Esfinge at CLEF 2008: Experimenting with answer retrieval patterns. Can they help?

Luís Costa

Linguateca, SINTEF ICT, NORWAY

luis.costa@sintef.no

Abstract. Esfinge is a general domain Portuguese question answering system which has been participating at QA@CLEF since 2004. It uses the information available in the "official" document collections used in QA@CLEF (newspaper text and Wikipedia), but additionally it also uses information from the Web as an additional resource when searching for answers. Where it regards the use of external tools, Esfinge uses a syntactic analyzer, a morphological analyzer and a named entity recognizer. This year an alternative approach to retrieve answers was tested: whereas in previous years, search patterns were used to retrieve relevant documents, this year a new type of search patterns was also used to extract the answers themselves. Besides that we took advantage of the main novelty introduced this year by QA@CLEF organization which was that the systems could return up to three answers for each question, instead of the single answer allowed in previous editions. This enabled the investigation about how good were the second and third best answers returned by Esfinge (when the first answer is not correct). The experiments revealed that the answer retrieval patterns created for this participation improve the results, but only for definition questions. Regarding the study of the three answers returned by Esfinge, the conclusion was that when Esfinge answers correctly a question, it does so usually with its first answer.

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 Management]: Languages—Query Languages

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, named-entity recognition, anaphoric reference, Portuguese

1 Introduction

The proposed task in this year at QA@CLEF was quite similar to the previous year. The main novelty was the systems could return up to three answers for each question. Besides taking advantage of that possibility, our participation focused in using an alternative approach to retrieve answers. In previous years, search patterns were used to retrieve relevant documents to particular questions. This year we also used search patterns to extract the answers themselves.

The following sections describe in detail the system architecture used this year, how the answer retrieval patterns were created and the results obtained in the official runs. There is a final section where the results are discussed and where directions for future work are indicated.

2 Esfinge in 2008

Esfinge has been participating at CLEF since 2004. These participations are described in detail in [2, 3, 4, 5]. Figure 1 gives an overview of the system used this year:
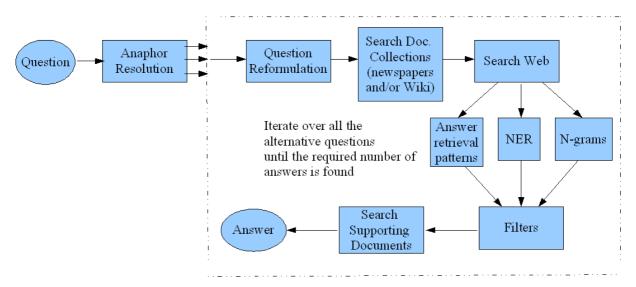

Figure 1. Modules used in Esfinge

The *Anaphor Resolution* module is the first module in Esfinge. It adds to the original questions, a list of alternative questions where the anaphors are tentatively resolved. This module uses the analysis of the PALAVRAS parser [1] to identify the anaphoric element in a question and a list of candidates to replace it in the context of the other questions in the same topic.

For each of the alternative questions and until the requested number of answers is found:

The Question Reformulation Module transforms the question into patterns of text where it is plausible to find answers. This is done using two different approaches: a purely string matching technique and an alternative approach which uses the analysis of PALAVRAS.

Esfinge starts with the string matching technique. This technique uses patterns which have an associated score giving an indication about how likely the pattern will help to retrieve relevant text passages.
For example the question *Quem foi Baden Powell de Aquino?* matches with the patterns (simplified here for illustration purposes):

Quem ([^\s?]) ([^?]*)\??/"$2 $1"/10*
Quem ([^?])\??/$1/1*

Which in turn generate the following (Text pattern/Score) pairs:

"Baden Powell de Aquino foi"/10
foi Baden Powell de Aquino/1

2.1. Searching and supporting answers

Text patterns are then searched in the document collections (newspapers and Portuguese Wikipedia) using the Search Document Collections module in order to find text passages that are relevant to the question. These patterns are also searched in the Web using Yahoo's search API.
Then, all the relevant texts retrieved are analyzed in order to obtain candidate answers. Three techniques are used for that purpose: extraction of answers using the answer retrieval patterns created for this year's participation, the named entity recognizer (NER) system SIEMÊS [6] and an n-grams computing module. Figure

2 illustrates how these techniques are used: namely each of them is used in sequence until Esfinge finds the required number of answers.
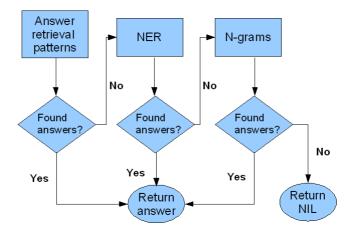

Figure 2. Answer retrieval techniques

Esfinge begins by extracting answers using answer retrieval patterns. These patterns associate questions with their respective answers. They are different from the patterns used to retrieve relevant documents because they include the position where the answers should appear.

The following pattern is a simplified example of the patterns used by Esfinge. The patterns that are actually used are a bit more general in the sense that they can for example cater for some alternative verb tenses (*é*, *são*, *eram*) and alternative articles (*a*, *o*, *as*, *os*, *um*, *uma*):

O que é a __X__? → *__X__ (__ANSWER__)*

This means that the answer to a question like *O que é a __X__?* can be retrieved inside parenthesis following the string *__X__*.

First, Esfinge checks which patterns match with the question (left hand side of the rule). The pattern in the right hand side of the rule is then checked against the relevant documents to the question with the purpose of finding candidate answers.

The candidate answers are then ranked according to their frequency, length and the score of the passage from where they were retrieved using the formula:

Candidate answer score = $\sum$ (F * S * L), through the passages retrieved in the previous modules where:

F = Candidate answer frequency
S = Score of the passage
L = Candidate answer length[1]

At this stage Esfinge has a list of candidate answers {A_1, A_2 … A_n}. These candidate answers are then tested using the following filters:

• A filter that excludes answers contained in the questions. For example, the answer *partido* (party) is not a good answer to the question *A que partido pertence Zapatero?* (To which party does Zapatero belong?).

• A filter that excludes answers contained in a list of undesired answers (very frequent words that usually can not answer questions). This list includes words like *parte* (part)*, antigo* (old)*, pessoas* (people)*, mais* (more) and was created based on experiments performed with the system. At present, it contains 96 entries.

---

[1] This parameter is only used for the answers obtained through the n-grams module. For the other answers this is set to 1.

An answer that passes all the filters proceeds to the next module which checks whether there are documents in the collections which support the answer.

If Esfinge does not find the requested number of answers using the answer retrieval patterns, it tries to get more answers using the named entity recognizer (NER) system SIEMÊS [6]. This system is used for the questions which imply specific types of answers like *Place*, *People*, *Quantity*, *Date*, *Height*, *Duration*, *Area*, *Organization* or *Distance*. Esfinge uses pattern matching to check whether it is possible to infer the type of answer for a given question. For example, questions starting with *Onde* (Where*)* imply an answer of type *Place*, questions starting with *Quando* (When) imply an answer of type *Date*, questions starting with *Quantos* (How Many) imply an answer of type *Quantity*. For these questions, SIEMÊS tags the relevant text passages in order to count the number of occurrences of named entities belonging to the expected categories. The identified named entities are then ranked, filtrated and checked for the existence of documents which can support them in a similar manner as described previously for the answers obtained using answer retrieval patterns.

In case the previous efforts still do not yield the necessary number of answers, Esfinge uses its last answer retrieval technique which consists in using an n-grams module. The n-grams computed through this module are also ranked, filtrated and checked for the existence of documents which can support them in a similar manner as described previously for the other two techniques. The answers obtained through n-gram harvesting, however, are submitted to an additional filter that uses the morphological analyzer jspell [7] to check the PoS of the words contained in the answer. Jspell returns a list of tags for each of the words. Esfinge rejects all answers in which the first and last word are not tagged as one of the following categories: adjectives (adj), common nouns (nc), numbers (card) and proper nouns (np).

2.2. Alternative techniques used to find relevant texts

If at this point of execution, Esfinge did not retrieve the required number of answers, the next step will be to select a new set of relevant texts (this time patterns based on the analysis of the question by PALAVRAS [1]). These patterns are created using the main verb of the question, its arguments and adjuncts and entities from previous questions belonging to the same topic. From this stage, Esfinge repeats the steps described in sub-section 2.1.

In case the last step did not yield the required number of answers either, a last attempt is tried which consists in selecting relevant texts using again the patterns based on the analysis of the question by PALAVRAS, but excluding the verbs included in these patterns.

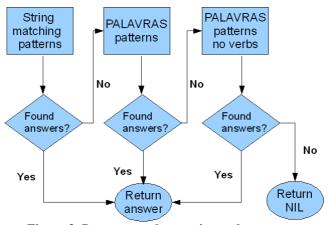Figure 3 gives an overview about the three techniques used by Esfinge to select relevant texts.

Figure 3. Patterns used to retrieve relevant texts

3 Creating the list of answer retrieval patterns

The hypothesis we wanted to test this year was whether it would be possible to extract useful answer retrieval patterns from the solutions available from the previous edition of CLEF. Unfortunately, only the solutions for 2007 questions were available (answers and text passages where they occur), since the requirement to support answers with text snippets was only introduced on that year.

The following is an example of the solution for the question *Quem é o Lampadinha?* (Who is Lampadinha?) :

<pergunta ano="2007" id_org="X" new_id_org="070101" categoria="D" tipo="PERSON" restrição="NO" ling_orig="PT" tarefa_pt="0101" tópico="082">
*<texto>Quem é o **Lampadinha**?</texto>*
<resposta n="1" docid="Professor Pardal 6afa">
um pequeno andróide com uma lâmpada no lugar da cabeça</resposta>
*<extracto n="1" resposta_n="1">"Pardal é ajudado frequentemente por **Lampadinha** (criado por Barks em 1953), **um pequeno andróide com uma lâmpada no lugar da cabeça**, que é considerado sua maior invenção (ao lado do ""chapéu pensador"", um dispositivo em forma de telhado com chaminé habitado por corvos, que o ajuda a ter idéias)."</extracto>*
</pergunta>

From this solution one can derive the following pattern: *Quem é o __X__ ?* ➔ *__X__ *, __ANSWER__,*

This pattern means that the answer for a question of the form *Quem é o X ?* can be retrieved following a comma which appears after an instance of X. The asterisk (*) stands for 0 or more characters after the sub-string *__X__* and immediately before a comma.

For the participation at CLEF, 24 answer retrieval patterns were derived. The process used to obtain these patterns was semi-automatic: they were derived automatically from the solution file, but then adjusted manually, not only in order to correct or complete them, but also to generalize them.

4 Results

Since the main goal of this participation was to evaluate the impact of the answer retrieval patterns described in the previous section, two official runs were submitted: *esfi081PTPT* which uses these patterns and *esfi082PTPT* which does not use them.

Table 1 and Figure 4 show the results of the official runs, considering all the questions (Total), only for factoid (F) or definition (D) questions. Table 1 includes results where only the first answer is evaluated (labeled *first answer* in the table and *esfg08Nptpt* in the figure), where the first and second answers are evaluated (*first 2 Answers* in the table and *esfg08Nptpt2* in the figure) and where all the three answers returned are evaluated (*first 3 Answers* in the table and *esfg08Nptpt3* in the figure). As an illustration, take a question for which only the third answer is correct: this question is only accounted in the columns labeled *first 3 answers*.

Runs	Right Answers (first answer)			Right Answers (first 2 Answers)			Right Answers (first 3 Answers)			Inexact Answers	Unsupported Answers
	F	**D**	**Total**	**F**	**D**	**Total**	**F**	**D**	**Total**		
esfi081ptpt	39	10	49	44	14	58	46	16	62	14	5
esfi082ptpt	39	2	41	43	3	46	46	6	52	18	6

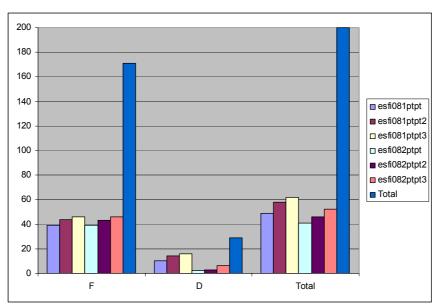**Table 1: Results of the official runs**

Figure 4. Results of the official runs

There were no further developments of the anaphor resolution module of Esfinge for this participation, but since a considerable amount of this year's questions (51 questions) require the processing of anaphors, a study was performed to evaluate how well Esfinge dealt with the anaphors in this year's questions. Table 2 summarizes the results of this analysis: more than half of the anaphors are well resolved (26 out of 51) and there were additionally 6 useful resolutions. By useful anaphor resolutions, we mean questions which are not grammatically correct or where it is not possible to say that the anaphor was completely resolved, but where the resolution can nevertheless lead to the discovery of right answers given the techniques used in Esfinge. If one does not consider the questions which require anaphor resolution, the precision increases slightly: 28% against 24.5% when considering all the questions.

Resolution of anaphors	Quantity
Well resolved	26
Useful, but not well resolved	6
Not resolved	19
Total	51

Table 2: Evaluation of the anaphor resolution module

An additional study was performed in order to find how Esfinge supports its right answers (newspaper text or Wikipedia) in the best run. The results, summarized in Table 3, reveal that Esfinge finds support in Wikipedia more often than in newspaper texts both for factoid and definition questions.

Type of Question	Answer supported in newspaper text	Answer supported in Wikipedia	Right Answers
Right Definition Answers	4	6	10
Right Factoid Answers	14	21	35
Right NIL Answers			4
Right Answers	18	27	49

Table 3: Distribution of the answer support

5 Discussion of the results and further work

We consider our participation at QA@CLEF this year fruitful and rewarding. In our opinion it was wise that the organization proposed a similar task as last year's since a good number of challenges remain to be achieved.

Nevertheless, the main novelty in this years's task which consisted in allowing the return of up to three answers for each question, allowed us to investigate how good the second and third best answers returned by Esfinge are. The conclusion regarding this matter is that when Esfinge answers correctly a question, it does so usually with its first answer (for example the best run has 49 right answers, but even when considering all the answers returned (3 for each question) the number of right answers amount only to 62 right answers.

Nonetheless the most relevant result obtained in this year's participation was that the answer retrieval patterns clearly improved the results for definition questions (the first answer is correct for 34% of the definition questions and there is a correct answer in one of the three returned answers for 55% of questions of this type), but the same does not applied for the factoid questions.

A more detailed error analysis is due in order to better understand the current limitations of Esfinge, but we believe that there is still improvement potential where it regards the use of answer retrieval patterns. We would like to deepen our research on how to create these patterns using a more automated approach (as stated we used a semi-automatic process taking as input last year's solutions). Equally there is also interest in investigating how the results can improve when more answer retrieval patterns are used.

References

1. Eckhard Bick. The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus: Aarhus University Press (2000)
2. Luís Costa. First Evaluation of Esfinge - a Question Answering System for Portuguese. In Carol Peters, Paul Clough, Julio Gonzalo, Gareth Jones, Michael Kluck & Bernardo Magnini (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), Heidelberga, Alemanha: Springer. LNCS, pp. 522-533, 2005. Revised Selected Papers.
3. Luís Costa: 20th Century Esfinge (Sphinx) solving the riddles at CLEF 2005. In Carol Peters, Frederic C. Gey, Julio Gonzalo, Gareth J.F. Jones, Michael Kluck, Bernardo Magnini, Henning Müeller & Maarten de Rijke (eds.): *6th Workshop of the Cross-Language Evaluation Forum (CLEF'2005)* (Vienna, Austria, 21-23 September 2005), Springer. LNCS 4022, pp. 467-476, 2006. Revised Selected Papers.
4. Luís Costa. Question answering beyond CLEF document collections. In Carol Peters, Paul Clough, Fredric C. Gey, Douglas W. Oard, Maximilian Stempfhuber, Bernardo Magnini, Maarten de Rijke & Julio Gonzalo (eds.): *7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006. Alicante, Spain, September 2006.* Berlin / Heidelberg: Springer, Lecture Notes in Computer Science, 2007. Revised Selected Papers.
5. Luís Miguel Cabral, Luís Fernando Costa and Diana Santos "What happened to Esfinge in 2007?". In Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Müller, Doug W. Oard, Anselmo Peñas, Vivien Petras & Diana Santos (eds.), *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers.* Berlin: Springer, Lecture Notes in Computer Science, 2008.
6. Luís Sarmento. SIEMÊS - A Named Entity Recognizer for Portuguese Relying on Similarity Rules. In Renata Vieira, Paulo Quaresma, Maria da Graça Volpes Nunes, Nuno J. Mamede, Cláudia Oliveira & Maria Carmelita Dias (eds.), *Computational Processing of the Portuguese Language: 7th International Workshop, PROPOR 2006. Itatiaia, Brazil, May 2006 (PROPOR'2006)* LNAI 3960, 13-17 May 2006, Berlin/Heidelberg: Springer Verlag, pp. 90-99.
7. Alberto Manuel Simões and José João Almeida. "Jspell.pm - um módulo de análise morfológica para uso em Processamento de Linguagem Natural". In Anabela Gonçalves & Clara Nunes Correia (eds.), *Actas do XVII Encontro Nacional da Associação Portuguesa de Linguística (APL 2001)* (Lisboa, 2-4 de Outubro de 2001), Lisboa: APL, pp. 485-495.