Esfinge – a Question Answering System in the Web using the Web

Luís Fernando Costa
Linguateca at SINTEF ICT
Pb 124 Blindern,
0314 Oslo, Norway
luis.costa@sintef.no

Abstract

Esfinge is a general domain Portuguese question answering system. It tries to take advantage of the great amount of information existent in the World Wide Web. Since Portuguese is one of the most used languages in the web and the web itself is a constantly growing source of updated information, this kind of techniques are quite interesting and promising.

1 Introduction

There are some question answering systems for Portuguese like the ones developed by the University of Évora (Quaresma and Rodrigues, 2005) and Priberam (Amaral et al, 2005), but these systems rely heavily on the pre-processing of document collections. Esfinge explores a different approach: instead of investing in pre-processing corpora, it tries to use the redundancy existent in the web to find its answers. In addition it has an interface on the web where everyone can pose questions to the system (http://www.linguateca.pt/Esfinge/).

Esfinge is based on the architecture proposed in (Brill, 2003). Brill suggests that it is possible to obtain interesting results, applying simple techniques to large quantities of data. The Portuguese web can be an interesting resource for such architecture. Nuno Cardoso (p.c.) is compiling a collection of pages from the Portuguese web and this collection will amount to 8.000.000 pages. Using the techniques described in (Aires and Santos, 2002) one can estimate that Google and Altavista index 34,900,000 and 60,500,000 pages in Portuguese respectively.

The system is described in detail in (Costa, 2005a, 2005b).

2 System Architecture

The inputs to the system are questions in natural language. Esfinge begins by transforming these questions into patterns of plausible answers. As an example, take the question Onde fica Braga? (Where is Braga located?). This generates the pattern "Braga fica" ("Braga is located") with a score of 20, that can be used to search for documents that might contain an answer to the question. The patterns used by the system have the same syntax as the one commonly used in search engines, quoted text meaning a phrase pattern.

Then, these patterns are searched in the Web (using Google at the moment) and the system extracts the first 100 document snippets created by the search engine. Some tests performed with Esfinge showed that certain types of sites may compromise the quality of the returned answers. With that in mind, the system uses a list of address patterns which are not to be considered (it does not consider documents stored in addresses that match these patterns). The patterns in this list (such as blog, humor, piadas) were created manually based on the fore mentioned tests.

The next step involves the extraction of word n-grams (length 1 to 3) from the document passages obtained previously. The system uses the Ngram Statistic Package (Banerjee and Pedersen, 2003) for that purpose.

These n-grams are scored using the formula:

N-gram score = $\sum$ (F * S * L), through the first 100 snippets resulting from the web search; where F is the n-gram frequency, S is the score of the search pattern that recovered the document and L is the n-gram length.

Identifying the type of question can be quite useful in the task of searching for an answer. For

example a question beginning with *When* suggests that most likely the answer will be a date. Esfinge has a module that uses the named entity recognition (NER) system SIEMES to detect specific types of answers. This NER system detects and classifies named entities in a wide range of categories (Sarmento, submitted). Esfinge used a sub-set of these categories, namely Human, Country, Settlement (including cities, villages, etc), Geographical Locations (locations with no political entailment, like for example Africa), Date and Quantity. When the type of question leads to one or more of those named entity categories, the 200 best scored word n-grams from the previous modules are submitted to SIEMES. The results from the NER system are then analysed in order to check whether it recognizes named entities classified as one of the desired categories. If such named entities are recognized, their position in the ranking of possible answers is pushed to the top (and they will skip the filter "Interesting PoS" described ahead).

In the next module the list of possible answers (by ranking order) is submitted to several filters:

- A filter that discards words contained in the questions. Ex: the answer Eslováquia is not desired for the question Qual é a capital da Eslováquia? (What is the capital of Slovakia?) and should be discarded.
- A filter that rejects answers included in a list of "undesired answers". This list includes very frequent words that do not answer questions alone (like pessoas/persons, nova/new, lugar/place, grandes/big, exemplo/example). It was built with the help of Esfinge log (which records all the answers analysed by the system). Later some other answers were added to this list, as a result of tests performed with the system. The list includes now 92 entries.
- A filter that uses the morphological analyzer jspell (Simões and Almeida, 2002) to check the PoS of the various tokens in each answer. This filter rejects the answers whose first and last answer are not common or proper nouns, adjectives or numbers. Using this simple technique it is possible to discard incomplete answers beginning or ending with prepositions or interjections for example.

Figure 1 describes the algorithm steps related to named entity recognition/classification in the n-grams and n-gram filtering.

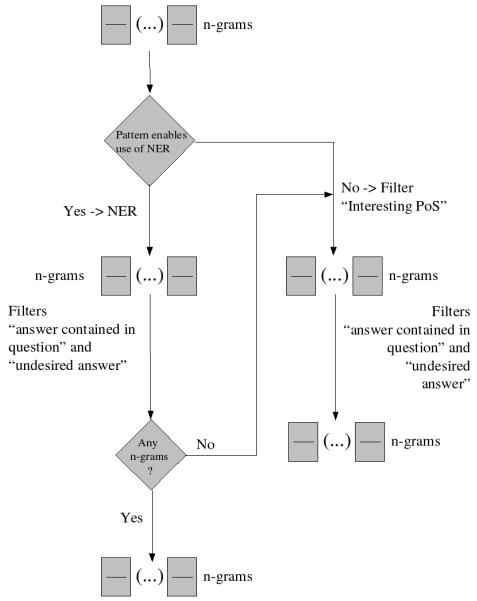

Figure 1. Named entity recognition/classification and filtering in the n-grams

The final answers of the system are the best scored candidate answers that manage to go through all the previously described filters. There is a final step in the algorithm where the system searches for longer answers. These are answers that include one of the best candidate answers and also pass all the filters. For example, the best scored answer for the question *Who is the British prime minister?* might be just *Tony*. However, if the system manages to recover the n-gram *Tony Blair* and this n-gram also passes all the filters, it will be the returned answer.

Figure 2 gives an overview of the several steps of the question answering algorithm.

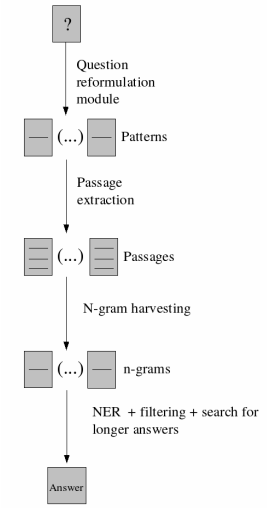

Figure 2. The architecture of Esfinge

Figure 3 shows how the system returns the results. Each answer is followed by some passages of documents from where the answers were extracted. Clicking on a passage, the user navigates to the document from which the passage was extracted. This enables the user to check whether the answer is appropriate or to find more information related to the formulated question.

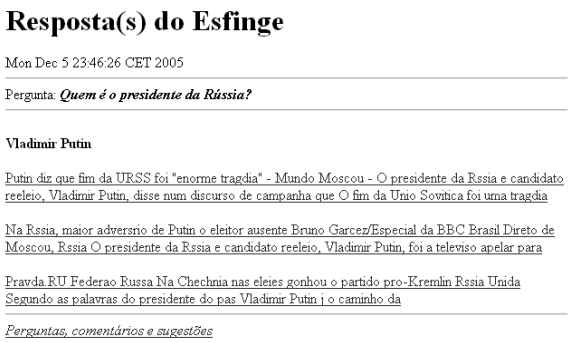

Figure 3. Esfinge answers to the question "Who is the Russian president?"

At the moment, Esfinge is installed in a Pentium 4 – 2.4 GHz machine running Red Hat Linux 9, with 1 GB of RAM memory and it can take from one to two minutes to answer a question.

Figure 4 shows the modules and data flow in the QA system. The external modules are represented as white boxes, while the modules specifically developed for the QA system are represented as grey boxes.

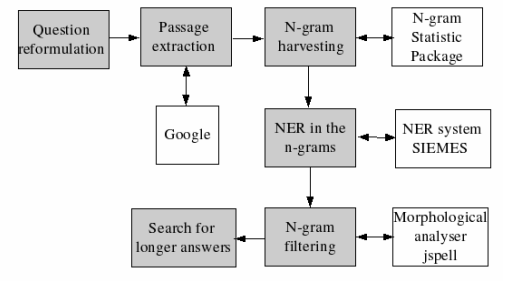

Figure 4. Modules and data flow

3 Results

In order to measure the evolution and the performance of the different techniques used, Esfinge participated in the QA task at CLEF in 2004 and 2005 (Vallin et al, 2005).

In this task the participants receive 200 questions prepared by the organization and a document collection. The systems are then supposed to return the answers to each question, indicating also the documents that support each of the answers. The questions are mainly factoid (ex: *Who is the president of South Africa?*), but there are also some definitions (ex: *Who is Angelina Jolie?*).

Esfinge needed some extra features to participate in the QA task at CLEF. While in its original version, the document retrieval task was left to Google, in CLEF it is necessary to search in the CLEF document collection in order to return the documents supporting the answers. For that purpose this document collection was encoded with CQP (Christ et al, 1999) and a document retrieval module was added to the system.

Two different strategies were tested. In the first one, the system searched the answers in the Web and used the CLEF document collection to confirm these answers. In the second experiment, Esfinge searched the answers in the CLEF document collection only.

Table 1 presents the results obtained by Esfinge at CLEF 2004 and 2005. Due to these participations some errors were detected and corrected. The table also includes the results obtained by the current version of the system with the CLEF questions in 2004 and 2005, as well as the results of the best system (U. Amsterdam) and the best system for Portuguese (University of Évora) in 2004 and 2005 (where Priberam's system for Portuguese got the best results among all the systems).

	System	Number of questions	Number (%) of exact answers
CLEF 2004	Esfinge	199	30 (15%)
	Esfinge (current version)	199	55 (28%)
	Best system for Portuguese	199	56 (28%)
	Best system	200	91 (46%)
CLEF 2005	Esfinge	200	48 (24%)
	Esfinge (current version)	200	61 (31%)
	Best system	200	129 (65%)

Table 1. Results at CLEF 2004 and 2005

We tried to investigate whether CLEF questions are the most appropriate to evaluate a system like Esfinge. With that intention 20 questions were picked randomly and Google was queried to check whether it was possible to find answers in the first 100 returned snippets. For 5 of the questions no answers were found, there were few occurrences of the right answer (3 or less) for 8 of the questions and for only 7 of the questions there was some redundancy (4 or more right an-

swers). There are more details about the evaluation of the system in (Costa, 2006).

4 Conclusions

Even though the results in CLEF 2005 improved compared to CLEF 2004, they are still far from the results obtained by the best systems. However, there are not many question answering systems developed for Portuguese and the existing ones rely heavily on the pre-processing of document collections. Esfinge tries to explore a different angle, namely the use of the web as a corpus where information can be found and extracted. It is not proved that CLEF questions are the most appropriate to evaluate the system. In the experiment described in the previous section, it was possible to get some answer redundancy in the web for less than half of the analyzed questions. We plan to study search engine logs, in order to find whether it is possible to build a question collection with real users' questions.

Since Esfinge is a project in the scope of Linguateca (http://www.linguateca.pt), it follows Linguateca's main assumptions. For example, the one stating that all research results should be made public. The web interface, where everyone can freely test the system was the first step in that direction, and now the source code of modules used in the system is freely available to make it more useful for other researchers in this area.

Acknowledgements

I thank Diana Santos for reviewing previous versions of this paper, Alberto Simões for the hints on using the perl module "jspell". Luís Sarmento, Luís Cabral and Ana Sofia Pinto for supporting the use of the NER system SIEMES. This work is financed by the Portuguese Fundação para a Ciência e Tecnologia through grant POSI/PLP/43931/2001, co-financed by POSI.

References

Rachel Aires & Diana Santos. "Measuring the Web in Portuguese". In Brian Matthews, Bob Hopgood & Michael Wilson (eds.), *Euroweb 2002 conference* (Oxford, UK, 17-18 December 2002), pp. 198-199.

Carlos Amaral et al. 2005. "Priberam's question answering system for Portuguese". In *Cross Language Evaluation Forum: Working Notes for the CLEF 2005 Workshop (CLEF 2005)* (Vienna, Austria, 21-23 September 2005).

Satanjeev Banerjee and Ted Pedersen. 2003. "The Design, Implementation, and Use of the Ngram Statistic Package". In: *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics* (Mexico City, February 2003) pp. 370-381.

Eric Brill. 2003. "Processing Natural Language without Natural Language Processing". *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics.* Mexico City, pp. 360-9.

Oliver Christ, Bruno M. Schulze, Anja Hofmann, and Esther König. 1999. The IMS Corpus Workbench: Corpus Query Processor (CQP): User's Manual. University of Stuttgart, March 8, 1999 (CQP V2.2)

Luís Costa. 2005. "First Evaluation of Esfinge - a Question Answering System for Portuguese". In Carol Peters et al (eds.), *Multilingual Information Access for Text, Speech and Images: 5th Workshop of the Cross-Language Evaluation Forum (CLEF 2004)* (Bath, UK, 15-17 September 2004), Heidelberg, Germany: Springer. Lecture Notes in Computer Science, pp. 522-533.

Luís Costa. 2005. "20th Century Esfinge (Sphinx) solving the riddles at CLEF 2005". In *Cross Language Evaluation Forum: Working Notes for the CLEF 2005 Workshop (CLEF 2005)* (Vienna, Austria, 21-23 September 2005).

Luís Costa. 2006. "Component evaluation in a question answering system". In *Proceedings of LREC 2006* (Genoa, Italy, 24-26 May 2006).

Paulo Quaresma and Irene Rodrigues. 2005. "A Logic Programming-based Approach to the QA@CLEF05 Track". In *Cross Language Evaluation Forum: Working Notes for the CLEF 2005 Workshop (CLEF 2005)* (Vienna, Austria, 21-23 September 2005).

Luís Sarmento. "SIEMÊS – a named entity recognizer for Portuguese relying on similarity rules" (submitted).

Alberto Simões and José João Almeida. 2002. "Jspell.pm - um módulo de análise morfológica para uso em Processamento de Linguagem Natural". In: Gonçalves, A. & Correia, C.N. (eds.): *Actas do XVII Encontro da Associação Portuguesa de Linguística (APL 2001)* (Lisboa, 2-4 Outubro 2001). APL Lisboa, pp. 485-495.

Alessandro Vallin et al. 2005. "Overview of the CLEF 2005 Multilingual Question Answering Track". In *Cross Language Evaluation Forum: Working Notes for the CLEF 2005 Workshop (CLEF 2005)* (Vienna, Austria, 21-23 September 2005).