Esfinge – a modular question answering system for Portuguese

Luís Costa

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

Abstract. Esfinge is a general domain Portuguese question answering system. It uses the information available in the Web as an additional resource when searching for answers. Other external resources and tools used are a translation tool, a syntactic analyzer, a morphological analyzer, a named entity recognizer and a database of word co-occurrences. In this third participation in CLEF, the main goals were to check whether a database of word co-occurrences could improve the answer scoring algorithm and to get more benefit from the use of a named entity recognizer that was used last year in quite sub-optimal conditions. This year results were slight better than last year's which is somehow surprising due to the fact that this year's question set included more definitions of type *Que é X?* (What is X?) with which the system had the worst results in previous participations. Another interesting conclusion from this year experiments is that the Web helped more with this year's questions than in previous years. While the best run using the Web achieved an accuracy of 25%, an experiment with the same algorithm but which did not use the Web achieved only 17% in accuracy. Again the main cause for failure this year was in the retrieval of relevant documents (56% of the errors). There is a web interface to a simplified version of Esfinge at http://www.linguateca.pt/Esfinge and in this page is also possible to download the modules used by the system.

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

1 Introduction

Esfinge is a question answering system developed for Portuguese inspired on the architecture proposed by Eric Brill [1]. Brill argues that it is possible to get interesting results, applying simple techniques to large quantities of data, using redundancy to compensate for lack of detailed analysis. The Web in Portuguese can be an interesting resource for such architecture (see [2]).

Esfinge participated at CLEF both in 2004 and 2005 editions demonstrating the applicability of this approach to Portuguese. Even though the main goal for the development of the system this year was to refactor and improve the code developed in previous years with a modular approach to make this work useful for other people interested in question answering and to make future developments easier, in this participation at CLEF 2006, we also managed to test the use of a new resource (a database of word co-occurrences) and to get more benefit from the use of a named entity recognizer that was used last year in quite sub-optimal conditions.

The modules used in this year's participation are described in the following section. In this paper we also present the results obtained in the official runs this year, an error analysis for these runs, the results obtained by this year's system with the 2004 and 2005 questions and the results obtained in other experiments that were not submitted. Based on the results and error analysis we also discuss some of the evaluation guidelines used in CLEF.

2 Esfinge 2006

CLEF 2006 brought some new challenges to the participating QA systems. The most relevant were:

- The answers needed to be supported by text snippets extracted from the document collection (last year only a document ID was required).

- Some list questions were included (questions requiring a list of items as answers).

Figure 1 gives a general overview of the algorithm used in Esfinge:

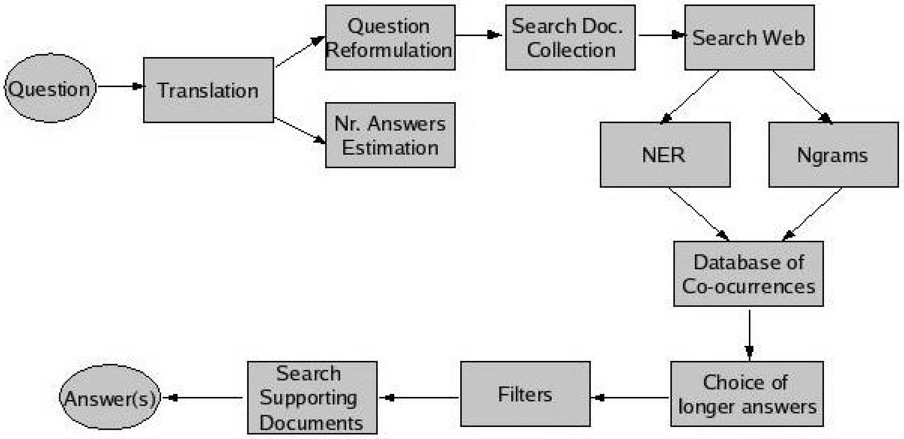

Figure 1. Modules used in Esfinge

If the questions are not in Portuguese, they are translated to Portuguese using the module Lingua::PT::Translate.

Esfinge uses the parser PALAVRAS [3] to estimate the number of answers for a question. It proceeds with the Question Reformulation Module which transforms the question into patterns of plausible answers. These patterns are then searched in the document collection using the Search Document Collection module.

If the patterns are not found in the document collection the system returns the answer NIL (no answer found) and stops its execution. On the other hand when the system finds the patterns in the document collection, it proceeds by searching the same patterns in the Web. Then, the texts retrieved from the Web and the document collection are analyzed using the named entity recognizer SIEMÊS [4] (for the questions that imply named entity categories as answers). The found named entities of the relevant categories are then ranked according to their frequency, length and the score of the passage form where they were retrieved. This ranking is in turn adjusted using the database of co-occurrences BACO [5] and the candidate answers (by ranking order) are analyzed in order to check whether they pass a set of filters and whether there is a document in the collection which supports them.

When Esfinge does not find the necessary answers using the previous techniques, it uses the n-grams module which counts the word n-grams in the texts retrieved from the Web and the document collection. These n-grams (as the named entities of the relevant categories) are then ranked according to their frequency, length and the score of the passage from where they were retrieved, ranking which is also adjusted using the database of co-occurrences BACO.

From the moment when Esfinge finds enough answers, it checks only candidate answers that include one of the previously found answers. It will replace one of the original answers if the new one includes the original answer, also passes the filters and has documents in the collection that can support it.

Each of these modules is described in the following sub-sections.

2.1 Translation

This module was used for the EN-PT run (questions in English, answers in Portuguese). The questions (except the ones of type *Where is* X) are translated with the module Lingua::PT::Translate freely available at CPAN which provides an easy interface to the Altavista Babelfish translating tool Questions of type *Where is* X are treated in a different way because the translation tool translates them to *Onde está X* (*is* can be translated as *está*, *é* or *fica* in Portuguese). Therefore this module translates these questions to *Onde fica X* which is a more natural translation.

2.2 Number of Answers Estimation

The motivation for this module arose from the inclusion of list questions in the question set this year.

To estimate the number of answers required by a question, Esfinge used the parser PALAVRAS [1]. The algorithm is very simple; the question is submitted to PALAVRAS:

- If the noun phrase (NP) is singular Esfinge tries to find exactly one answer (e.g. *Quem era o pai de Isabel II? / Who is Elizabeth The Second's father?*).

- If the NP is plural and the parser finds a numeral in the question (e.g. *Quais são as três repúblicas bálticas? / Which are the three Baltic states?*), the system will return this number of answers (three in this case).

- If the NP is plural and the parser does not find a numeral in the question (e.g. *Quais são as repúblicas bálticas? / Which are the Baltic states?*), Esfinge returns five answers (default).

This strategy is quite naive however, and after analyzing the question set we quickly realize that for some questions with a singular NP, it is quite likely more useful to obtain several answers (e.g. *Quem é Stephen Hawking? / Who is Stephen Hawking?* , *O que é o hapkido? / What is Hapkido?*, *Diga um escritor chinês./ Mention a Chinese writer.*). This will be studied in more detail in the future.

2.3 Question Reformulation

This module transforms questions into patterns of possible answers. A given question is matched against a list of question patterns which in turn are associated with pairs (Answer pattern/Score).

For example the question *Quem é Stephen Hawking?* matches with the patterns (simplified here for illustration purposes):

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

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

"Stephen Hawking é"/10
é Stephen Hawking/1

2.4 Search the Document Collection

The document collection was encoded using IMS Corpus Workbench [6]. Each document was divided in sets of three sentences. The sentence segmentation and tokenization was done using the Perl Module Lingua::PT::PLNbase developed by Linguateca and freely available at CPAN.

Esfinge uses the named entity recognizer (NER) SIEMÊS [4] to refine the patterns obtained by the question reformulation module. The pattern *é Stephen Hawking* (is Stephen Hawking) is a good example. SIEMÊS tags *Stephen Hawking* as a *<HUM>* (Human) and therefore the original pattern is converted to *é "Stephen Hawking"*. There are other cases, however, where Esfinge does not preserve the named entities identified by the NER system. An example are titles (e.g. *Presidente do Egipto / President of Egypt*), since in many occasions the tokens *Presidente* and *Egipto* are likely to appear in different positions in a sentence (or even in different sentences). As an illustration the following text (freely adapted from a text in the Portuguese document collection) can be relevant for answer extraction:

"Yesterday it was Election Day in Egypt. The party of the President Hosny Mubarak is expected to have the best result since 1979."

The system searches these refined patterns in the document collection. If it does not find any set of three sentences matching one of the patterns, it returns the answer NIL (no answer found) and stops its execution. Otherwise, Esfinge stores the matching passages in memory $\{P_1, P_2 \ldots P_n\}$. Stop-words without context are discarded in this search.

2.5 Search theWeb

The patterns obtained in the previous modules are then searched in the Web. Esfinge uses the APIs provided by Google and Yahoo search engines for that purpose. Esfinge stores the first 50 snippets (excluding, however, the snippets retrieved from addresses containing words like *blog* or *humor*) retrieved by each of the search engines $\{S_1, S_2 \ldots S_n\}$.

2.6 Named Entity Recognition and N-gram Harvesting

Two techniques are used to extract answers from the relevant passages retrieved from the Web and the document collection: named entity recognition and n-gram harvesting.

The NER system SIEMÊS [4] is used for the questions which imply answers of type *Place*, *People*, *Quantity*, *Date*, *Height*, *Duration*, *Area*, *Organization* and *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*, etc. For these questions, SIEMÊS tags the relevant text passages in order to count the number of occurrences of NEs belonging to the relevant categories. SIEMÊS was used in 54% of the answered questions in the PT-PT task this year.

For other questions, however, either the answer is not a named entity (definition questions) or it is very time consuming to create patterns to deal with them (questions of type *Qual X* / *Which X*, where *X* can be potentially anything). For these questions (or when not enough valid answers are obtained within the candidate answers extracted with the NER), Esfinge uses the n-grams module which counts the word n-grams in the relevant text passages.

The candidate answers (obtained either through NER or n-gram harvesting) 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

This year we managed to install a local installation of SIEMÊS in our server which enabled us to analyze all the relevant passages with this NER system. In average, 30 different named entities of the relevant types were identified in the passages retrieved from the Web for each question, while 22 different named entities of the relevant types were identified in the passages retrieved from the document collection. Last year, it was not possible to analyze the whole relevant passages for practical reasons, so only the most frequent sequences of 1 to 3 words extracted from these relevant passages were analyzed which meant that the NER system was used in quite sub-optimal conditions. Nevertheless, this sub-optimal use of the NER led to some improvement in the results, as described in [7]. This year we expected still larger improvement.

2.7 Database of Co-occurrences

The score computed in the previous modules is heavily dependent on word frequency, though, and, therefore, very frequent words may appear quite often in the results. The use in a question answering system of an auxiliary corpus to check the frequency of words in order to capture an overrepresentation of candidate answers in a set of relevant passages was successfully tested before, see [8]. With that in mind, we used the database of word co-occurrences BACO [5] this year, to take into account the frequency in which n-grams appear in a large corpus. BACO includes tables with the frequencies of word n-grams from length 1 to 4 of the Portuguese web collection WPT-03, a snapshot of the Portuguese Web in 2003 (1.000.000.000 words) [9]. The scores obtained in the previous module are then adjusted, giving more weight to more rare candidates using the formula:

Candidate answer adjusted score = Candidate answer score * log (Number of words in BACO / Candidate answer frequency in BACO)

2.8 Choice of Longer Answers

From the moment when Esfinge finds enough answers, it proceeds to check only candidate answers that include one of the previously found answers. For example, for the question *O que é a Generalitat?* (What is the Generalitat?), the system finds the answer *Catalunha* first, but afterwards finds the candidate answer *governo da Catalunha* (Government of Catalonia) that includes the first, also passes the filters and has documents in the collection that can support it. Therefore the answer Esfinge returns is *governo da Catalunha*.

2.9 Filters

At this stage Esfinge has an ordered list of candidate answers $\{A_1, A_2 \ldots A_n\}$. These candidate answers are then checked using a set of filters:

- A filter that excludes answers contained in the questions. For example, the answer *aspirina* (aspirin) is not a good answer to the question *Qual o principal componente da aspirina?* (What is the main component of the aspirin?).

- 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 is updated based on experiments performed with the system. At present, it contains 96 entries.

- The answers obtained through n-gram harvesting are also submitted to a filter that uses the morphological analyzer jspell [10] 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).

An answer that passes all the filters proceeds to the next module.

2.10 Search for Supporting Document

In this module, the system tries to find a document in the collection that can support a candidate answer. For that purpose, it looks for three sentence passages including an answer pattern used in the document retrieval, as well as a candidate answer. The search starts with the candidate answers and search patterns with better score.

3 Results

As in previous years we grouped the questions in a set of categories in order to get a better insight on which type of questions the system gets better results. For example, the category *Place* includes the questions where the system expects to have a location as answer such as the *Onde* (Where) questions and the *Que X* (Which X) questions where X=*cidade* (city) or X=*país* (country), in which the NER system is used to identify names of countries, cities or other locations in the retrieved documents. Other interesting categories are *People*, *Quantities* and *Dates* where the NER system is also used to find instances of those categories in the relevant texts. As the answer categories *Height*, *Duration*, *Area*, *Organization* and *Distance* are less represented in the question set, they are all grouped in the results presented in this paper.

Other categories are more pattern-oriented like the categories *Que é X* (What is X) or *Que|Qual X* (Which X).

Table 1 summarizes the results of the three official runs. Two runs which used the algorithm described in section 2 were sent in the PT-PT task. The only difference is that in Run 1 were considered word n-grams from length 1 to 3 while in Run 2, word n-grams from length 1 to 4 were used. The EN-PT run used word n-grams from length 1 to 3.

Type of question (after semantic analysis)		No. of Q. in 2006	No. (%) of correct answers			No. of Q. in 2005	No. (%) of correct Answers 2005	
			Run 1 PT-PT	Run 2 PT-PT	Run EN-PT		Best PT-PT Run	Run EN-PT
NER	People	29	9 (31%)	8 (28%)	2 (7%)	47	11 (23%)	5 (11%)
	Place	26	11 (42%)	10 (38%)	8 (31%)	33	9 (27%)	2 (6%)
	Date	20	3 (15%)	3 (15%)	1 (5%)	15	3 (20%)	2 (13%)
	Quantity	7	1 (14%)	1 (14%)	1 (14%)	16	4 (25%)	1 (6%)
	Height, Duration, Area, Organization, Distance	5	1 (20%)	1 (20%)	0	4	1 (25%)	0
n-grams	Que|Qual X [1]	60	11 (18%)	10 (17%)	9 (15%)	34	8 (24%)	6 (17%)
	Que é X [2]	36	10 (28%)	11 (31%)	4 (11%)	15	2 (13%)	0
	Quem é <HUM> [3]	9	3 (33%)	2 (22%)	4 (44%)	27	6 (22%)	6 (22%)
	Como de chama / Diga X [4]	8	1 (13%)	0	0	9	4 (44%)	2 (22%)
Total		200	50 (25%)	46 (23%)	29 (15%)	200	48 (24%)	24 (12%)

[1] Which X, [2] What is X, [3] Who is <HUM>, [4] What is X called / Name X

Table 1. Results by type of question

This year there was a large increase in the number of definition questions of type *Que é X?* and a considerable reduction in the number of questions of type *Quem é <HUM>?*. Theoretically this evolution would make Esfinge's task more complicated, as it was precisely for the questions of type *Que é X?* that the system obtained the worst results last year.

However, Esfinge managed to have even a slightly better result than last year in Run 1. This was accomplished through considerable improvements in the results with the aforementioned questions of type *Que é X?*, as well as with the answers of type *People* and *Place*.

In Run 2, Esfinge managed to answer correctly some questions for which it was not possible to have a complete answer in Run 1 (Run 1 used word n-grams up to length 3, so when the n-gram module was used, the maximum answer length was 3 and therefore it was not possible to give an exact answer to questions that required lengthier answers). However, as Run 2 used longer n-grams, 7 of the answers that were returned even though including the correct answer, included also some extra words. As these answers were classified as Inexact, the global results of Run 2 are slightly worse that the ones in Run 1. Not surprisingly, definitions of type *Que é X?* were the only type of questions with some improvement. If we consider the correct answers plus the answers including the correct answer and some more words then we obtain 51 answers in Run 1 and 53 answers in Run 2.

Regarding the EN-PT there was a slight improvement in the number of exact answers, even though only small details were adjusted in the automatic translation of some questions.

Table 2 summarizes the main causes for wrong answers. As in last year, most of the errors in the PT-PT task reside in the document retrieval. More than half of the errors occur because the system is not able to find relevant documents to answer the questions.

The other two main causes for errors are the answer scoring algorithm and the answer supporting module. The category Others includes among other causes: the answer needing more than four words (only word n-grams up to length four are considered), missing patterns to relate questions to answer categories, NER failures and the retrieved documents from the Web not including any answer.

Problem	No. of wrong answers		
	Run 1 PT-PT	Run 2 PT-PT	Run EN-PT
Translation	-	-	106 (62%)
No documents retrieved in the document collection	84 (56%)	84 (55%)	25 (15%)
Answer scoring algorithm	37 (24%)	30 (19%)	19 (11%)
Answer support	10 (7%)	11 (7%)	13 (8%)
Others	19 (13%)	29 (19%)	8 (5%)
Total	150	154	171

Table 2. Causes for wrong answers

Since in more than half of the questions Esfinge is not able to retrieve relevant documents and returns NIL, it is interesting to check the results that are obtained considering only the questions for which the system is able to find relevant documents. These results are summarized in table 3.

Type of question (after semantic analysis)		No. of answered (total) Q. in 2006 PT-PT	No. (%) of correct answers		No. of answered (total) Q. in 2006 EN-PT	No. (%) of correct answers
			Run 1 PT-PT	Run 2 PT-PT		Run EN-PT
	Place	13 (26)	6 (46%)	5 (38%)	7 (26)	2 (29%)
NER	People	12 (30)	8 (67%)	7 (58%)	4 (30)	1 (25%)
	Date	8 (20)	2 (25%)	2 (25%)	3 (20)	0
	Height, Duration, Area, Organiza-tion, Distance	5 (5)	1(20%)	1 (20%)	1 (5)	0
	Quantity	3 (7)	0	0	4 (7)	0
n-grams	Que é X	31 (36)	9 (29%)	10 (32%)	22 (36)	3 (14%)
	Que|Qual X	11 (60)	3 (27%)	2 (18%)	7 (60)	1 (14%)
	Quem é <HUM>	9 (9)	3 (33%)	2 (22%)	9 (9)	4 (44%)
	Como de chama / Diga X	7 (8)	1 (14%)	0	2 (8)	0
Total		99	33 (33%)	29 (29%)	59	11 (19%)

Table 3. Results for the questions for which the system finds relevant documents

Seen from this viewpoint, the results for the answers of type *People* and *Place* are even more encouraging. On the other hand, the values in parentheses (total number of questions) point out that the document retrieval for questions of type *Que|Qual X?* is particularly bad (the system only finds relevant documents for one sixth of these questions). Esfinge has also serious problems with the temporally restricted questions: it only finds relevant documents in the collection for 2 of the 20 temporally restricted questions and it does not answer any of these correctly.

One interesting issue is to determine whether the system is really improving its performance for a particular type of questions or whether the questions of a particular type are easier or harder from one year to the other. To get some insight on this issue, we tested this year's system with the questions from 2004 and 2005 as well, see Table 4.

Type of question (after semantic analysis)		No. of Q. in 2005	No. (%) of correct answers Esfinge 2006	No. of Q. in 2004	No. (%) of correct answers Esfinge 2006
NER	People	47	18 (38%)	43	20 (47%)
	Place	33	17 (52%)	41	21 (51%)
	Quantity	16	4 (25%)	18	3 (17%)
	Date	15	9 (60%)	15	6 (40%)
	Height, Duration, Area, Organization, Distance	4	1 (25%)	5	1 (20%)
n-grams	Que|Qual X	34	9 (26%)	42	8 (19%)
	Quem é <HUM>	27	9 (33%)	17	2 (12%)
	Que é X	15	2 (13%)	15	5 (33%)
	Como se chama/ Diga X	9	5 (56%)	3	1(33%)
Total		200	74 (37%)	199	67 (34%)

Table 4. Results with 2004 and 2005 questions

Even though there was a considerable improvement in this years performance with the definitions of type *Que é X?,* the same improvement is not visible with 2005 questions. This indicates that as more questions of this type were introduced in this year's edition, the global level of difficulty for this type of questions decreased and therefore there was not a real improvement of the system for this type of questions. On the other hand, the improvement in the questions of type *People* and *Place* is consistent with the results over the questions from the previous years, which confirms that Esfinge improved its performance with questions of these types.

4 Other experiments

In addition to the two submitted runs, some more experiments were performed. Table 5 gives an overview of the results obtained in these experiments. Run 3 used the same algorithm used in Run 1 except that it did not use the database of word co-occurrences BACO. Run 4 did not use the Web, only the document collection was used to extract the answers.

Type of question		No. of Q. in 2006	No. (%) of exact answers		
			Run 3	Run 4	Best in 2006
NER	People	29	7 (24%)	6 (21%)	9 (31%)
	Place	26	10 (38%)	10 (38%)	11 (42%)
	Date	20	3 (15%)	3 (15%)	3 (15%)
	Quantity	7	1 (14%)	1 (14%)	1 (14%)
	Height, Duration, Area, Organization, Distance	5	1 (20%)	0	1 (20%)
n-grams	Que|Qual X	60	11 (18%)	9 (15%)	11 (18%)
	Que é X	36	10 (28%)	3 (8%)	10 (28%)
	Quem é <HUM>	9	3 (33%)	0	3 (33%)
	Como se chama/ Diga X	8	1 (13%)	1 (13%)	1 (13%)
Total		200	47 (24%)	33 (17%)	50 (25%)

Table 5. Results of the non-official runs

From the results in Run 3, one can conclude that the use of the database of word co-occurrences to adjust the ranking of the candidate answers did not improve the results in a significant way, but it is somehow surprising that the questions correctly answered in the official run that were not answered correctly in Run 3 are precisely questions with answers of type *Person* and *Place*, obtained with the NER recognizer. One would expect that the use of BACO would be more useful with answers obtained using n-gram harvesting.

The results of Run 4 are quite interesting on the other hand. While the best run using the Web achieved an accuracy of 25%, this experiment with exactly the same algorithm except that it does not use the Web achieved only an accuracy of 17%. This is a bit surprising since in last year experiments the difference was not this large (24% using the Web and 22% with the document collection only as reported in [11]). Another issue concerns the fact that the document collection is getting older and therefore it should be more difficult to find a clear answer in the Web (which is constantly updated with new information) for some of the questions that one can make based in this collection. Questions like *Quem é o presidente de X?* (Who is the president of X?) or *Quem é <HUM>?* are examples of questions where the Web can give some steadily noisier information as the collection gets older. The fact that more definitions of type *Que é X?* were included this year at the expense of questions of type *Quem é <HUM>?* can explain this somehow surprising result in Run 4.

5 Evaluation guidelines

While analyzing Esfinge's results, I stumbled upon some interesting cases which may be worthwhile to discuss. I realized for example that the answers evaluated as incomplete (X) may include quite distinct situations:

- Literally incomplete answers (ex: *Internacional de Atletismo* instead of *Federação Internacional de Atletismo*).

- Answers that include the right answer and some more words that do not make the answer completely useless (ex: "*superficie de Marte*", "*Suécia, Thomas*" , "*Eric Cantona, ídolo*" where the exact answers are *Marte*, *Suécia* and *Eric Cantona*)

- Answers that contain more information than what was asked (ex: *abril de 1912 / April 1912* where the answer was considered wrong because only the year was asked). In the experiment with the 2005 questions there was another variation of this "error": the system answered *escritor peruano* (Peruvian writer) to the question *Qual a nacionalidade de Mario Vargas Llosa?* (What is the nationality of Mario Vargas Llosa?).

Whereas in my opinion the third type of answers should be evaluated as correct (there is nothing wrong with some additional (potentially useful) information, I think the first and second type of errors should be differentiated. The first type could be labeled as "Partially correct by Shortage" and the second type as "Partially correct by Excess" (this method was proposed in [12] and used in the NER Evaluation Contest HAREM [13] for example).

6 Concluding Remarks

I think that the QA task at CLEF is moving in the right direction. This year's edition had some novelties that I consider very positive. For example, the fact that in this year systems were required to provide document passages to support their answers, whereas in last year the document ID sufficed, makes the task more user-centered and realistic. It is not realistic to assume that a real user would be satisfied with an answer justification that consisted in a (sometimes) quite large document. Another positive innovation was not to provide the type of question to the systems.

This year results consisted in the refactoring of the code produced in previous years that now is available to the community in http://www.linguateca.pt/Esfinge/ . Regarding results of the runs themselves, the system managed to maintain the same level of performance even though questions were more difficult (and interesting) this year. It was also interesting to notice that the use of the Web as an auxiliary resource proved to be more crucial than in previous years. This might indicate that more information (and more reliable) is retrieved by Google and Yahoo in Portuguese, or that the questions this year were less time-dependent.

7 Future Work

The document retrieval in the document collection continues to be the main problem affecting Esfinge's performance in CLEF. To improve the results, it is necessary to improve the question reformulation module or/and the document retrieval module. One approach can be to incrementally simplify the queries, first removing the less important words until some possibly relevant documents are retrieved. Query expansion mechanisms (eventually using thesauri and/or ontologies) can provide some more sophisticated improvements in this area. The automatic extraction of patterns can also be an interesting working methodology. Still regarding the question reformulation module, an evaluation of the patterns currently in use can enable an updating of their scores.

Regarding the answer algorithm, as the results obtained using the database of co-occurrences did not improve the results as much as it could be expected, there might be some space for improvements in the way this resource is being used in Esfinge.

8 Acknowledgements

I thank Diana Santos for reviewing previous versions of this paper, Luís Sarmento for providing support for the use of both the NER system SIEMÊS and the database of word co-occurrences BACO.

This work has been produced in the scope of Linguateca, which is financed by the Portuguese Fundação para a Ciência e Tecnologia through grant POSI/PLP/43931/2001, co-financed by POSI.

References

1. Brill, E.: Processing Natural Language without Natural Language Processing. In: Gelbukh, A. (ed.): CICLing 2003. LNCS 2588. Springer-Verlag Berlin Heidelberg (2003) pp. 360-9
2. Gomes, D., Silva, M.J.: Characterizing a National Community Web ACM Transactions on Internet Technology (TOIT), volume 5, issue 3, pp. 508-531, August 2005.
3. Bick, E.: The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus: Aarhus University Press (2000).
4. 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.
5. Sarmento, L.: BACO - A large database of text and co-occurrences. In *Proceedings of LREC 2006* (Genoa, Italy, May 22-28, 2006).
6. Christ, O., Schulze, B.M., Hofmann, A. & Koenig, E.: The IMS Corpus Workbench: Corpus Query Processor (CQP): User's Manual. University of Stuttgart, March 8, 1999 (CQP V2.2)
7. Costa, L. & Sarmento L.: Component Evaluation in a Question Answering System. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odjik & Daniel Tapias (eds.), *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)* (Genoa, Italy, 22-28 May 2006), pp. 1520-1523.
8. Clarke, C. L. A., Cormack G. V. & Lynam T. R.: Exploiting Redundancy in Question Answering. In *Research and Development in Information Retrieval* (2001), pp. 358—365.
9. Cardoso N., Martins B., Gomes D. & Silva, M.J.: WPT 03: a primeira colecção pública proveniente de uma recolha da web portuguesa. In Diana Santos (ed.), *Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa*. IST Press, 2006.
10. 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 da Associação Portuguesa de Linguística (APL 2001) (Lisboa, 2-4 Outubro 2001). APL Lisboa (2002) pp. 485-495
11. Costa, L.: 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, Áustria, 21-23 September 2005), Springer. Lecture Notes in Computer Science 4022 , pp. 467 – 476. Revised Selected Papers.
12. Rocha P. & Santos D.: CLEF: Abrindo a porta à participa internacional em avaliação de RI do português. In Diana Santos (ed.), *Avaliação conjunta: um novo paradigma no processamento computacional da língua portuguesa*. IST Press, 2006.
13. Santos D., Seco N., Cardoso N. & Vilela R.: HAREM: An Advanced NER Evaluation Contest for Portuguese. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odjik & Daniel Tapias (eds.), *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)* (Genoa, Italy, 22-28 May 2006), pp. 1986-1991.