QA@L^{2}F, second steps at QA@CLEF

Luísa Coheur, Ana Mendes, João Guimarães
Nuno J. Mamede, Ricardo Ribeiro
L^{2}F/INESC-ID Lisboa
Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
qa-clef@l2f.inesc-id.pt

Abstract

This paper describes the participation of QA@L^{2}F, the question-answering system from L^{2}F/INESC-ID, at the QA track of CLEF in 2008.

Making intensive use of a Natural Language Processing chain (which includes, among others, a morphological analyzer, a disambiguation module, a multi-word recognizer, a chunker and a named entities recognizer), QA@L^{2}F is based on a three module approach to answer questions: corpora pre-processing (where the information sources are processed and potentially relevant information is extracted), question interpretation (where the question is converted into a frame) and answer extraction (where different strategies are used to retrieve the final answer to the input question).

QA@L^{2}F system was created in 2007 and had its first participation at CLEF07, with results we considered auspicious. Nevertheless, with the objectives of correcting some detected failures, increasing the percentage of questions the system deals with and correctly answers, and also experiment new techniques using the same processing tools, the system suffered modifications during this year: the question interpretation step was improved to better profit from the results of the Natural Language Processing chain; an anaphora solver module was introduced, which allowed us to answer some questions containing backwards references; finally, some other small improvements were done on the system, especially in the answer extraction module.

QA@L^{2}F had 20% of precision at the competition this year, which represents an increase in the number of correct answers returned by the system of 6%, as compared to the last year results. The system highest accuracy values are on definition questions, in which it achieved 60.714% of precision. However, much work is still to be done in order to improve the system's results, like, for instance, the introduction of an answer validation module, in order to minimize the number of answers given with different type from the expected type, which was the case this year with 10 of our wrong answers.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [**Database Managment**]: Languages—*Query Languages*

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Question Interpretation, Anaphora Resolution

1 Introduction

In this paper we present QA@L^{2}F, the question-answering (QA) system developed at L^{2}F/INESC-ID, as well as the results it obtained at CLEF 2008.

QA systems aim at returning the exact answer for a question formulated in natural language from a (usually) very large amount of text collections. While some QA systems are said to be domain-specific, as they are focused on particular information that concerns a specific topic (like WEBCOOP [3] for the tourism domain), open-domain QA systems, like Priberam's QA system [2] and Senso [10], both for Portuguese, deal with general questions. These two heavily relly in documents processing. However, some systems employ other resources, like Esfinge [4], again for Portuguese, which uses data available on the Web. Moreover, Esfinge makes use of a named entities recognizer, like RAPOSA [11] does in both question analysis and snippet searching, taking in consideration the benefits of using named entities in QA, like it has been proven by several experiments on different languages [12, 7].

QA@L^{2}F is an open-domain QA system for the Portuguese language, created in the year of 2007, that also uses a named entities recognizer in its processing. This paper focus the main differences in the system since last year [6]. The rest of the paper is organized as follows: section 2 describes the most recent version of the QA@L^{2}F system, presenting the modifications introduced in it since 2007; section 3 shows, discusses and compares the evaluation results; finally, section 4 concludes and points to future work.

2 QA@L^{2}F

QA@L^{2}F system makes use of a three step approach and deeply rellies in L^{2}F's Natural Language Processing (NLP) chain. A short description of QA@L^{2}F's steps are depicted as follows:

- Corpus Pre-processing: the available information sources were partially processed in order to extract potentially relevant information (specifically, named entities and relations between concepts). The resulting database – created last year – is used by QA@L^{2}F; however, since it is based on last year's named entities recognizer, poorer than the one we possess nowadays, an information extraction step is also executed online, using the current named entities recognizer and up to date linguistic patterns;

- Question Interpretation: the question is interpreted and transformed into a frame, which can be mapped into an SQL query or used to search relevant snippets;

- Answer Extraction: according with the question type, different strategies are used in order to find the answer.

This section continues by describing the modifications in the system since CLEF 2007, including a different approach for the question interpretation step and a new anaphora solver.

2.1 Question Interpretation

The current question interpretation step differs from last year's process. It is now partly independent from the named entities recognition, namely the identification of the question type and target, as well as the main verb, adjectives and some adverbs. This decision was due to the fact that the named entities recognizer we use is still under development and based on several layers of rules, applied in a pipeline.

In last year's system, in order to profit from the most recent version of the named entities recognizer (XIP [1]) [1], we used several rules in its last layer to extract other relevant information from the question, like its type and target. However, the triggers to these rules are highly dependent on (the continuous) modifications in XIP's previous layers. Being so, we decided to perform

[1] As it will be explained later on, XIP is not only responsible for the recognition of named entities; however, for the sake of simplicity, we decided to introduce it here as a named entities recognizer.

this step independently: the named entities recognizer is used only to retrieve the named entities in the question, while the other important features (e.g., type and target) are collected using other tools in the NLP chain.

As a consequence of this change, in this new version of QA@L^{2}F we can get the most from the newer version of the named entities recognizer and put our efforts on a stable extractor for the rest of the relevant information.

2.1.1 Question Interpretation process

The question interpretation process involves several tools, described in the following:

- Palavroso [5], which performs a morphological analysis;
- MARv [9], which disambiguates the result of the morphological analyser;
- RuRriCo (an improved version of PAsMo [8]), which is a ruled based tool, that recognizes multi-word terms, collapses them into single tokens and can also split tokens;
- XIP, which returns the input organized into chunks, connected by dependency relations, and also identifies and classifies the named entities in the input.

Figure 1 illustrates the entire question interpretation process used in QA@L^{2}F.

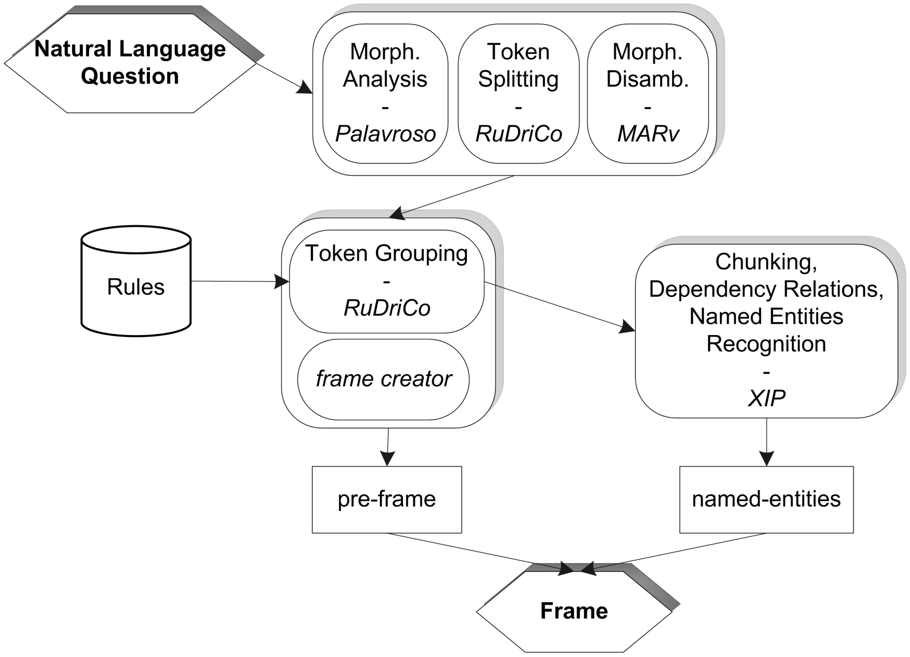
Natural Language Question
Morph. Analysis - Palavroso
Token Splitting - RuDriCo
Morph. Disamb. - MARv
Rules
Token Grouping - RuDriCo
frame creator
Chunking, Dependency Relations, Named Entities Recognition - XIP
pre-frame
named-entities
Frame

Figure 1: Question interpretation in QA@L^{2}F.

2.1.2 Rules

From Figure 1, it can be noticed that RuDriCo plays a central role in QA@L^{2}F's question interpretation. In fact, this tool is responsible for identifying the question type and target, as well as other relevant elements, like the question subtype and important verbs. The following example, shows a RuDriCo rule.

```
S1 ['onde','CAT'/C1]
S2 ['ser','CAT'/C2]? 'que' [L3,'CAT'/C3]?
S4 [L4,'CAT'/'nascer']
S5 ['o','CAT'/'art']?
S6 [L6,'CAT'/'noun']
S7 [L7,'HMM'/'true']*
S10 ['?','CAT'/C10] -->
   S1 ['onde', 'CAT'/C1, 'type'/'onde_verb']
   S4 [L2,'CAT'/'verb']
   S6@+S7@* [L6@+L7@*, 'type'/'target'].
```

Rules are patterns that match against labeled text. The previous RuDriCo rule means that:

- if there is a sequence in the question having:
  - a word that has "onde" (*where*) as lemma, captured by S1;
  - a conjugation of the verb "ser" (*to be*) and a "que" (*that*), both optional – marked with the question mark "?"; [2]
  - a verb "nascer" (*to be born*), captured by S4;
  - a noun ("noun"), captured by L6, and then a sequence that ends at the question mark, captured by S7;
- then it is created a sequence with:
  - the same first word (S1), lemma "onde", same category and type "onde_verb";
  - the same main verb (S4);
  - L6 plus S7, which is tagged as "target".

For instance, considering the question "Onde nasceu a Florbela Espanca?" (*Where was Florbela Espanca born?*), and after being processed by Palavroso, Marv and RuDriCo, by using the previous described rule, RuDriCo returns the following output:

```
<sentence>
  <word name="Onde">
    <class root="onde">
      <id atrib="CAT" value="adv"/>
      <id atrib="type" value="onde_verb"/>
    </class>
  </word>
  <word name="nasceu">
    <class root="nascer">
      <id atrib="CAT" value="verb"/>
    </class>
  </word>
  <word name="Florbela Espanca">
    <class root="Florbela espancar">
      <id atrib="type" value="target"/>
    </class>
  </word>
</sentence>
```

In parallel, XIP, the responsible tool for named entities recognition, identifies the named entities in the question.

[2] In Portuguese, some questions can start either by an interrogative pronoun, like "Onde nasceu a Florbela Espanca?" (literally *Where was born Florbela Espanca?*), or by an interrogative pronoun followed by the inflected verb to be and the pronoun that, like "Onde é que nasceu a Florbela Espanca?" (literally *Where is that was born Florbela Espanca?*). In such cases, both constructions are possible and have the same meaning.

2.1.3 Frames

During the question interpretation a frame is created. This frame is then mapped into SQL or used to extract relevant snippets from the database. Each frame consists in the following elements:

- the name of the script that should be called, regarding the question type;
- the question target;
- a set of entities identified by the named entities recognizer;
- a set of auxiliar (and optional) elements from the question such as:
  - the target-type;
  - main verbs;
  - adjectives;
  - adverbs.

After the processing done by RuDriCo, the frame creator builds a pre-frame. Using the tags added by RuDriCo, the frame creator outputs the script to be called, the question target and so on. For instance, considering the previous example, the script `script-wiki-target.pl` is identified by the label `onde_verb` and the identified target is `Florbela Espanca`.

As the named entities recognizer identifies "Florbela Espanca" as a `PERSON`, the merge of the two inputs results in:

```
<frame>
  <script name="script-wiki-target.pl"/>
  <target value="Florbela Espanca"/>
  <entidades>
    <entidade type="PEOPLE" value="Florbela Espanca"/>
  </entidades>
  <auxiliares>
    <auxiliar type="verb" value="nasceu"/>
  </auxiliares>
</frame>
```

After a XML transformation the final frame is created for the input question "Onde nasceu a Florbela Espanca?":

```
where/script-wiki-target.pl
target="florbela espanca"
entities people="florbela espanca"
auxiliares verb="nasceu"
```

The obtained script is then called and uses its arguments either to build the SQL query or to obtain the snippets that may contain the answer.

2.2 Anaphora resolution

Current QA@L^{2}F system integrates an anaphora resolution module that addresses:

1. ellipsis, in which the question starts by a conjunction and is followed by a noun;
2. pronouns;
3. ellipsis, in which the question consists either by a single interrogative pronoun/adverb or by an interrogative pronoun/adverb followed by verbs like *to be* or *stay*;

4. ellipsis, in which the question consists either by an interrogative pronoun/adverb followed either by a noun or by a noun and a verb or only by a verb like *die*;

5. all the situations envolving more than one noun or verbs not belonging to the previous mentioned sets of verbs.

Consider again the (reference) question "Onde nasceu a Florbela Espanca?" (*Where was Florbela Espanca born?*). Next examples illustrate the presented situations, respectively:

1. E Saramago? (*And Saramago?*)
2. Quem era ela? (*Who was she?*)
3. Quando? (*When?*)
4. Onde morreu? (*Where did (she) die?*) [3]
5. Quantos poemas escreveu? (*How many poems did (she) write?*) [3]

In order to implement this, we made the choice to manipulate the obtained frames and not the surface question itself. It should also be mentioned that all the anaphoras were solved having the first question of a group of questions as the reference, which is not always the case.

The following example illustrates the anaphora solving process. Let us start by considering the previous sentences as the unique input of QA@L^{2}F. The following frames are obtained:

1. E Saramago? (*And Saramago?*)
   ```
   target="saramago"
   entities people="saramago"
   ```

2. Quem era ela? (*Who was she?*)
   ```
   who/script-who-people.pl
   ```

3. Quando? (*When?*)
   ```
   when/script-when.pl
   ```

4. Onde morreu? (*Where did (she) die?*)
   ```
   where/script-where.pl
   auxiliars verb="morreu"
   ```

5. Quantos poemas escreveu? (*How many poems did (she) write?*)
   ```
   howM/script-wiki-target.pl
   auxiliars target-type="poemas" verb="escreveu"
   ```

The folowing frame results from the reference question *Where was Florbela Espanca born?*:

Onde nasceu a Florbela Espanca?
```
where/script-wiki-target.pl
target="florbela espanca"
entities people="florbela espanca"
auxiliares verb="nasceu"
```

Being so, the following replacements are made (respectively to each identified situation):

1. the script called and the auxiliars from the first sentence frame are added to the anaphoric frame;

2. the target from the first sentence frame is added to the anaphoric frame;

[3] In Portuguese, the pronoun is optional.

3. the target, entities and auxiliars from the first sentence frame are added to the anaphoric frame;

4. the target from the first sentence frame as well as its entities are added to the frame. Auxiliars are also added as long as they don't have the same type of an auxiliar from the obtained frame (for instance *verb*);

5. the target from the first sentence frame is added to the anaphoric frame.

As a result of the mentioned process, we obtain the following as final frames:

1. E Saramago? (*And Saramago?*)
   ```
   where/script-wiki-target.pl
   target="saramago"
   entities people="saramago" verb="nasceu"
   ```

2. Quem era ela? (*Who was she?*)
   ```
   who/script-who-people.pl
   target="florbela espanca"
   ```

3. Quando? (*When?*)
   ```
   when/script-wiki-target.pl
   target="florbela espanca"
   entities people="florbela espanca"
   auxiliares verb="nasceu"
   ```

4. Onde morreu? (*Where did (she) die?*)
   ```
   where/script-wiki-target.pl
   target="florbela espanca"
   auxiliars verb="morreu"
   ```

5. Quantos poemas escreveu? (*How many poems did (she) write?*)
   ```
   howM/script-wiki-target.pl
   target="florbela espanca"
   entities people="florbela espanca"
   auxiliars target-type="poemas" verb="escreveu"
   ```

Using this anaphora solver, that still needs strong improvements, we were able to successfully generate the correct frame for 13 of the 52 anaphoric situations. In fact, 4 of these 13 frames were incorrect due to errors occurred in the generation of the reference frame; however, since these errors were not directly due to the anaphora solver, we considered those results as being correct. This means that we were able to generate the correct frames for 25% of the anaphora's situations.

2.3 Other improvements

In the answer extraction step, we introduced a method for retrieving answers based on the words proximity in the text, that works similarly in both Wikipedia and newspaper corpora. For every relevant passage in the corpus, snippets are searched that contain the auxiliar concepts which are also in the question, such as verbs or adjectives. All the named entities that match the expected answer type are extracted by using the NLP chain. Afterwards, the distances between the auxiliar concepts and the extracted named entities are measured [4], and the named entity of the expected type with smallest distance to an auxiliar concept is retrieved as the final answer.

Consider, for instance, the question "Quantos jogadores tem uma equipe de Basquete?" (*How many players has a basketball team?*). It has as target "Basquete" (*basketball*) and as auxiliar

[4] If two words appear together in a text, the distance between them is "1"; if two words have a one word in-between, the distance is "2"; and so on...

word "jogadores" (*players*). The implemented method allows the answer extraction step to return "5" as final answer, based on the sentence "É jogado por dois times de 5 jogadores, que têm como objectivo..." (*It's played by two teams of 5 players, that have as goal...*). In this case, the distance between the named entity "5" and the auxiliar word "jogadores" is the smallest possible.

Also, we developed several linguistic patterns in order to detect and create more complex dependencies between concepts. These were introduced in the NLP chain that supports our system. As an example, see the sentence "Nascido em Coimbra, em 10 de Novembro de 1913, Álvaro Cunhal..." (*Born in Coimbra, on the 10th of November 1913, Álvaro Cunhal...*). Patterns were created to gather the non-ambiguous information that Álvaro Cunhal was born in Coimbra (represented internally by the dependency `LOCATION_OK(Álvaro Cunhal, Coimbra, Nascido)`) and on the 10th of November 1913 (represented internally by the dependency `DATE_OK(Álvaro Cunhal, 10 de Novembro de 1913, Nascido)`).

Finally, based on the fact that Wikipedia's text is presented in a semi-structured way, we created a module to extract information directly from Wikipedia's tables. This can be used, for instance, to faster retrieve answers to questions like "What is the capital of ... ?" or "What is the language spoken in ...?" as they can be easily found in Wikipedia and there is no need to perform any kind of time-consuming processing.

3 Evaluation

QA@L^{2}F was evaluated at CLEF, using Portuguese as source and target language. Table 1 shows the obtained results. The system had better overall results this year: 20% of correct answers, *versus* 14% last year. However, the number of wrong answers continues high (150), even if it has decreased from 166 since 2007.

Right	Wrong	ineXact	Unsupported	Accuracy over the FIRST answer (%)
40	150	5	5	40/200 = 20%

Table 1: QA@L^{2}F results at CLEF 2008.

Table 2 shows the detailed results for each question type. Just like what happened at the competition in 2007, the system obtained this year the best results in the definition questions. Also, the accuracy in factoids questions improved: we had 22 factoid questions answered correctly (corresponding to 13.580% of precision), *versus* 8 (5.03%) last year. Moreover, the system answered right to one list question: last year no correct answers were given to any question of this type.

Question Type	Total	Right	Wrong	ineXact	Unsupported	Accuracy (%)
Factoids	162	22	132	3	5	22/162 = 13.580%
Lists	10	1	8	1	0	1/10 = 10.0%
Definition	28	17	10	1	0	17/28 = 60.714%
Temporally Restricted	16	1	14	0	1	1/16 = 6.250%

Table 2: QA@L^{2}F results for each question type.

One thing to be mentioned is that we did not profit from the fact that the system could return 3 answers. In fact we only presented 230 answers: 184 single answers, 2 double answers and 14 triple answers.

Finally, we would like to mention that several answers were extracted from Wikipedia's tables and, although the page from where they were extracted was correctly identified, they were considered unsupported.

4 Conclusions and future work

Although the entire system needs strong improvements, we believe that there are many small things to be done in QA@L^{2}F that can make it achieve better results, such as:

- the answer type should be validated. This year, 10 out of the 150 wrong questions do not have the expected type from the question. Being so, if we have a tool that is able to say that that something is a `PERSON` or a `LOCATION` (for instance), it will not be difficult, if one is expecting a `PERSON` or a `LOCATION`, to validate it. This will certainly give better results, when articulated with redundancy, than using redundancy by itself. We are aware that this will certainly lead to a hierarchy of named entities types;
- taking advantage of the possibility of returning 3 answers to each question;
- improve the anaphora solver. For instance, the system only solves anaphoras based on the frame constructed for the first question of a group of related questions (the reference question, in bold in the example).

  Onde nasceu Florbela Espanca? *Where was Florbela Espanca born?*

  Onde é que ela morreu? *Where did she died?*

  Quando? *When?*

  In such cases, the anaphora solver presents an undiserable behaviour: on the third question, instead of searching "When did Florbela Espanca die?", the system will try to find the date when Florbela Espanca was born. This and other improvements should be done in our anaphora solver.

References

[1] Salah Aït-Mokhtar, Jean-Pierre Chanod, and Claude Roux. A Multi-Input Dependency Parser. In *Proceedings of the Seventh IWPT (International Workshop on Parsing Technologies)*, Beijing, China, October 2001.

[2] Carlos Amaral, Adán Cassan, Helena Figueira, Andr Martins, Afonso Mendes, Pedro Mendes, Cludia Pinto, and Daniel Vidal. Priberam's question answering system in qa@clef 2007. *Working Notes for the CLEF 2007 Workshop*, 2007.

[3] F Benamara. Cooperative question answering in restricted domains: the WEBCOOP experiment. *ACL 2004 Workshop on Question Answering in Restricted Domains*, 2004.

[4] Luís Fernando Costa. Esfinge a question answering system in the web using the web. In *EACL*. The Association for Computer Linguistics, 2006.

[5] Jos Carlos Medeiros. Anlise Morfolgica e Correco Ortogrfica do Portugus. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, Portugal, 1995.

[6] Ana Mendes, Lusa Coheur, Nuno J. Mamede, Ricardo Ribeiro, Fernando Batista, and David Martins de Matos. QA@L2F, first steps at QA@CLEF. In *CLEF 2007 Proceedings (to appear)*, Lecture Notes in Computer Science. Springer, 2008.

[7] Diego Mollá, Menno van Zaanen, and Luiz A.S. Pizzato. Named entity recognition for question answering. In *Proceedings ALTW 2006*, pages 51–58, 2006.

[8] Joana Paulo Pardal and Nuno J. Mamede. Terms Spotting with Linguistics and Statistics. In *Proceedings of the international workshop "Taller de Herramientas y Recursos Lingusticos para el Espanl y el Portugus", IX Iberoamerican Conference on Artificial Intelligence (IBERAMIA 2004)*, pages 298–304, November 2004.

[9] Ricardo Ribeiro, Nuno J. Mamede, and Isabel Trancoso. Using Morphossyntactic Information in TTS Systems: comparing strategies for European Portuguese. In *Computational Processing of the Portuguese Language: 6th International Workshop, PROPOR 2003, Faro, Portugal, June 26-27, 2003. Proceedings*, volume 2721 of *Lecture Notes in Computer Science*. Springer, 2003.

[10] Jos Saias and Paulo Quaresma. The Senso Question Answering Approach to Portuguese QA@CLEF-2007. *Working Notes for the CLEF 2007 Workshop*, 2007.

[11] Lus Sarmento and Eugnio Oliveira. Making RAPOSA (FOX) smarter. *Working Notes for the CLEF 2007 Workshop*, 2007.

[12] Antonio Toral, Elisa Noguera, Fernando Llopis, and Rafael Muñoz. Improving question answering using named entity recognition. In *Proceedings of the 10th NLDB congress*, Lecture notes in Computer Science, Alicante, Spain, 2005. Springer-Verlag.