

Making RAPOSA (FOX) smarter

Luís Sarmiento and Eugénio Oliveira
Faculdade de Engenharia da Universidade do Porto
las@fe.up.pt eco@fe.up.pt

Abstract

In this paper we describe RAPOSA, a question answering system for Portuguese, and we present detailed results of the participation in the QA@CLEF 2007 evaluation task. We explain how we improved our system since the last participation in QA@CLEF, by expanding the set of rules associated with Question Parsing and Answering Extraction. We will finish by pointing lines for future work, and by introducing the concept of “tagging by example” that allows improving the answer extraction stage by learning how identify in text elements that are potentially useful in answering certain types of questions.

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, Questions beyond factoids

1 Introduction and Motivation

The main goal of our participation in QA@CLEF for Portuguese with the RAPOSA system was to be able to gain further insights into the problem of semantic analysis of text over a realistic and application-oriented such as is the Question Answering (QA) scenario. Our previous participation in the QA track [9] (our first) was mainly intended to be an alternate evaluation scenario for our wide-scope named-entity recognizer (NER) system, SIEMÊS [6], which was, and still is, the main analysis component of RAPOSA: both analysis of questions and extraction of answers rely heavily in the NER system. However, in last year’s participation RAPOSA suffered from several limitations in specific point of its pipeline that degraded the final performance of the system, in levels sometimes superior to those that resulted from problems in the NER system itself. Therefore, our participation in 2006, although successful in enabling us to detect interesting problems in our NER system, and in helping us to define new directions for future work, lacked the representativeness that we wished for testing our NER.

The main concern this year was to expand the set of different types questions that the system attempts to answer, by improving both the question parsing and the answer extraction stages, where the role of NER system is crucial. However, there were two challenges that we decided not

to face yet this year. First, we did not attempt to answer list/enumeration questions (10 in this year's test set). Additionally, we did not try to answer context dependent questions because no co-reference resolution module was available for that purpose. When a group of questions was found in the test set, only the first question of the group, which had no resolution dependencies, was considered. The co-reference resolution problem is still out of the scope of our current work.

Another interesting point in this years participation was related with using Wikipedia as the source collection for extracting answers. There has been a great interest recently in using Wikipedia for all sorts of information extraction tasks, and given the continuous growth of this resource, the trend is that such type of works become even more common.

2 RAPOSA

RAPOSA is a question answering system for Portuguese (mono-lingual). Contrary to many other systems that clearly separate the information extraction stage (where facts are extracted from text bases and stored in some intermediate fact database) and retrieval stage (where, given a question, the system retrieves candidate answers from the fact database and generates an answer), RAPOSA tries to provide continuous on-line processing chain from the question to the answer.

There are three important advantages in following such an approach. First, during development time, any change in the RAPOSA can be immediately tested, making it possible to speed-up the entire development process. Second, and since the focus is on direct processing of text information, adapting RAPOSA to answer questions based on a different text collection is simply a matter of developing the interface for querying the new text sources. And third, various question-answering heuristics for specific types of questions can be implemented and used almost instantaneously, making it very simple to expand the set of question that RAPOSA can answer.

The downside of this approach, however, is that it usually takes much more time to find a possible answer than what would be acceptable for a real-time system. For this reason we are considering in future versions of RAPOSA, to develop an hybrid system which relies on previously semantically-tagged text to try to answer a question, only using plain text sources (which will be annotated on demand) whenever the answer is not found by searching the tagged text.

The architecture of RAPOSA is currently a pipeline consisting of 6 blocks (see figure 1).

1. **Question Parser:** receives a question in raw text, identifies the type of question, its arguments, possible restrictions and other relevant keywords, and transforms it into a canonical form. The admissible types of answer for the question are also identified at this stage. The question parser uses the SIEMÊS NER for identifying mentions to named-entities and other related phrases in the question.
2. **Query Generator:** this module is responsible for preparing a set of query objects from the previously parsed question so that the following Snippet Searcher module can invoke the required search engines. The Query Generator selects which words must necessarily occur in target text snippets (for examples, a name that was identified by the Question Parser), and which words are optional. Some query expansion techniques could be applied in this module, but currently only very simple regular expression generalizations are made (stripping suffixes and changing them by wild-cards).
3. **Snippet Searcher:** the Snippet Searcher uses the information given by the Query Generator for querying the available text bases and retrieving text snippets where candidate answer may eventually be found. The collections that RAPOSA is currently able to query are the CLEF collection for Portuguese, a MySQL encoded version of the XML dump of the portuguese Wikipedia ¹, and the BACO [7] text base (we plan to develop an interface to web search engines soon). After being retrieved, all text snippets are annotated by SIEMÊS (NER and related phrases).

¹Available from the University of Amsterdam: <http://ilps.science.uva.nl/WikiXML>

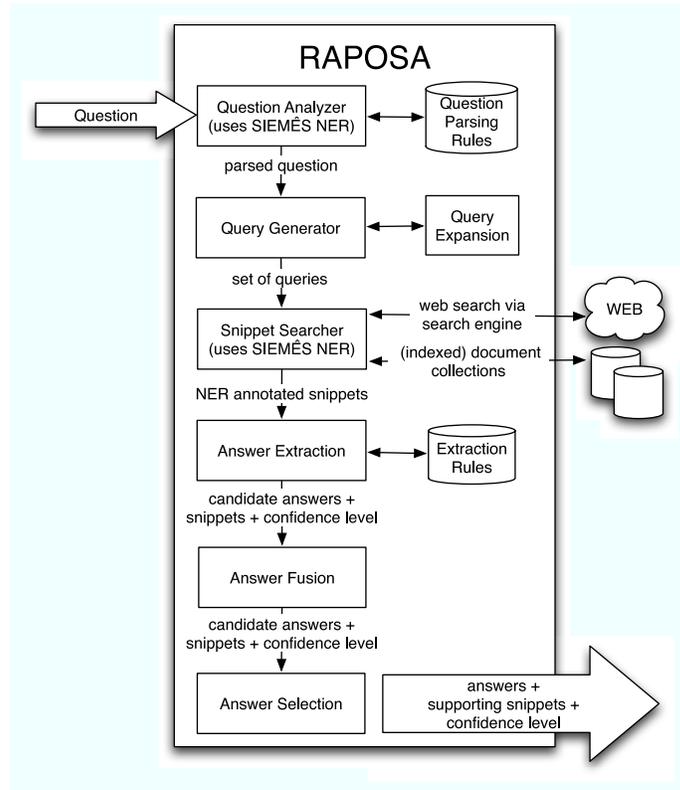


Figure 1: The basic components of RAPOSA

4. **Answer Extractor:** The Answer Extractor takes as input (i) the question in canonical form and (ii) the list of NER annotated snippets given by the Snippet Searcher and tries to identify candidate answers in them.

The Answer Extractor has two possible strategies to find candidate answers. The first one is based on a set of *context evaluation rules*. These rules check the presence of certain specific contexts (words or semantic tags) around the position of the argument of the question (note that the argument of the question must be present in the text snippets retrieved by the Snippet Extractor), and extracts specific elements for being considered as candidate answers. Context evaluation rules lead to high precision results, but they are usually too restrictive, which affects the recall. These rules have been developed manually, which is a severe limitation on the scalability of this strategy.

For this reason we developed a second strategy for answer extraction is much simpler: to consider as possibly answer any element found in the annotated text snippets that is semantically compatible to the expected semantic category of the answer for question at stake (as identified by the Question Parser). Although more than one compatible element may exist in the snippet being analyzed, this strategy can actually lead to good results. provided that there is enough redundancy in the answer. In such cases, the most frequently found candidate in all the analyzed text-snippets has a good chance of being the correct answer. We call this the *simple type checking strategy*.

5. **Answer Fusion:** the role of the Answer Fusion module is to cluster lexically different but semantically equivalent (or overlapping) answers in to a single “answer group”. At this moment, this module is not yet developed: it simply outputs what it receives from the Answer Extractor.

6. **Answer Selector:** selects one of the candidate answers produced by the Answer Fusion module and chooses the best supporting text / answer justification among previously extracted text snippets. Finally, the Answer Selector also assigns a confidence value to that answer.

There are other question answering systems for Portuguese that follow either a similar shallow parsing and on-line approach [2], or that make use of more deeper parsing technologies and knowledge bases ([1] and [5]).

3 Improving RAPOSA

Last year, the Question Analyzer had a set of 25 rules to analyze the type of question and to identify its elements. These rules were able to address only a few types of questions, mostly (i) factoid questions about dates, places, and quantities, and (ii) questions about people, both factoid and simple definition question. This year, a significant amount of work was invested in improving and creating rules for question parsing, both for definition and for factoid question.

We manually extended the set of rules for processing factoid questions to 72 rules. RAPOSA is now able to process factoid questions for which answers can be dates, several type of numeric values (quantities, dimensions and money) or entities (location, organization, geo-political entities, people, facilities and infrastructures, or other undefined entities). Also, the set of rules to process definition questions was expanded to 30 rules to deal with definitions regarding people, acronyms and miscellaneous objects. Additionally, we added some chunking capabilities to the Question Analyzer module, that allow the identification of some compound words and specific noun phrases, prior to the actual parsing of the question. This enables RAPOSA to deal with questions that do not directly mention named-entities.

The Answer Extraction module was also improved with the addition of several new extraction rules. Last year, RAPOSA had 25 context evaluation rules for dealing with question of the type “Who is < job title >?” and 6 to deal with questions of the type “Who is < person name >?”. For all other question addressed last year we would extract answer candidates using the simple type checking strategy. This year, we were able to make some minor corrections on the context evaluation rule that we already had for “Who is...?” questions, and we developed additional context evaluation rules for dealing with acronyms (15 rules), organizations (2 rules), miscellaneous definitions (9 rules). For the factoid questions we kept the same type checking strategy that we followed last year. The only difference is that we developed more associations between questions and answer types because the Question Parser is now able to deal with more types of factoid questions.

The final improvement made this year was the possibility of searching Wikipedia text. We converted the XML version of the portuguese Wikipedia into a tabular format which was then loaded into a MySQL database. From all the information that is provided in the XML dumps only the text contained in the paragraphs of Wikipedia articles was kept, i.e. links, category information, related information boxes were ignored.

4 Results

We submitted for evaluation only one run, for the monolingual Portuguese question set. The number of questions that RAPOSA attempted to answer was less than the 200 questions included in the test set. As mentioned before we skipped all list/enumeration questions and we did not try to answer dependent question since we had no method for tackling the co-reference problem. In fact, for any group of dependent questions RAPOSA only tried to parse the first question: the rest of the questions were ignored, which immediately excluded 50 dependent questions. From the remaining question RAPOSA was only able to parse 107 questions. The unparsed questions received the NIL answer, which was almost always the incorrect answer.

We are mainly concerned in analyzing the performance of RAPOSA over the 107 question that it was able to parse, because those are the ones for the whole information extraction pipeline was

activated. We will divide the analysis of results in two sections, which correspond in practice to the two types of strategies that were used for extracting answer candidates: (i) definitions questions (29), for which we used *context evaluation rules*, and (ii) factoid questions (78) for which we used the *simple type checking strategy*.

Also, for several reasons, as explained in [8], the Snippet Extractor of RAPOSA was frequently unable to find text snippets for extracting possible answers. The text search capabilities of RAPOSA are still in a very early stage of development, and they are basically not more than simple SQL queries over MySQL encoded text databases. Thus, many of the answers produced by RAPOSA are NIL (48 %) for the simple reason that it was not able to find any text snippet. This issue will be matter of future work, so for now we will focus mainly in the precision of RAPOSA when it is able to find an answer (i.e. answer is not NIL). The results we present in the next tables will address the following parameters for each type of question:

- #: number of question that RAPOSA attempted to answer
- T: number of question for which a Text answers was found (i.e. not NIL)
- NIL: number of question answered NIL
- R, X, W: Right, ineXact and Wrong answers
- R | T: Right questions given that a Text answered was provided

Table 1 on presents the aggregated results over all the question. The last two columns contain the values of the overall precision (right answers / total questions attempted) and the precision when a non-NIL answer was produced. In the two following sections we will detail the results for the case of definition questions (DEF) and factoid questions (F).

type	#	T	NIL	%NIL	R	X	W	R T	P_{all}	$P_{R T}$
DEF + F	107	56	51	48.0	35	2	71	30	0.33	0.54

Table 1: Global result for the 107 questions that RAPOSA was able to parse.

4.1 Definitions Questions

We consider *Definition Questions* all questions for which the answer is a phrase or a sentence that provides an explanation about a named-entity or concept. These include questions about:

- **people** of the form “Who is <person name>?”, such as “Quem é George Vassiliou?” / “Who is George Vassiliou?” . We will refer to such questions by DEF_HUM.
- **acronyms** (similar to the acronym expansion task) of the form “What is <acronym>?” such as “O que é a FIDE?” / “What is FIDE?”. These questions will be identified by DEF_ACR.
- **miscellaneous objects** of the form “What is <something>?” such as “O que é a Granja do Torto?” / “What is Granja do Torto?”. We will refer to this questions as DEF_MISC.

RAPOSA successfully parsed and attempted to answer 29 definition questions. There were 31 definition question in this year’s test set, so RAPOSA was able to parse nearly all of them. Table 2 (similar to Table 1) presents an overview of the results obtained.

Since RAPOSA could use either Wikipedia or the Portuguese CLEF corpus for extracting the answer, it is also interesting to check how many times each of them was actually used for that purpose. For the 16 questions for which RAPOSA found 7 non-nil answer in Wikipedia a 9 non-nil answers in CLEF corpus.

type	#	T	NIL	%NIL	R	X	W	R/T	P_{all}	$P_{R T}$
DEF_HUM	8	8	0	0.0	7	1	0	7	0.88	0.88
DEF_MISC	18	6	12	67	7	0	11	6	0.39	1.00
DEF_ACR	3	2	1	33	2	0	1	2	0.67	1.00
Σ	29	16	13	45	16	1	12	15	0.59	0.94

Table 2: The performance of RAPOSA in answering 29 Definition Question

4.2 Factoid Questions

Factoid questions include all sorts of question for which the answer is a simple factual elements, such the name of a named-entity (person, organization, location, etc), a date, a numerical value, etc. In this years test set there where 159 questions that were classified as factoid questions. RAPOSA specializes the general category of factoid questions into different sub-categories according to the type of answers that they require:

- person (F_PER): the correct answer will be a name of a person (real or fictional) or a character, or in some cases, a given group (such as the name of a rock band). E.g.: “Quem foi o último rei da Bulgária?” / “Who was the last King of Bulgaria?”
- geo-political entities (F_GPE): these questions have as admissible answers the names of geo political entities such as countries, states, cities, etc. E.g.: “Qual é a capital de Timor Ocidental?” / “What is the capital of East Timor?”
- organization (F_ORG): the answer for these questions must the name of an organization E.g.: “A que partido pertencia Spadolini?” / “Which party did Spadolini belong to?”
- entity (F_ENT): this is a more general case of the previous question where the type of the entity that corresponds to the correct answer is not very well defined (or requires deeper semantics analysis for its definition) and can be any either a person, organization or a GPE.
- location (F_LOC): these questions have as an answer the name of a geographic location. E.g. “Onde é que nasceu o Vítor Baía?” / “Where was Vítor Baía born?”
- infra-structures (F_INFRA): the expected answer is a reference to a infra-structures, such as a brigde, a road, etc. E.g.: “Qual o túnel ferroviário mais comprido do mundo?” / “Which is the world’s longest train tunnel?”
- date / time (F_DATE): the expected answer is a date, a specific time or a time period. E.g. “Em que ano foi fundada a Bertrand?” / “In which year was Bertrand established?”
- dimensions (F_DIM): the answer must be a value along with an measurement unit. E.g. “Qual o diâmetro de Ceres?” / “What is the diameter of Ceres?”
- F_MNY: the answer for these questions is an amount of money. E.g. “Quanto custou a Mars Observer?” / “What as the cost of the Mars Observer?”
- F_NUM: the expected answer is simply a number. E.g.: “Quantas ilhas tem Cabo Verde?” / “How many islands does Cabo Verde have?”
- F_RVAL: the type of answer for this question will be a relative value such as a ratio or a percentage. E.g.: “Que percentagem dos finlandeses fala sueco?” / “What is the percentage of Finnish that can speak Swedish?”

Table 3 (similar to previous) contains the results by type of factoid question. RAPOSA extracted most of his non-nil answers from Wikipedia - 35 answers - while the CLEF corpus was only used for extracting 5 answers.

type	#	T	NIL	%NIL	R	X	W	R T	P_{all}	P_{RT}
F_DATA	15	9	6	40	5	1	9	4	0.33	0.44
F_GPE	15	9	6	40	6	0	9	6	0.40	0.67
F_ORG	6	1	5	83	1	0	5	0	0.17	0.00
F_HUM	7	4	3	43	1	0	6	1	0.14	0.25
F_ENT	9	3	6	67	1	0	8	1	0.11	0.33
F_INFRA	2	0	2	100	0	0	2	0	0.00	-
F_LOC	4	2	2	50	2	0	2	1	0.50	0.50
F_DIM	5	4	1	20	0	0	5	0	0.00	0.00
F_MNY	1	1	0	00	0	0	1	0	0.00	0.00
F_NUM	11	7	4	36	2	0	9	2	0.18	0.29
F_RVEL	3	0	3	100	0	0	3	0	0.00	-
Σ	78	40	38	49	18	1	59	15	0.23	0.38

Table 3: The performance of RAPOSA in answering 78 Definition Question

5 Discussion of the Results

5.1 Definitions Questions

Considering first the definitions questions (31 in the test set), we can see that RAPOSA had a reasonably good performance, both from a recall and from a precision point of view. The most remarkable result is the extremely high precision of RAPOSA, in the cases where a non-nil answer is produced. This indicates that current context evaluation rules, when successful in extracting an answer, provide almost always the correct answer.

However, in about 45% of cases no candidate answer was extracted, and hence a NIL answer was produced. An incorrect NIL answer can occur in two cases: (i) when no snippet was found from which we could extract an answer candidate, or (ii) when the context evaluation rules were not able to match any of the snippets found. As explained before problem (i) will be subject of future work, and we will now focus on problem (ii). Since most of the incorrect nil answers were produced while attempting to answer miscellaneous definition questions, for which we only had 9 evaluation rules and a small related vocabulary, we have a strong indication that the problem here is the lack of enough context evaluation rules, which results in the low recall values for these cases. Clearly, there is a huge room for improvement by simply adding more evaluation rules.

5.2 Factoid Questions

The precision obtained in answering factoid question was significantly inferior to that obtained definition questions. One immediate reason for this is related to how answers are extracted from retrieved text snippets, i.e. by using the simple type checking mechanism. While such a relaxed strategy seems to work slightly better for dates, GPE and location factoids, it has clearly very low performance for the other types of questions, and seems especially bad in dealing with numeric factoids. Also surprising, but compatible with last years results was the relatively low performance of RAPOSA for factoid questions regarding people.

One recurrent problem in answering factoid questions using the type checking strategy is that it is relatively frequent to find in a given text snippet more than one element (named-entity / numerical item) compatible with the type we are looking for. When there is enough redundancy in data (i.e. we may find more than one different text snippet that can provide us the answer), we might still be able to extract the correct element just by checking the most frequent one over all retrieved snippets. This works relatively well in many occasions, and RAPOSA has been relying on this so far. But the chances of choosing the correct answer decrease when there is less text available related to the question at stake. For less frequently documented subjects, for which less text snippets can be found, we end up choosing more or less randomly and obtaining many

incorrect answers.

This seems to be specially the case for certain factoid question that focus on numeric facts, either because these facts are usually related to very specific subjects (e.g. technical data) with less text available for search, or because there are many distinct ways of expressing the same (or approximate) numeric quantity. We believe that these two factor combined make it quite difficult to extract the correct answer in these cases.

5.3 Comparison with 2006

From a global point of view, we were able to significantly improve the results obtained last year both in recall and in precision. In 2006, we were submitted two rounds, $R1_{2006}$ that only had answers extracted using the context evaluation rules, and $R2_{2006}$ which, for several types of factoid questions, we also used the simple type checking mechanism to extract answers.

One of the most important improvements this year is that we greatly increased the number of different types of questions that RAPOSA can now try to answer. Last year, in round $R1_{2006}$ RAPOSA was only able to answer 34 questions, because it only had context evaluation rules for extracting answers for “Who is < job title >?” and “Who is < person name >?” questions. For $R2_{2006}$, RAPOSA was configured also to extract answer for factoid question using the simple type checking strategy, so it attempted to answer 74 questions. This year, mostly because of the work invested in (i) creating new question parsing rules for several types of factoid questions (which involves being able to assign one or more possible answer types to each type of parsed question) and (ii) creating rules to parse more definition questions (namely acronyms and miscellaneous), we were able to greatly increase the number of question that RAPOSA tried to answer (109 against 79).

If we consider questions for which RAPOSA used context evaluation rules (i.e. comparing current results regarding definition question), we might be lead to think that there was no improvement at all because last in year’s $R1_{2006}$ run RAPOSA attempted two answer 34 question while this year RAPOSA only attempted to answer 29. However, in this year’s question set there was a huge reduction in the “Who is < person name >?” questions, which were the majority of definition questions in last year’s set. On the other hand, this year we were able to answer 18 miscellaneous a 3 acronyms definition questions, which were completely ignored types last year.

As far as precision is concerned, we were able to significantly improve the results. Last year, RAPOSA achieved and global precision of 0.18 in run $R1_{2006}$ and 0.23 in run $R2_{2006}$ and a precision of 0.31 in run $R1_{2006}$ and 0.29 in run $R2_{2006}$ if we consider only non-nil answers. The overall results obtained this year (Table 1) are much higher in both cases: 0.33 and 0.54. The increase is mostly justified by the high precision that RAPOSA was able to obtain for definition questions (for which several context evaluation rules are available).

6 Future Work

While there are certainly many possibilities for improvement there are two points where we should invest more in the near future, and which are somehow connected. The first point is concerned with reducing the number of NIL answers that RAPOSA produces because of the insufficient number text snippets it extracts from text sources, sometimes none at all. In many cases the answer to a question can be found in the available text sources but the context surrounding that answer is expressed in different words of those contained in the questions. Currently, the Query Generator is not able to deal with situations where the interesting texts snippets would be found via generalization, specialization or paraphrasing of the words in the question.

Since we lack the lexical-semantic resources for Portuguese, such as thesauri or paraphrase dictionaries, to improve query generation in an efficient a scalable way, we have to find automatic ways of dealing with the problem. We are thus considering an approach that involves the automatic generation of paraphrases such as the ones described in [4] and [3]. One interesting “side effect” of increasing the number of snippets retrieved from the text sources is that we are also potentially

increasing the redundancy contained in the text to be analyzed, making it also more probable to find a correct answer, especially when using the simple type checking mechanisms for answering factoid questions.

Our second concern is the identification in free text of elements that, isolated or in specific contexts might be relevant to the answer certain types of questions. For example, for answering like questions like “Who is/was < person name >?”, e.g. “Who was Fernando Pessoa?” text snippets such “... the poet Fernando Pessoa...”, would contain (part of) the correct answer. Therefore we would like to semantically annotate such text snippets - “... *the*[*poet*]_{JOB} [*Fernando Pessoa*]_{HUM}...” - in order to perform answer extraction based on such high-level semantic tags. Our NER system already executes part of this job but it becomes very difficult to expand its semantic annotation rules, and related lexicon, to accommodate all variations needed for efficient open-domain question answering.

Therefore, our current line of research is on developing systems that perform *tagging by example*. Our goal is to have a system that, starting from a few examples of what elements would be good sources for answering a given type of question (the tagged snippet showed previously) would learn by bootstrapping how to detect and tag similar instance in text. For example, starting from “... *the* [*poet*]_{JOB} [*Fernando Pessoa*]_{HUM}...”, we would like the system to be able to annotate “similar” text instances such as “... *the* [*painter*]_{JOB} [*Pablo Picasso*]_{HUM}...” or “...*the* [*Queen of the Netherlands*]_{JOB} [*Beatrix Wilhelmina Armgard van Oranje-Nassau*]_{HUM}...” , which would all be a good source of information for answering “Who is/was < person name >?”. Ideally, the system would generalize and be able to also to tag other variations than convey similar and additional information such as “... *the* [[*german*] [*composer*]_{JOB}] [*Richard Wagner*]_{HUM}...” or “...[*António Guterrez*]_{HUM}, *the* [[*former*] [*portuguese*] [*prime-minister*]_{JOB}]], ...”.

For other types of questions, one would also (manually) point the learning system to other elements in text where the corresponding answers could be found. This would allow to semantically annotate other elements that could be then used to answer those types of questions. The underlying learning system follows a *lightly-supervised* learning strategy, which starting from a set of seed examples is able to learn similar instances by a bootstrapping mechanism. The core of the system is the ability to identify *type similarity* between elements (names, nouns, noun-phrases, adjectives), so as to be able to find in text instances that convey parallel or nearly-parallel information. Our recent experiments [10] in trying to identify type similar named-entities look promising, and suggest that the *tagging by example* approach may be viable and scalable to help answer extraction.

7 Conclusion

We were able to significantly increase the performance of RAPOSA, both on recall and precision by improving two fundamental blocks of the QA pipeline: Question Parsing and Answer Extraction. In both cases improvements were achieved by manually expanding the corresponding set rules. We were able to achieve high precision results in answering Definition Questions, mostly due to the good performance of the *context evaluation rules*. The results obtained in answering factoid questions using the more relaxed *simple type checking strategy* are still modest, but are slightly better the ones obtained last year.

We have pointed out ways for improving the global performance of the system. The first is including more efficient query expansion techniques in the text snippet retrieval stage: this will help reducing the number of NIL answers and increase the redundancy among the candidate answers, and thus also the overall precision of RAPOSA. The second, is based on the concept of “tagging by example” which would allow us to identify in text possible answers for several types of questions, based on a few seed examples of what text snippets that can contain such answers look like. We believe that this is a promising approach for improving the performance of RAPOSA in a scalable way, both for definitions and for factoid questions.

8 Acknowledgments

This work was partially supported by grant SFRH/BD/ 23590/2005 from Fundação para a Ciência e Tecnologia (Portugal), co-financed by POSI.

References

- [1] Adán Cassan, Helena Figueira, André Martins, Afonso Mendes, Pedro Mendes, Cláudia Pinto, and Daniel Vidal. Priberam’s question answering system in a cross-language environment. In Alessandro Nardi, Carol Peters, and José Luis Vicedo, editors, *Working Notes of the Cross-Language Evaluation Forum Workshop*. Alicante, Spain, 2006.
- [2] 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, and Julio Gonzalo, editors, *7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006. Alicante, Spain, September 2006. Revised Selected papers*, Lecture Notes in Computer Science. Springer, Berlin / Heidelberg, 2007.
- [3] Pablo Duboue and Jennifer Chu-Carroll. Answering the question you wish they had asked: The impact of paraphrasing for Question Answering. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 33–36, New York City, USA, June 2006. Association for Computational Linguistics.
- [4] D. Florence, F. Yvon, and O. Collin. Learning paraphrases to improve a question answering system. In *Proceedings of the 10th Conference of EACL Workshop Natural Language Processing for Question-Answering*, 2003.
- [5] Paulo Quaresma and Irene Pimenta Rodrigues. A logic programming based approach to qa@clef05 track. In Carol Peters, Fredric C. Gey, Julio Gonzalo, Henning Müller, Gareth J. F. Jones, Michael Kluck, Bernardo Magnini, and Maarten de Rijke, editors, *CLEF*, volume 4022 of *Lecture Notes in Computer Science*, pages 351–360. Springer, 2005.
- [6] Luis Sarmiento. SIEMÊS - a named entity recognizer for Portuguese relying on similarity rules. In *PROPOR 2006 - Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada*, pages 90–99, ME - RJ / Itatiaia, Rio de Janeiro - Brasil, 13 a 17 de Maio 2006.
- [7] Luís Sarmiento. BACO - A large database of text and co-occurrences. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odjik, and Daniel Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC’2006)*, pages 1787–1790, Genoa, Italy, 22-28 May 2006.
- [8] Luís Sarmiento. Hunting answers with RAPOSA (FOX). In Alessandro Nardi, Carol Peters, and José Luis Vicedo, editors, *Working Notes of the Cross-Language Evaluation Forum Workshop*. Alicante, Spain, 20-22 September 2006.
- [9] Luís Sarmiento. A first step to address biography generation as an iterative QA task. In Carol Peters, Paul Clough, Fredric C. Gey, Douglas W. Oard, Maximilian Stempfhuber, Bernardo Magnini, Maarten de Rijke, and Julio Gonzalo, editors, *7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006. Alicante, Spain, September 2006. Revised Selected papers*, Lecture Notes in Computer Science. Springer, Berlin / Heidelberg, 2007.
- [10] Luís Sarmiento, Valentin Jijkoun, Maarten de Rijke, and Eugénio Oliveira. “More Like These”: Growing Entity Classes from Seeds. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management (CIKM 2007)*, 2007.