

# Using Answer Retrieval Patterns to Answer Portuguese Questions

Luís Fernando Costa

Linguatca, Oslo Node, SINTEF ICT  
Pb 124 Blindern, 0314 Oslo, 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) and 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. We also evaluated the second and third best answers returned by Esfinge. This evaluation showed that when Esfinge answers correctly a question, it does so usually with its first answer. Furthermore, the experiments revealed that the answer retrieval patterns created for this participation improve the results, but only for definition questions.

**Keywords:** Question Answering, Portuguese, Answer Extraction

## 1 Introduction

The proposed task in this year at QA@CLEF [1] was quite similar to the previous year. The main novelty was that 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 a new type of 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 at CLEF 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:

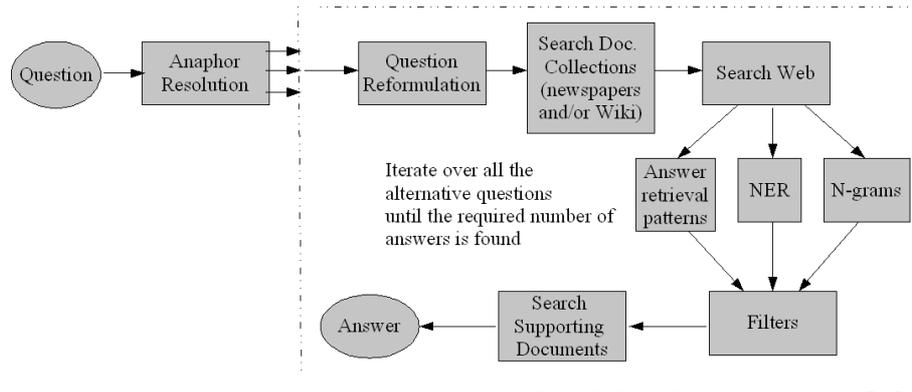


Fig. 1. Modules used in Esfinge

For a question, there can be up to three main iterations involved: the first over an eventual set of anaphor resolved question candidates, then an iteration over the different types of patterns used to retrieve relevant text passages and finally an iteration over the three techniques used to obtain candidate answers.

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 [6] 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. This module is described in much more detail in [5].

Then, for each of the alternative questions generated by the *Anaphor Resolution* module and until Esfinge finds the requested number of answers:

The *Question Reformulation* module transforms the question into patterns that will be later used to retrieve text passages which are relevant to the question. 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?* (Who was Baden Powell de Aquino?) matches with the patterns (simplified here for illustration purposes):

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

*Quem ([^?]\*)\?/?\$/I/I*

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<sup>1</sup>.

Subsequently, Esfinge analyzes all the retrieved relevant texts 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, using named entity recognition (NER) and using an n-grams harvesting module. These techniques are used in sequence until Esfinge finds the required number of answers.

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 instance 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\_\_?* (What is the X?) can be retrieved inside parenthesis following the string *\_\_X\_\_*.

First, Esfinge checks which patterns match with the question (left hand side of the rules). The patterns on the right hand side of the rules are then searched in 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 and L = Candidate answer length<sup>2</sup>.

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

---

<sup>1</sup> <http://developer.yahoo.com/search/web/V1/webSearch.html>

<sup>2</sup> This parameter is only used for the answers obtained through the n-grams module. For the other answers this is set to 1.

- A filter that excludes answers contained in the question. For instance, 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.

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 NER system SIEMÊS [7]. 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* and questions starting with *Quantos* (How Many) imply an answer of type *Quantity*. For these questions, Esfinge uses SIEMÊS to tag 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: n-grams harvesting. The n-grams obtained through the *N-grams* 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 [8] to check the PoS of the words contained in the answer. Jspell returns a list of tags for each of the words and Esfinge rejects all answers in which the first and last word are not tagged as one of the following categories: adjectives, common nouns, numbers and proper nouns.

## 2.2. Alternative Techniques Used to Find Relevant Texts

If at this stage, Esfinge did not retrieve the required number of answers, the next step will be to select a new set of relevant texts (this time using patterns based on the analysis of the question by PALAVRAS [6]). 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 patterns without verbs based on the analysis of the question by PALAVRAS.

### 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 editions of CLEF. Unfortunately, only the solutions for 2007 questions were available for Portuguese (answers and text passages where they occur).

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 ?* (Who is 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 2008, 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 shows 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 answers were evaluated, where the first and second answers were evaluated and where all the three answers returned were evaluated. As an illustration, take a

question for which only the third answer was correct: this question was only accounted in the columns labeled *First 3 answers*.

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

Runs	Right Answers (First Answer)			Right Answers (First 2 Answers)			Right Answers (First 3 Answers)		
	F	D	Total	F	D	Total	F	D	Total
<b>esfi081ptpt</b>	39	10	49	44	14	58	46	16	62
<b>esfi082ptpt</b>	39	2	41	43	3	46	46	6	52

Table 2 summarizes the performance of the three techniques used to obtain answers in the best run (we studied only the first answer for each question). As previously mentioned these techniques are used in sequence until the required number of answers is obtained. The first technique which is tested (answer retrieval patterns) was used in a small number of questions, but with a good precision (46% of right answers for the answers obtained using this technique). NER, the second technique, catered for most of Esfinge’s right answers, but with lower precision (32%). The last technique, n-grams harvesting, had the lowest precision (6%) and contributed with less right answers to the overall result.

**Table 2.** Origin of the answers

Answer origin	Answers	Right Answers
Answer Retrieval Patterns	24	11
NER	93	30
N-gram	63	4
NIL	20	4
Total	200	49

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

**Table 3.** Distribution of the answer support

Type of Question	Answer Supported with Newspaper Text	Answer Supported with 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 4 provides the results of the detailed error analysis performed for the best runs of Esfinge in 2008 and 2007<sup>3</sup>.

<sup>3</sup> The 2007 analysis was not performed on an official run (those results were compromised by severe bugs), but on the best run out of the repetitions executed after the bugs were corrected.

**Table 4.** Causes for wrong answers in the best run

<b>Problem</b>	<b>Wrong Answers in 2008</b>	<b>Wrong Answers in 2007</b>
Co-reference Resolution	20	25
Wrong or Incomplete Search Patterns	5	63
Document Retrieval Failure	44	33
Named Entity Recognition	13	3
Answer Scoring Algorithm	45	24
Mistake in the Supported Answer Filter	13	7
Others	11	10
Total	151	165

## 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.

The main novelty in this year'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 instance the best run had 49 right answers, but even when considering all the answers returned (3 for each question) the number of right answers amounted only to 62 right answers.

Regarding the study on the performance of the three techniques used to obtain answers, our findings were that the first technique which is tested (answer retrieval patterns) had a good precision (46%) in the small number of questions where it was used; NER, the second technique, catered for most of Esfinge's right answers, at a lower precision (32%); the last technique, N-gram harvesting, had the lowest precision (6%) and contributed with less right answers to the overall result. However, this last technique is only used when it is not possible to find answers with the other two. This means that it is probably used with a considerable number of the hardest questions.

The error analysis shows that, comparing with 2007 experiments, errors occur more often in modules which appear later in the system's workflow. Whereas in 2007, most errors were caused by wrong or incomplete search patterns, this year they were mainly caused by document retrieval failure or the answer scoring algorithm (Table 4).

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 was a correct answer in one of the three returned answers for 55% of questions of this type), but the same does not apply for the factoid questions. These patterns were used in a small number of questions, but the precision of the answers was quite good (46%). This good precision confirmed our intuition that the best order to search for answers would be the

one used in our system: first using answer retrieval patterns, then NER and finally n-grams harvesting.

We believe that there is still improvement potential where it regards the use of answer retrieval patterns. Therefore, we would like to deepen our research on how to create these patterns with a more automated approach (as stated we used a semi-automatic process taking as input last year's solutions). Additionally, there is also interest in investigating how the results can improve when more answer retrieval patterns are used.

### Acknowledgments

This work was done in the scope of the Linguatca project, jointly funded by the Portuguese Government and the European Union (FEDER and FSE) under contract ref. POSC/339/1.3/C/NAC.

### References

1. Forner, P., Peñas, A., Alegria, I., Forăscu, C., Moreau, N., Osenova, P., Prokopidis, P., Rocha, P., Sacaleanu, B., Sutcliffe, R., Sang, E.T.K.: Overview of the CLEF 2008 Multilingual Question Answering Track". In this volume.
2. Costa, L.: First Evaluation of Esfinge - a Question Answering System for Portuguese. In Peters, C., Clough, P., Gonzalo, J., Jones, G., Kluck, M., Magnini, B. (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, Alemanha: Springer. LNCS, pp. 522-533, 2005. Revised Selected Papers.
3. Costa, L.: 20<sup>th</sup> Century Esfinge (Sphinx) solving the riddles at CLEF 2005. In Peters, C., Gey, F.C., Gonzalo, J., Jones, G., Kluck, M., Magnini, B., Müeller, H., de Rijke, M. (eds.): *6<sup>th</sup> 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. Costa, L.: Question answering beyond CLEF document collections. In Peters, C., Clough, P., Gey, F.C., Oard, D.W., Stempfhuber, M., Magnini, B., de Rijke, M., Gonzalo, J. (eds.): *7<sup>th</sup> 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. Cabral, L.M., Costa, L.F., Santos, D.: "What happened to Esfinge in 2007?". In Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.), *Advances in Multilingual and Multimodal Information Retrieval: 8<sup>th</sup> 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. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus: Aarhus University Press (2000)
7. Sarmento, L.: SIEMÊS - A Named Entity Recognizer for Portuguese Relying on Similarity Rules. In Vieira, R., Quaresma, P., Nunes, M.G.V., Mamede, N.J., Oliveira, C., Dias, M.C. (eds.), *Computational Processing of the Portuguese Language: 7<sup>th</sup> International Workshop, PROPOR 2006. Itatiaia, Brazil, May 2006 (PROPOR'2006)* LNAI 3960, 13-17 May 2006, Berlin/Heidelberg: Springer Verlag, pp. 90-99.
8. Simões, A.M., Almeida, J.J. : "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 Nacional da Associação Portuguesa de Linguística (APL 2001)* (Lisboa, 2-4 de Outubro de 2001), Lisboa: APL, pp. 485-495.