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

Luís Fernando Costa

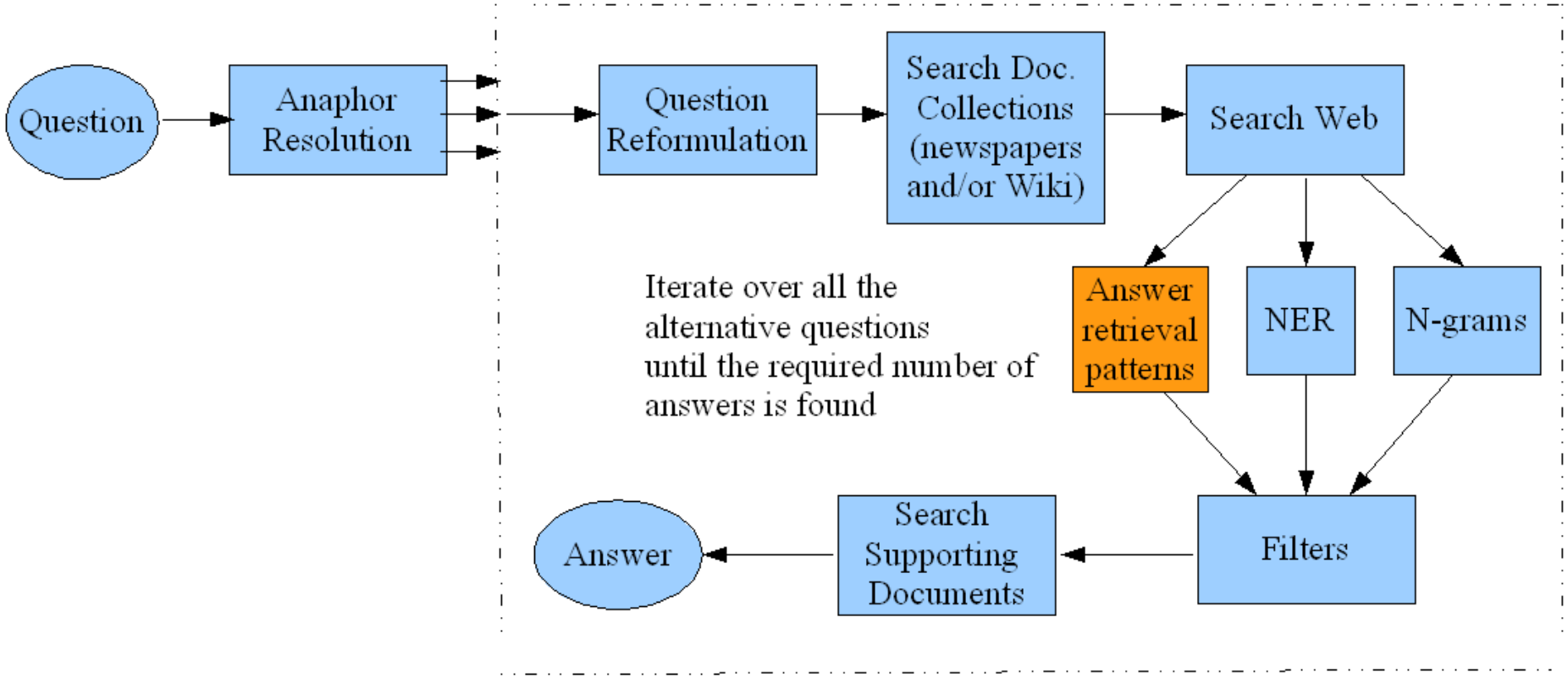
Outline

- Introduction
- Architecture of Esfinge
- Answer retrieval patterns
- Results
- Conclusions

Introduction

- General domain Portuguese QA system.
- Use of information redundancy to retrieve answers (Newspaper collection, Wikipedia and Web) (Brill, 2003).
- Esfinge uses a syntactic analyzer, a morphological analyzer and a named entity recognizer.
- Esfinge participates in CLEF since 2004.

Architecture of Esfinge



Anaphor Resolution

Uses the analysis of the PALAVRAS parser (Bick, 2000) 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 (Cabral, Costa & Santos, 2008) .

Example

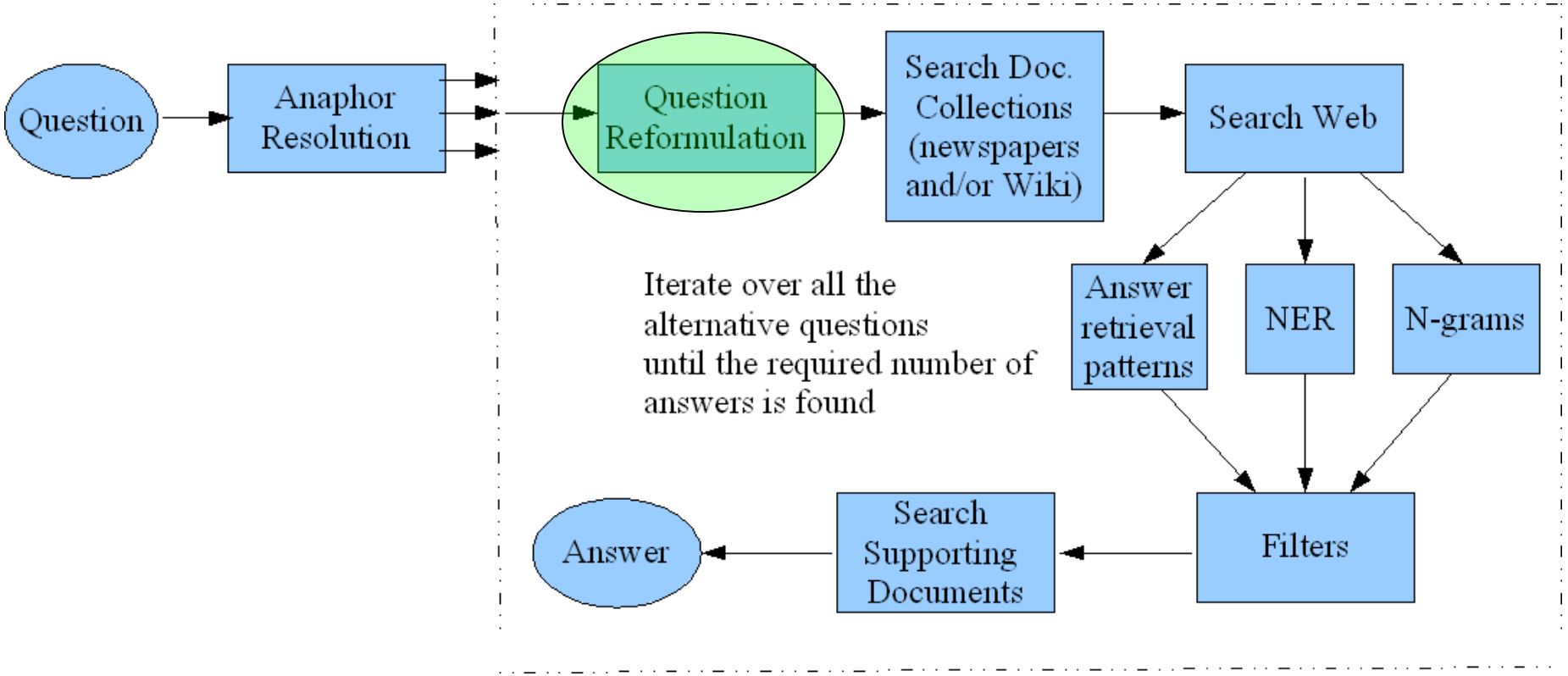
Q2: "Quem foi o criador de Tintin?"

Anaphor resolution candidates for the next questions: criador de Tintin , Tintin

Q3: "Quando é que ele foi criado?"

Anaphor resolved questions: "Quando é que o criador de Tintin foi criado?"
"Quando é que Tintin foi criado?"

Architecture of Esfinge



Question reformulation

Search patterns derived from the question.

Example

“Que país declarou a independência em 1291?”

1) String matching patterns:

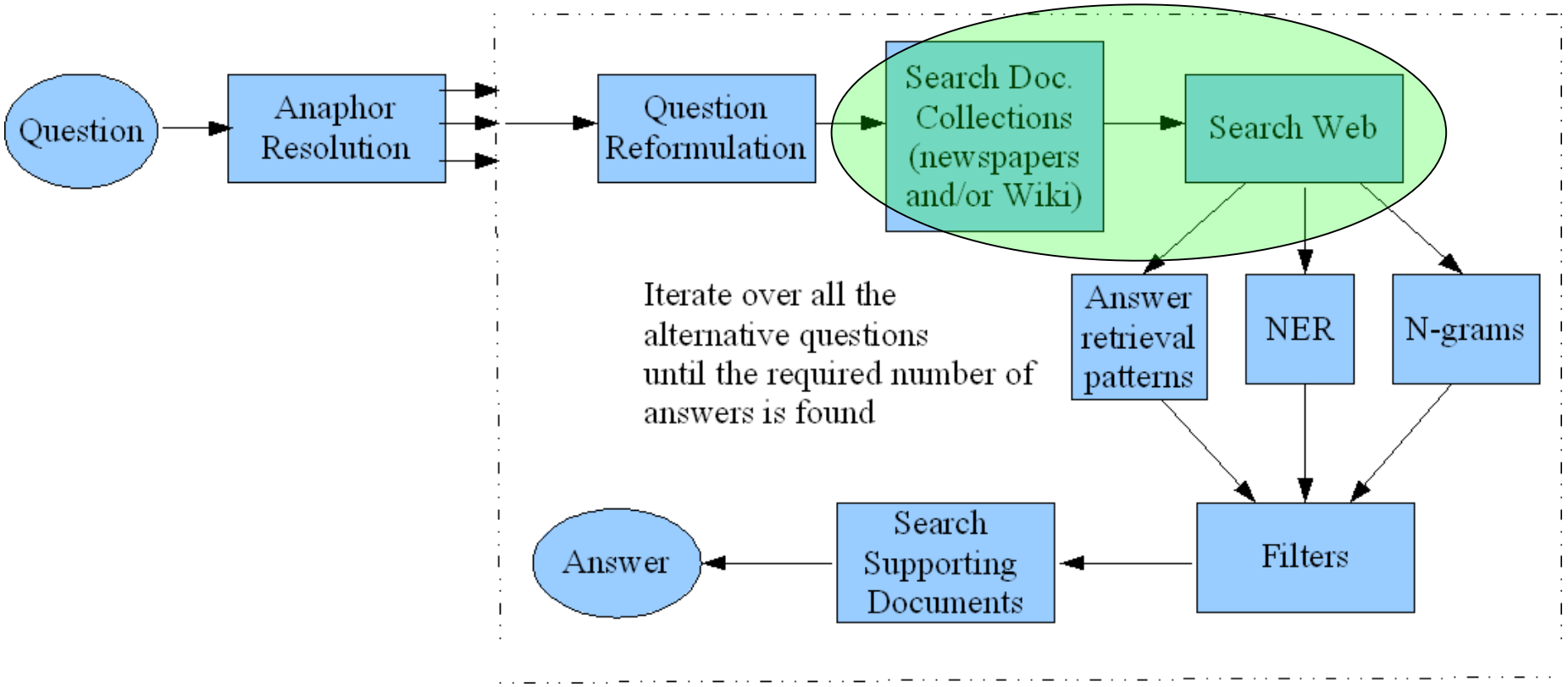
"declarou a independência em 1291“ país / score = 20
país declarou a independência em 1291 / score = 1

2) Patterns generated using the syntactic parser PALAVRAS:

declarou; a independência em 1291;
a independência em 1291; (without verbs)

Architecture of Esfinge

Search relevant text passages in the document collections & Web



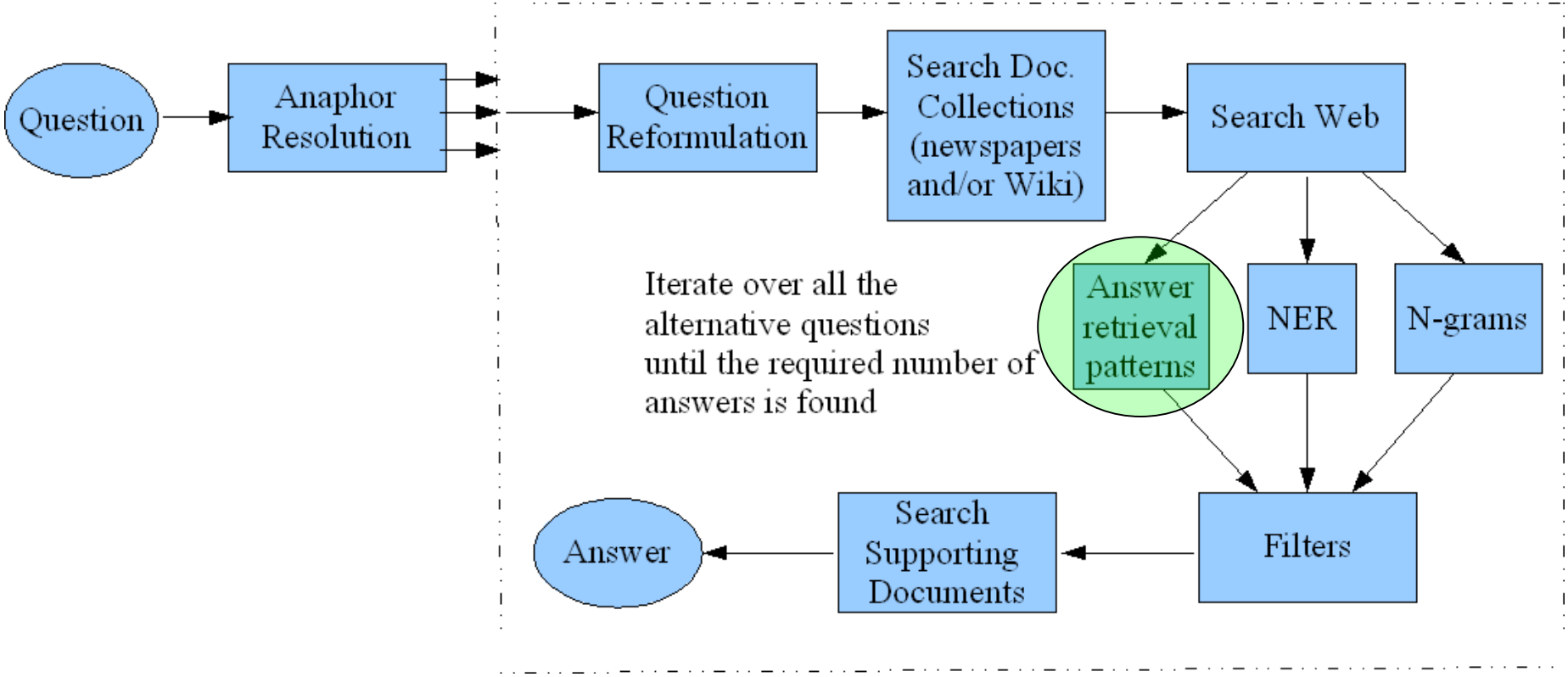
Retrieval of Relevant Text Passages

Search patterns are then used to query CLEF collections (newspaper text and Wikipedia) and a search engine (Yahoo API was used).

Result: Set of “relevant” text passages for the question

Architecture of Esfinge

Find candidate answers using answer retrieval patterns



Answer retrieval patterns

- Idea: use the solutions from previous years to extract (semi-)automatically answer retrieval patterns.

- Available resources: Solutions from QA@CLEF 2007.

How to find answer retrieval patterns?

<pergunta ano="**2007**" id_org="**X**" categoria="**D**" tipo="**OTHER**" restrição="**NO**" ling_orig="**PT**" tarefa_pt="**0023**" tópico="**020**">

<texto>**O que é a navegação de cabotagem?**</texto>

<resposta n="**1**" docid="**F950416-041**">**transporte entre portos do país**</resposta>

<extracto n="**1**" resposta_n="**1**">**O relator do tópico da reforma constitucional que trata da navegação de cabotagem (transporte entre portos do país),**</extracto>

</pergunta>

Question

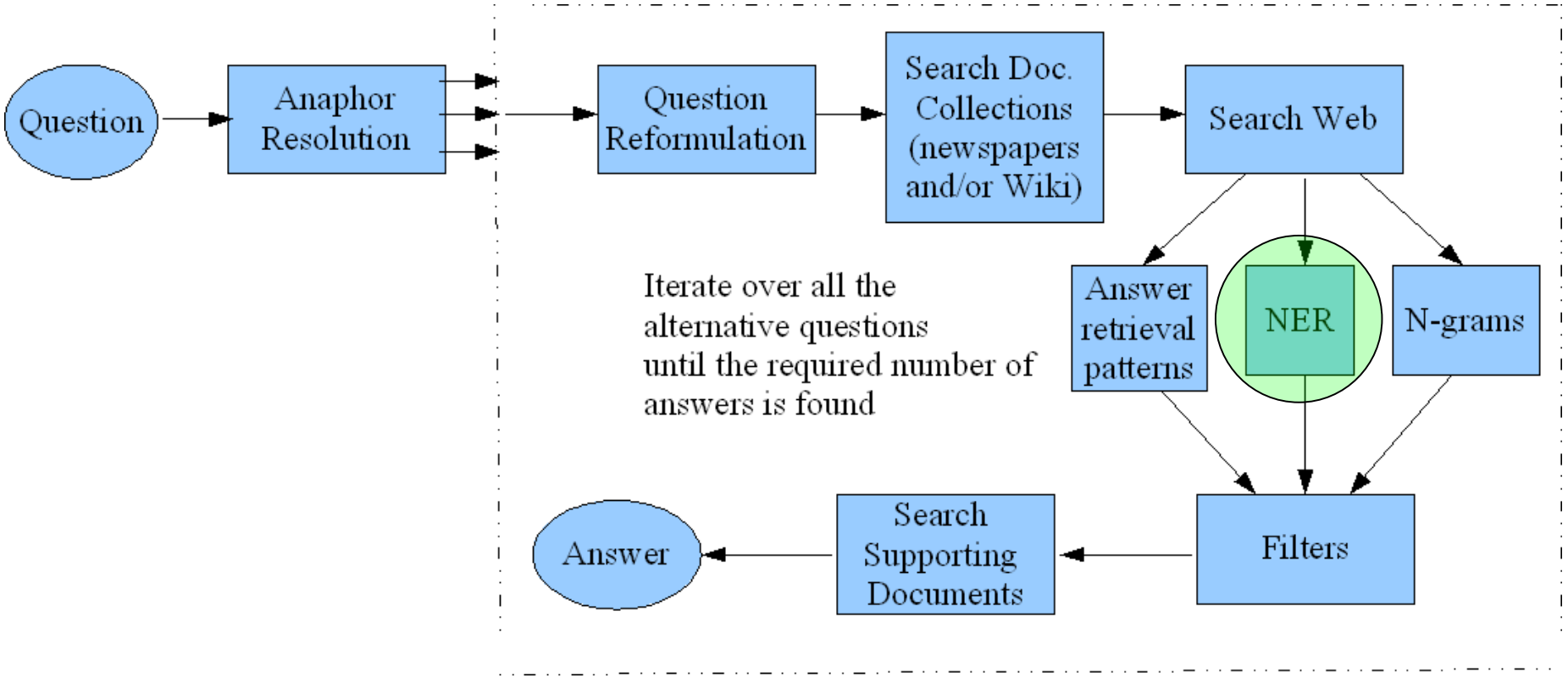
Answer

O que é a __X__? → __X__ (__ANSWER__)

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.

Architecture of Esfinge

Find candidate answers using named entity recognition



Answer extraction and ranking using a named entity (NE) recognition system

Uses the analysis of the NER system SIEMÊS (Sarmento, 2006)

Questions can give an hint about the answer type.

Quando *viveu Franz Liszt? => (period of time),*
Onde *é a sede da Interpol? => (place)*

Quem é o presidente da França? => (human)

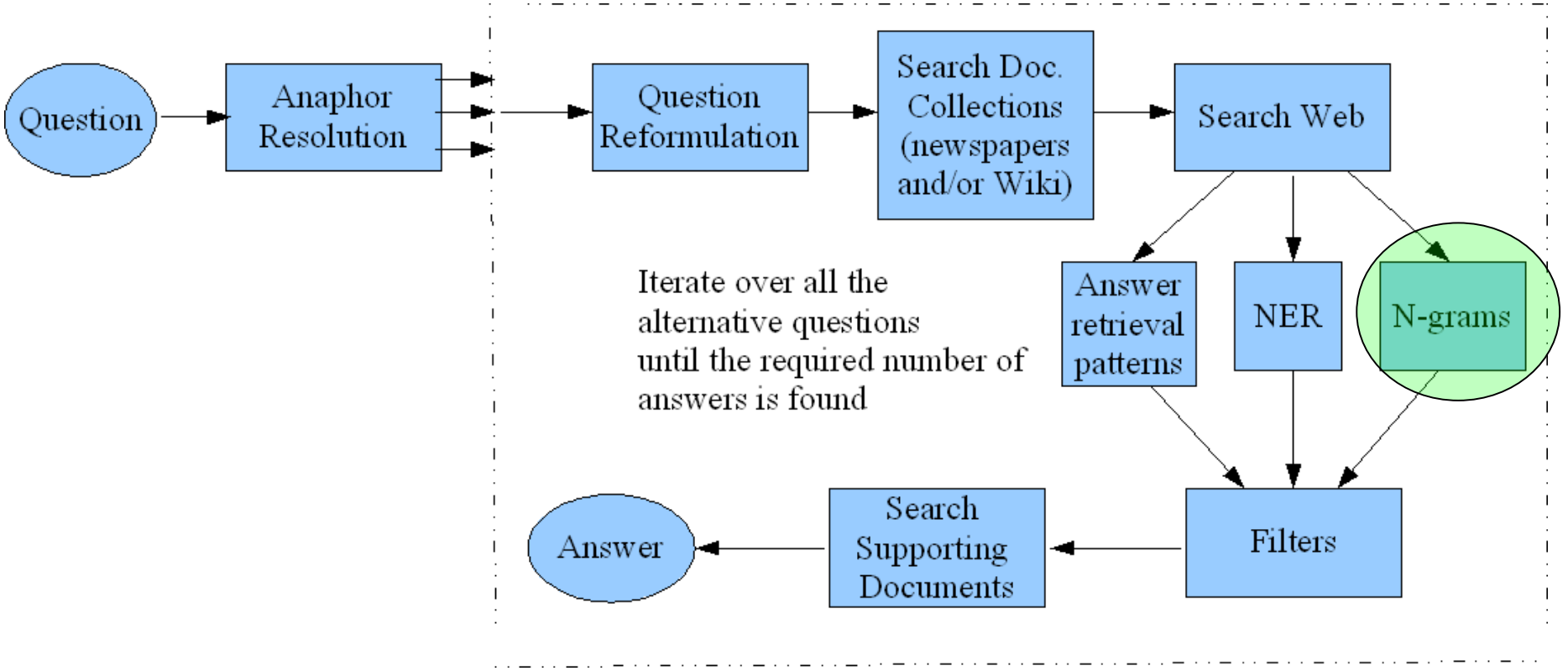
"No dia 14 de setembro de 2004 o presidente da França, Jacques Chirac visitou a Finlândia"

No dia <TEMPO TIPO="DATA">14 de setembro de 2004</TEMPO> o <SERES TIPO="CARGO">presidente da França</SERES> , <SER TIPO="HUM">**Jacques Chirac**</SER> visitou a <LOC TIPO="PAIS">Finlândia</LOC>

NE score = ∑ (NE frequency * Passage score * NE lenght)

Architecture of Esfinge

Find candidate answers using n-gram harvesting



Answer extraction and ranking using n-gram harvesting

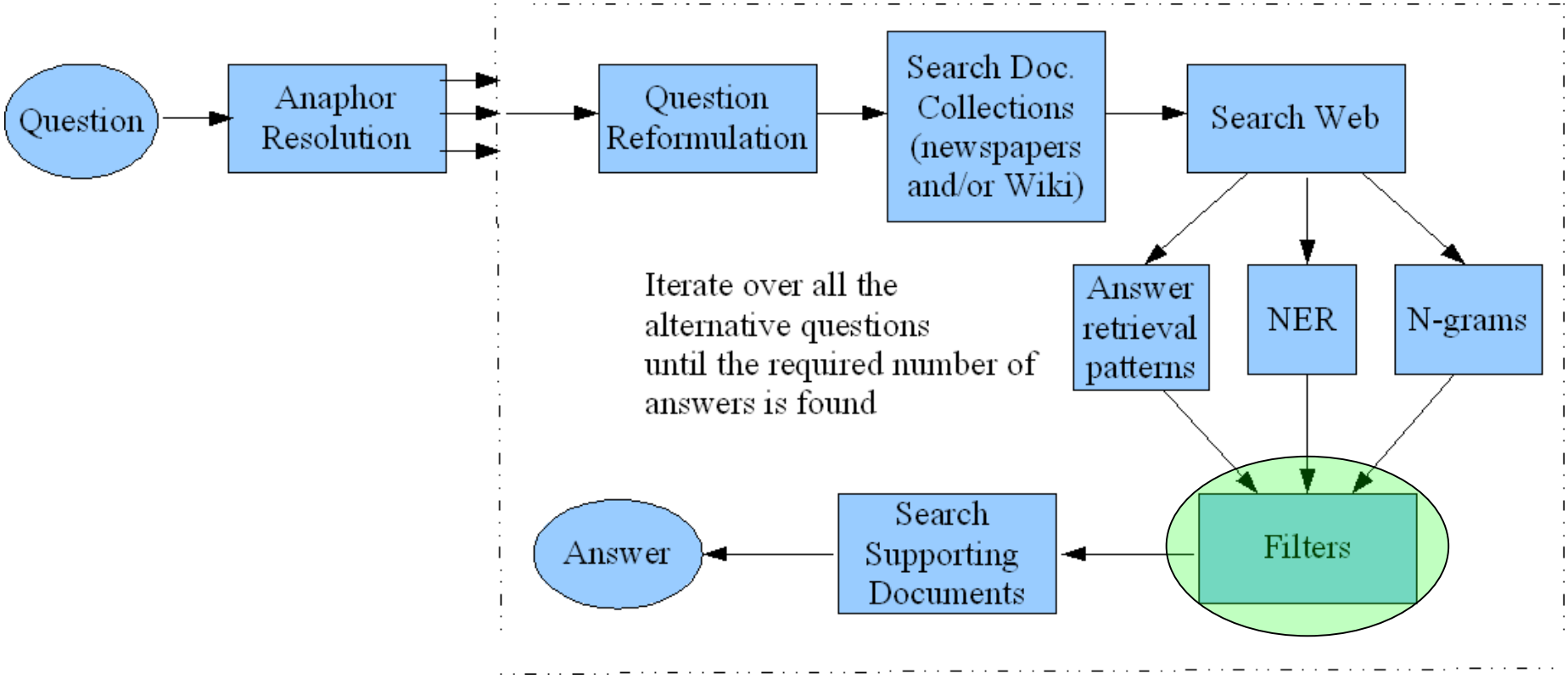
"No dia 14 de setembro de 2004 **o presidente** da França, Jacques Chirac visitou a Finlândia. Teve uma breve reunião com **o presidente** finlandês."

N-gram	Frequency
o presidente	2
presidente	2
No dia 14	1
dia 14	1
dia 14 de	1

Ng score = ∑ (Ng frequency * Passage score * Ng lenght)

Architecture of Esfinge

Filtering of candidate answers

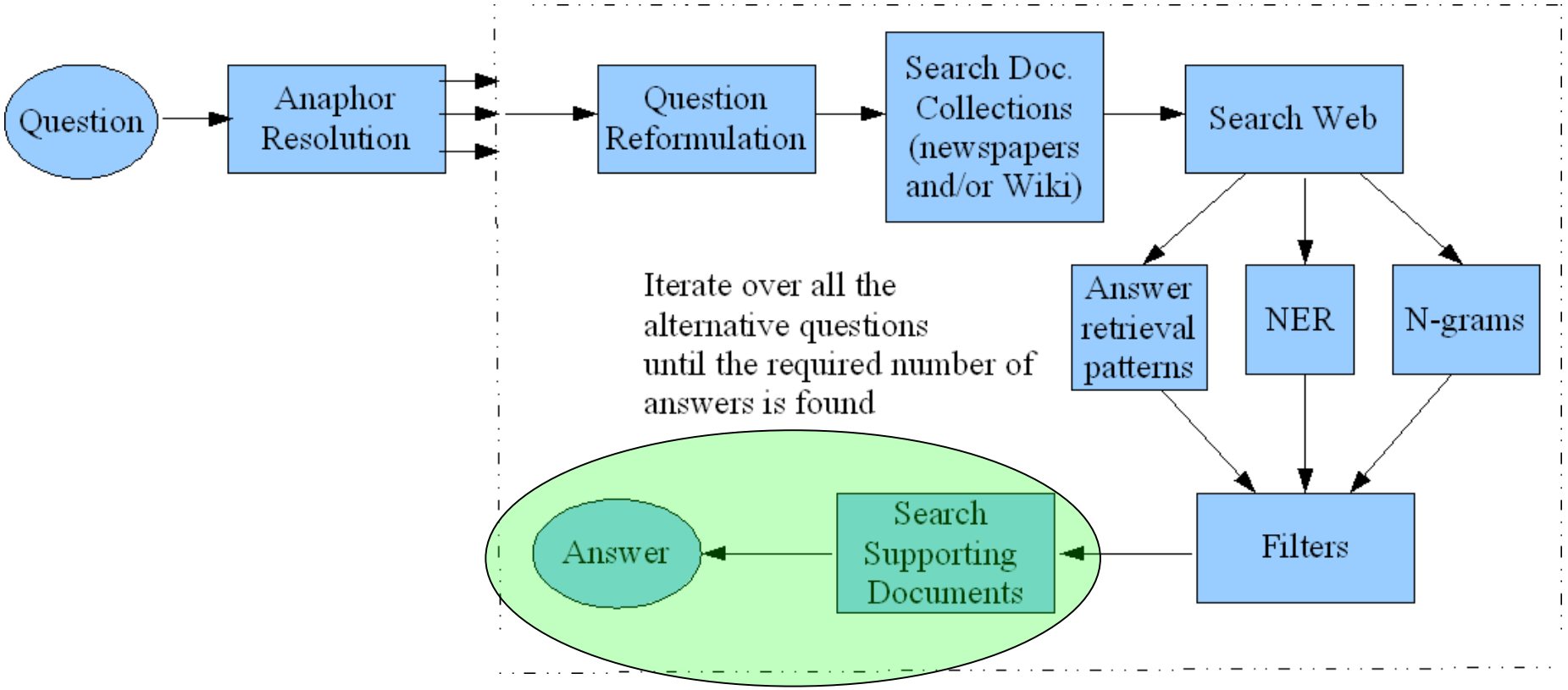


Answer filtering

- Filter candidate answers:
  - Contained in the question:
    - Ex: *Quem era o pai de Isabel II? (Isabel II is not a good answer)*
  - Based on the part of speech of the words in the answer (answers which first and final token are not adjectives, common nouns, numbers or proper nouns are filtered). Morphological analyzer jspell (Simões & Almeida, 2001) used to check PoS.
    - Ex: o presidente , o , presidente (OK) , No dia 14, No dia, dia 14 (OK)

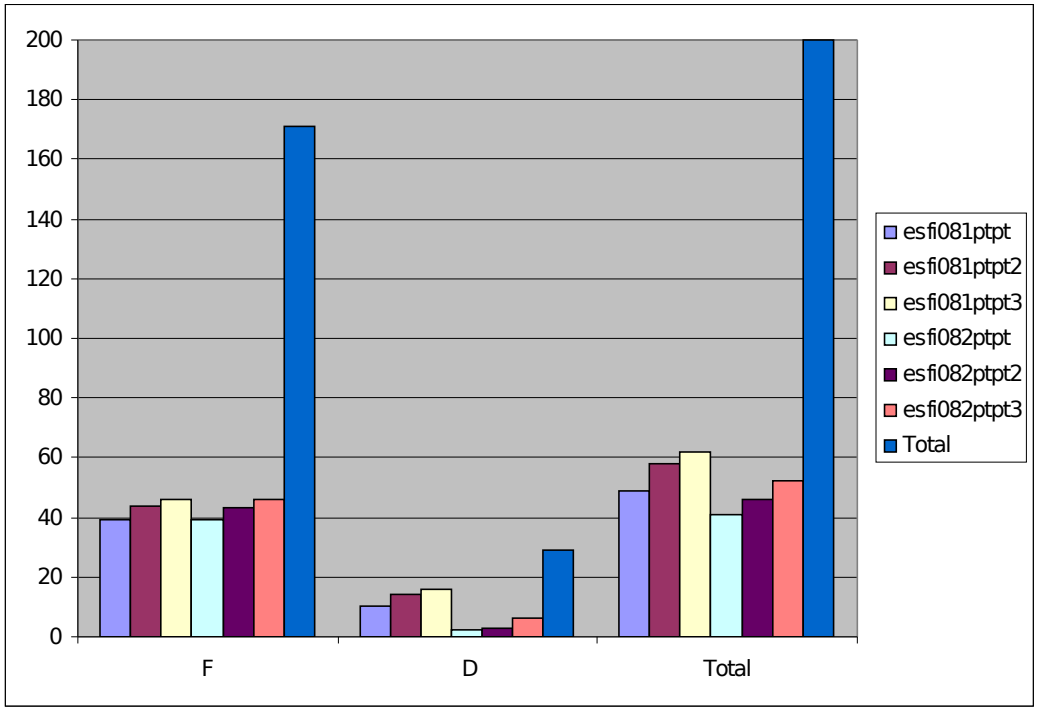
Architecture of Esfinge

Search relevant text passages in the document collections & Web



Results

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



Conclusions

- 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.

- 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.

Future work

- More detailed error analysis.
- Update the publicly available distribution at:

http://www.linguateca.pt/Esfinge/

- We can derive more answer retrieval patterns and improve results, but to improve them significantly, Esfinge probably needs to use more semantics: dictionaries and/or ontologies.

References

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.

Eckhard Bick. The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus: Aarhus University Press (2000).

Eric Brill. Processing Natural Language without Natural Language Processing. In: Gelbukh, A. (ed.): CICLing 2003. LNCS 2588. Springer-Verlag Berlin Heidelberg (2003) pp. 360-9.

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.

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.

Acknowledgments

- This work was done in the scope of the Linguateca, contract nº339/1.3/C/NAC, project jointly funded by the Portuguese Government and the European União Europeia FEDER/FSE