Answering Portuguese Questions

Luís Fernando Costa and Luís Miguel Cabral

Linguateca, Oslo Node, SINTEF ICT, Norway
{Luis.Costa, Luis.M.Cabral}@sintef.no

Abstract. Esfinge is a general domain Portuguese question answering system that participated in the last four editions of CLEF. This system uses the Web as a fundamental resource in its architecture, using information redundancy rather than sophisticated annotations of the document collections to retrieve answers. In this paper we describe experiments that took as starting point the version of Esfinge that participated at the evaluation contest CLEF 2007. These experiments consisted in using different types of search patterns to retrieve relevant documents for questions, as this issue (document retrieval) was responsible for most of the errors occurred at CLEF 2007.

Keywords: Question answering, Portuguese, question reformulation

1 Architecture of Esfinge

In this paper we will give a short description of the Portuguese question answering system Esfinge [1], as well as of a set of experiments performed with this system using different types of search patterns to retrieve relevant documents to answer questions.

The architecture of Esfinge is composed by a pipeline of modules that handles each question in order to provide one answer.

The questions are initially fed to an *Anaphor Resolution* module which caters for the resolution of anaphors. This module adds, to the original question, a list of alternative questions where the anaphors are (hopefully) resolved.

Then, Esfinge iterates over the set of alternative questions created in the previous module:

- *The Question Reformulation* module transforms the question into patterns of plausible answers. This is done using two different approaches: a) using a set of pre-defined pattern pairs that associate patterns of questions with patterns of plausible answers, producing a set of pairs (answer pattern, score) or b) using PALAVRAS [2] analysis to identify the main verb, its arguments and adjuncts and some entities from previous topic questions which are used to create search patterns.
- The *Search Document Collections* module then uses these patterns to search in document collections. If no documents are retrieved, execution stops and NIL is returned meaning that the system is not able to answer the question.

- Otherwise it is possible to proceed by searching the same patterns in the Web using Google's and Yahoo's search APIs (this is optional).
- Then, all text passages retrieved by the previous modules are analysed by the named entity recognition system SIEMES [3] and an n-grams module in order to obtain candidate answers, ranking then according to their frequency, length and the score of the passage from where they were retrieved (these parameters are multiplied in order to define the score of each candidate answer).
- This ranking is in turn adjusted using the BACO database of co-occurrences [4].
- Then, the candidate answers (by ranking order) are analysed to check if they pass a set of filters (these filters are used to exclude answers that are contained in the questions, very frequent words and answers where the constituent words have an unlikely sequence of PoS[1]). Answers are also checked where it regards to the existence of documents in the collections supporting them.
- From the moment that Esfinge finds a possible answer, it will only check candidates that include that answer in order to find more complete answers.

After iterating over all alternative questions, Esfinge has a set of possible answers. That is when the module *Answer Selection* comes to play. This module aims to select the best answer to the given question, which will be the final answer to be returned.

2 Experimental setup

The error analysis in [1] pointed out several causes for the wrong answers provided by Esfinge. These included among others: wrong or incomplete search patterns, document retrieval failure, missing patterns to identify the type of answer (type of named entities) and problems with the search in Wikipedia.

Our initial work evolved around adding more patterns to identify answers which are named-entities and updating the existing ones based on the results of the aforementioned error analysis. Additionally the Wikipedia collection was re-indexed for not allowing searches on words shorter than 3 characters and for lacking the last sentence in some cases.

In the baseline results in this paper, Esfinge uses therefore an updated answer type identification functionality and a new Wikipedia index. Additionally we did the following experiments:

More complete search patterns. According to the error analysis in [1], wrong or incomplete search patterns were the main cause for wrong answers (63 of the 165 wrong answers). We found that out of this 63, in 41 of them the problem was that the interrogative noun phrase had not been catered for in the created search patterns. This meant that important words were being left out, which frequently led to the retrieval of not relevant text passages. For example for the question *Que país declarou a independência em 1291?, the* word *país* was not included in the search patterns.

We adapted the part of the *Question Reformulation* module that uses PALAVRAS analysis to create search patterns in order to include the complete noun phrases.

[1] Jspell (http://search.cpan.org/dist/Lingua-Jspell/) was used for this purpose.

Search patterns without verbs. Document retrieval failure was the second more frequent cause for wrong answers (33 out of the 165 wrong answers).

The search patterns created by Esfinge use the words as they appear in the questions, but observing the solutions of the questions in QA@CLEF 2007 one can realize that sometimes in the supporting snippets some of the words in the question do not appear (synonyms appear instead of them, verbs appear in different tenses, etc.).

Since Esfinge does not use annotated document collections or dictionaries, we decided to experiment what could be achieved by not including verbs in the search patterns used to retrieve relevant passages. For that purpose we created an option in the *Question Reformulation* module to create search patterns without verbs. These patterns were used when no answer could be retrieved with the complete patterns.

Combining two types of search patterns. As described in section 1, Esfinge uses two different techniques to create search patterns to retrieve relevant passages: a) Using a pattern file that associates patterns of questions with patterns of plausible answers; b) Creating patterns using PALAVRAS analysis to identify the main verb, its arguments and adjuncts.

[1] reports experiments where sets of answers obtained using different information sources were combined/merged. The results of these experiments were worse than some of the original sets of answers. In this paper we decided to test a different answer combination approach, namely what could be achieved combining the two types of search patterns.

3 Evaluation and discussion of the results

The questions used to test the system were the 200 questions used at QA@CLEF 2007 for the PT-PT track (questions and answers in Portuguese) [5]. We are aware that it is questionable to use the same set of questions in the error analysis and in a subsequent evaluation, but creating a new set of questions is a very time-consuming task. However our experiments are not tailored to this particular set of questions, instead they try to address general problems detected in the error analysis.

Table 1 presents the results obtained in the experiments described in the previous section. The line "CLEF 2007" refers to the results of the best run described in [1].

Table 1. Results of the experiments (F: Factoid questions; D: Definition questions)

Description	Right Answers				Unsupported Answers	Inexact Answers (missing words)	Inexact Answers (too many words)	Good supporting snippets
	All	NIL	F	D				
CLEF 2007	35	5	28	7	1	6	1	59
Baseline	34	5	29	5	4	7	1	58
More Complete Search Patterns	35	7	31	4	4	7	1	60
Without verbs	41	4	37	4	7	7	1	71
Combination	44	3	39	5	8	6	1	76

Table 2 gives an overview of the main causes for errors in the experiment with the best results.

Table 2. Causes for wrong answers in the best run

Cause	CLEF 2007	Combination
Co-reference resolution	25	23
Wrong or incomplete search patterns	63	15
Document retrieval failure	33	12
Answer scoring algorithm	24	60
Answer support testing	7	27
Other	6	19
Total	165	156

The best results appeared in the run which combined two different types of search patterns which not surprisingly also had a lower number of correct NIL answers. It is also worth to note that the improvements were obtained only in the factoid questions.

Nevertheless, the most significant result of our evaluation was obtained in the error analysis performed for the best run: even though the final results were not strikingly better, Table 2 shows that we managed to move the errors to a later stage in the system execution. Whereas at CLEF 2007, most of the errors were due to wrong and incomplete search patterns and document retrieval failure, in the combination experiment described in this paper most of the errors occurred in the answer scoring algorithm and in testing whether an answer is supported by a text snippet.

Acknowledgments. This work was done in the scope of the Linguateca project, jointly funded by the Portuguese Government and the European Union (FEDER and FSE) under contract ref. POSC/339/1.3/C/NAC. We would also like to thank Diana Santos for the work related to the use of PALAVRAS.

References

1. Cabral, L.M., Costa, L.F., Santos, D.: What happened to Esfinge in 2007?. In 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, Revised Selected Papers (2008), Berlin: Springer. LNCS.
2. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus: Aarhus University Press (2000)
3. Sarmento, L.: SIEMÊS - a named entity recognizer for Portuguese relying on similarity rules. In 7th Workshop on Computational Processing of Written and Spoken Language (PROPOR'2006) (Itatiaia, RJ, Brasil, 13-17 May 2006), Springer, pp. 90-99.
4. Sarmento, L.: BACO - A large database of text and co-occurrences. In Proceedings of LREC'2006 (Genoa, Italy, 22-28 May 2006), pp. 1787-1790.
5. Giampiccolo, D., Forner, P., Peñas, A., Ayache, C., Cristea, D., Jijkoun, V., Osenova, P., Rocha, P., Sacaleanu, B., Sutcliffe, R.: Overview of the CLEF 2007 Multilingual Question Answering Track. In 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, Revised Selected Papers. Berlin: Springer, LNCS, 2008.