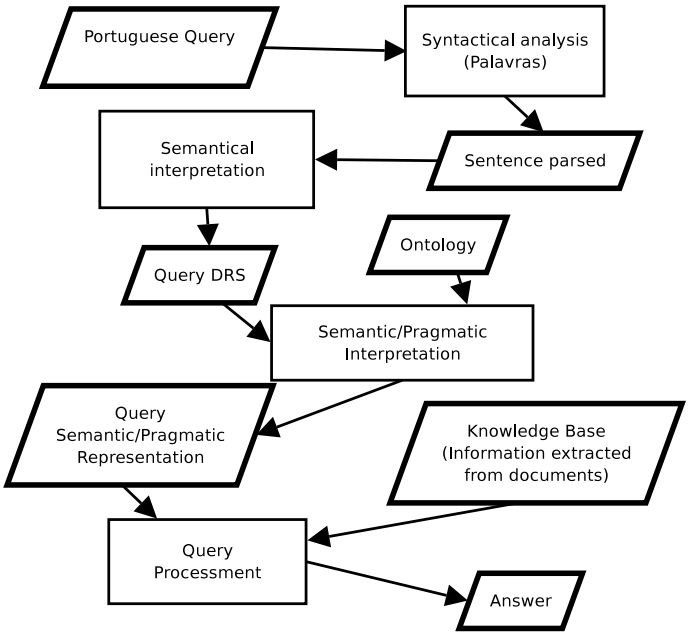

Fig. 2. Query Processing

Below we present an example of the output of Palavras for sentence (3.1):

Um patologista defendeu que Jimi Hendrix morreu de asfixia após ter ingerido álcool e uma dose excessiva de barbitúricos. (3.1)
"A pathologist argued that Jimi Hendrix died of asphyxia after drinking alcoholic beverages and an excessive dose of barbiturics".

The syntactical structure of this sentence is the following:

```
sta(fcl, subj(np, n(art('um', 'M', 'S', <arti> ), 'Um'),
             h(n('patologista','M', 'S', <Hprof> ),'patologista')),
        p(v_fin('defender', 'PS', '3S', 'IND'),'defendeu'),
        acc(fcl, sub( conj_s('que'), 'que')),
        subj(prop('Jimi_Hendrix', 'M/F', 'S'), 'Jimi_Hendrix'),
        p(v_fin('morrer', 'PS', '3S', 'IND'), 'morreu'),
        piv(pp, h(prp('de'),'de'),
            p(np,h(n('asfixia','F','S', <sick> ), 'asfixia'),
              n(pp, h(prp('após'), 'após'),
                      p(icl,p(vp, aux(v_inf('ter'),'ter'),
                            mv(v_pcp('ingerir'), 'ingerido')),
                      acc(n('álcool','M','S',<cm-liq>), 'álcool'),
                      co(conj_c('e'), 'e'),
                      acc(np, n(art('um', 'F','S',<arti>), 'uma'),
                          h(n('dose','F','S'), 'dose'),
```

```
n(adj('excessivo','F','S'), 'excessiva'),
n(pp, h(prp('de'),'de'),
  p(n('barbitúrico',
       'M','P'),'barbitúricos','.')))))))).
```

This structure is represented in Prolog and is used as the input of the semantic analyzer.

4 Semantic Analysis

The semantic analysis rewrites the syntactical structure in to a discourse representation structure [3], DRS. At present, we only deal with sentences as if they were factual, i.e., sentences with existential quantification over the discourse entities. So, our discourse structures are sets of referents, existentially quantified variables, and sets of conditions, predicates linked by the conjunction *and*.

The semantic interpreter rewrites each syntactic tree into a set of discourse referents and a set of conditions integrated in the document DRS. In order to delay the commitment with an interpretation (the attachment) of prepositional phrases, we use the relation *rel* with 3 arguments, the preposition and two discourse entities, to represent the prepositional phrases.

The semantic/pragmatic interpretation of the predicate *rel* will be responsible to infer the adequate connection between the referents. For instance, the sentence 'A viuva do homem'/ 'The widow of the men', is represented by the following DRS:

```
drs(entities:[A:(def,fem,sing),B:(def,male,sing)],
    conditions:[widow(A), men(B), rel(of,A,B)])
```

As it can be seen in the next section, this representation allows the semantic/pragmatic interpretation to rewrite the DRS, obtaining the following structure:

```
drs(entities:[ A:(def, fem, sing), B:(def, male, sing)],
    conditions:[married(A,B), person(A), person(B), dead(B)])
```

In order to show an example of a syntactical tree transformation into a DRS, we show sentence (3.1) rewritten :

```
drs(entities:[A:(indef,male,sing),B:(def,male/fem,sing),
              C:(def,fem,sing),D:(def,male,sing),
              E:(indef,fem,sing)],
    condições:[pathologist(A),argue(A,B),name(B,'Jimmy Hendrix'),
               died(B),rel(of,B,C),asphyxia(C),rel(after,C,D),
               drinking(D), alcohol(D), dose(D), excessive(D),
               rel(of,D,E), barbiturics(E)])
```

User queries are also interpreted and rewritten into DRS. For instance, the question:

"Como morreu Jimi Hendrix?/How did Jimi Hendrix died?" (4.1)

is transformed into the following discourse structure:

```
drs(entities:[F:(def,male/fem,sing),G:interrog(que),male,sing]
    conditions:[died(F), name(F,'Jimmy Hendrix'), rel(of,F,G)])
```

This representation is obtained because "Como/How" is interpreted as "de que/of what". In the semantic-pragmatic interpretation and in the query processing phase, the structure (4.1) might unify with sentence (3.1) and we may obtain the following answer: "Jimi Hendrix died of asphyxia".

5 Ontology and Knowledge Representation

In order to represent the ontology and the extracted ontology instances (individuals and relations between those individuals), we use an extension to logic programming, ISCO[4, 5], which allows Prolog to access databases. This technology is fundamental to our system because we have a very large database of referents: more than 10 millions only for the Público newspaper. Databases are defined in ISCO from ontologies.

The QA system uses two ontologies defined with different purposes:

– an ontology aiming to model common knowledge, such as, geographic information (mainly places), and dates; it defines places (cities, countries, ...) and relations between places.

– an ontology generated automatically from the document collection [7, 8]; this ontology, although being very simple, allows the representation of the documents domain knowledge.

The ontology can be defined directly in ISCO or in OWL (Ontology Web Language) and transformed in ISCO [8].

The knowledge extraction process identifies ontology instances, individuals and relations between individuals, and they are inserted as rows in the adequate database table.

Consider sentence (3.1), with semantic representation in page 5, the information extracted from this sentence would generate several tuples in the database. The information extraction process includes a step where first order logical expressions are *skolemized*, i.e., each variable existentially quantified is replaced by a different identifier:

`(123,''Jimmy Hendrix'')` is added to table *name*

`(123)` is added to table *die*

`(124)` is added to table *asphyxia*

`rel(de,123,124)` is added to table *rel*

In the information extraction process, our system uses the first interpretation of each sentence, without taking into account other possible interpretations of the sentence. This is done to prevent the explosion of the number of interpretation to consider for each document sentence. This way we may miss some sentences

correct interpretation but the QA system performance does not seem to decrease because the document collection content is redundant (many sentences convey the same meaning).

In order to enable the identification of the document sentence that gives rise to a knowledge base fact, we add information in the database linking referents with the documents and sentences were they appeared. For instance the tuple `(123,'publico/publico95/950605/005',4)` is added to table *referred_in*.

6 Semantic/Pragmatic Interpretation

Semantic/pragmatic interpretation process is guided by the search of the best explanation that supports a sentence logical form in a knowledge base built with the ontology description, in ISCO, and with the ontology instances. This strategy for pragmatic interpretation was initially proposed by [9].

This process uses as input a discourse representation structure, DRS, and it interprets it using rules obtained from the knowledge ontology and the information in the database.

The inference in the knowledge base for the semantic/pragmatic interpretation uses abduction and finite domain constraint solvers.

Consider the following sentence:

"X. é a viuva de Y." ("X. is the widow of Y.".)

which, by the semantic analysis, is transformed into the following structure: one DRS, three discourse referents, and a set of conditions:

```
drs(entities:[A:(def,fem,sing),B:(def,fem,sing),C:(def,male,sing)]
    conditions:[name(A, 'X.'), widow(B), rel(of,B,C), is(A,B)])
```

The semantic/pragmatic interpretation process, using information from the ontology, will rewrite the DRS into the following one:

```
drs(entities:[A:(def,fem,sing), C:(def,male,sing)]
    conditions:[person(A,'X.',alive,widow),
                person(C,'Y.',dead,married),married(A,C)])
```

The semantic/pragmatic interpretation as the rules:

```
widow(A):- abduct( person(A,_,alive,widow)).
rel(of,A,B):- person(A,_,_,widow),
              abduct(married(A,B),person(B,_,dead,married)).
```

The interpretation of *rel(of,A,B)* as

married(A,B),person(B,Name,dead,married) is possible because the ontology has a class *person*, which relates persons with their name, their civil state (single, married, divorced, or widow) and with their alive state (dead or alive).

7 Answer Generation

The generation of the answer is done in two steps:

1. Identification of the database referent that unifies with the referent of the interrogative pronoun in the question.
2. Retrieval of the referent properties and generation of the answer.

As an example, consider the following question:
"Quem é a viuva de X.?" ("Who is the widow of X?")
This question is represent by the following DRS, after syntactical and semantical analysis:

```
drs(entities:[A:(who,male/fem,sing), B:(def,fem,sing),
              C:(def,male,sing)],
    conditions:[is(A,B), widow(B), rel(of,B,C), name(C,'X')])
```

The semantic/pragmatic interpretation of this question is done using the ontology of concepts and it allows to obtain the following DRS:

```
drs(entities:[A:(who,fem,sing), C:(def,male,sing),
    conditions:[person(A,_,alive,widow),
                person(C,'X',dead,married), married(A,C)])
```

The first step of the answer generator is:
To keep the referent variables of the question and to try to prove the conditions of the DRS in the knowledge base. If the conditions can be satisfied in the knowledge base, the discourse referents are unified with the identifiers (skolem constants) of the individuals.
The next step is to retrieve the words that constitute the answer:
In this phase we should retrieve the conditions about the identified referent *A* and choose which ones better characterize the entity. Our first option is to choose a condition with an argument *name* (*name(A,Name)* or as in the example *person(_,Name,_,_)*.
However, it is not always so simple to find the adequate answer to a question. See, for instance, the following questions:
What crimes committed X?
How many habitants has Kalininegrado?
What is the nationality of Miss Universe?
Who is Flavio Briatore?
In order to choose the best answer to a question our systems has an algorithm which takes into account the syntactical category of the words that may appear in the answer and it tries to avoid answers with words that appear in the question. Questions about places or dates have a special treatment involving the access to a database of places or dates.
Note that several answers may exist for a specific question. In CLEF05 we decided to calculate all possible answers and to choose the most frequent one.

8 Evaluation

The evaluation of our system was performed in the context of CLEF – Cross Language Evaluation Forum – 2005. In this forum a set (200) of questions is elaborated by a jury and given to the system. The system's answers are, then, evaluated by the same jury.

Our system had the following results:

25% correct answers (50 answers).
1.5% correct but unsupported answers (3 answers).
11% inexact answers – too many (or too few) words (22 answers).
62.5% wrong answers (125 answers).
The answer-string "NIL" was returned 117 times.
The answer-string "NIL" was correctly returned 12 times.
The answer-strings "NIL" in the reference are 18

The system had 125 wrong answers, but it is important to point out that 105 of these wrong answers were NIL answers, i.e., situations were the system was not able to find any answer to the questions. So, only in 10% of the situations (20 answers) our system gave a really wrong answer.

The major problem with the remaining 105 no-answers is the fact that, due to time constraints we were not able to process the collection of documents from the Folha de S. Paulo newspaper. At present, we do not know how many of these no-answers would be answered by this collection, but we expect our results to improve significantly.

A preliminary analysis of the other incorrect answers showed that the main cause of problems in our system is related with lack of knowledge: wrong syntactical analysis; lack of synonyms; and, mostly, an incomplete ontology. In fact, most problems are related with incorrect pragmatic analysis due to an incomplete ontology.

However, and taking into account that our answers were computed using only one collection of documents (the Público newspaper), and comparing with the CLEF2004 results, we believe our system produced good and promising results. In fact, it showed to have a quite good precision on the non NIL answers: only 10% of these answers were wrong.

9 Conclusions and Future Work

We propose a system for answering questions supported by the knowledge conveyed by a collection of document. The system architecture uses two separate modules: one for knowledge extraction and another one for question answering.

Our system modules uses natural language processing techniques, supported by well known linguistic theories, to analyze the documents and query sentences in every processing phases: syntactic, semantic and pragmatic analysis.

The process of knowledge extraction is defined using logic a programming framework, ISCO, that integrates: representation and inference with ontologies, and the access to external databases to add and retrieve ontology instances. The process of query answering is defined in the same logic programming framework.

The system main source of problems are:

First, poor coverage of the defined ontology: the lack of knowledge in the ontology prevents us of relating query conditions with the document sentences conditions; this explains why system gets 117 NIL answers when it only should get 12.

Next, errors in NLP tools. The PALAVRAS has some troubles in parsing some document and query sentences. The errors in the parsing of a query is more problematic than the problems in the analysis of the document sentences. We hope in the near future to solve the problems in parsing the user queries. The semantic interpretation module, developed by us, also has some problems in rewriting some parser trees. These problems also appear in the processing of the user queries.

As future work, we intend to improve our ontology and our linguistic resources, namely the use of a general domain thesaurus. The improvement of some NLP tools is another area needing much work.

References

1. Bick, E.: The Parsing System "Palavras". Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus University Press (2000)
2. Karlsson, F.: Constraint grammar as a framework for parsing running text. In Karlgren, H., ed.: Papers presented to the 13th International Conference on Computational Linguistics. Volume 1753., Helsinki, Finland, Springer-Verlag (1990) 168–173
3. Kamp, H., Reyle, U.: From Discourse to Logic:An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Dordrecht: D. Reidel (1993)
4. Abreu, S.: Isco: A practical language for heterogeneous information system construction. In: Proceedings of INAP'01, Tokyo, Japan, INAP (2001)
5. Abreu, S., Quaresma, P., Quintano, L., Rodrigues, I.: A dialogue manager for accessing databases. In: 13th European-Japanese Conference on Information Modelling and Knowledge Bases, Kitakyushu, Japan, Kyushu Institute of Technology (2003) 213–224 To be published by IOS Press.
6. Quaresma, P., Rodrigues, I.: Using dialogues to access semantic knowledge in a web legal IR system. In Moens, M.F., ed.: Procs. of the Workshop on Question Answering for Interrogating Legal Documents of JURIX'03 – The 16th Annual Conference on Legal Knowledge and Information Systems, Utrecht, Netherlands, Utrecht University (2003)
7. Saias, J., Quaresma, P.: Using nlp techniques to create legal ontologies in a logic programming based web information retrieval system. In: Workshop on Legal Ontologies and Web based legal information management of the 9th International Conference on Artificial Intelligence and Law, Edinburgh, Scotland (2003)
8. Saias, J.: Uma metodologia para a construção automática de ontologias e a sua aplicação em sistemas de recuperação de informação – a methodology for the automatic creation of ontologies and its application in information retrieval systems. Master's thesis, University of Évora, Portugal (2003) In Portuguese.
9. Hobbs, J., Stickel, M., Appelt, D., Martin, P.: Interpretation as abduction. Technical Report SRI Technical Note 499, 333 Ravenswood Ave., Menlo Park, CA 94025 (1990)