The UPV at QA@CLEF 2006

Davide Buscaldi and José Manuel Gomez and Paolo Rosso and Emilio Sanchis
Dpto. de Sistemas Informticos y Computación (DSIC),
Universidad Politcnica de Valencia, Spain
{dbuscaldi, jogomez, prosso, esanchis}@dsic.upv.es

August 19, 2006

Abstract

This report describes the work done by the RFIA group at the Departamento de Sistemas Informáticos y Computación of the Universidad Politécnica of Valencia for the 2006 edition of the CLEF Question Answering task. We participated in three monolingual tasks: Spanish, Italian and French. The system used is a slightly revised version of the one we developed for the past year. The most interesting aspect of the work is the comparison between a Passage Retrieval engine (JIRS) specifically aimed to the Question Answering task and a standard, general use search engine such as Lucene. Results show that JIRS is able to return high quality passages.

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

General Terms

Measurement, Algorithms, Performance, Experimentation

Keywords

Question Answering, Passage Retrieval, Answer Extraction and Analysis

1 Introduction

QUASAR is the mono/cross-lingual Question Answering (QA) System we developed for our first participation to the past edition of the CLEF QA task. It is based on the JIRS Passage Retrieval (PR) system, specifically oriented to this task, in contrast to most QA systems that use classical PR methods [3, 1, 7, 5]. JIRS can be considered as a language-independent PR system, because it does not use any knowledge about lexicon and syntax of the language during question and passage processing phases. One of the objectives of this participation was the comparison of JIRS with a classical PR engine (in this case, Lucene[1]). In order to do this, we actually implemented two versions of QUASAR, which differ only for the PR engine used. With regard to the improvements over last year's system, our efforts were focused on the Question Analysis module, which in contrast to the one used in 2005 does not use Support Vector Machines in order to classify the questions. Moreover, we moved towards a stronger integration of the modules.

The 2006 CLEF QA task introduced some challenges with respect to the previous edition: list questions, the lack of a label distinguishing "definition" questions from other ones, and the

[1] http://lucene.apache.org

introduction of another kind of "definitions", that we named *object definitions*. This forced us to change slightly the class ontology we used in 2005. This year we did not participate to the cross-language tasks.

In the next section, we describe the structure and the building blocks of our QA system. In section 3 we discuss the results of QUASAR in the 2006 CLEF QA task.

2 Architecture of QUASAR

The architecture of QUASAR is shown in Fig.1.

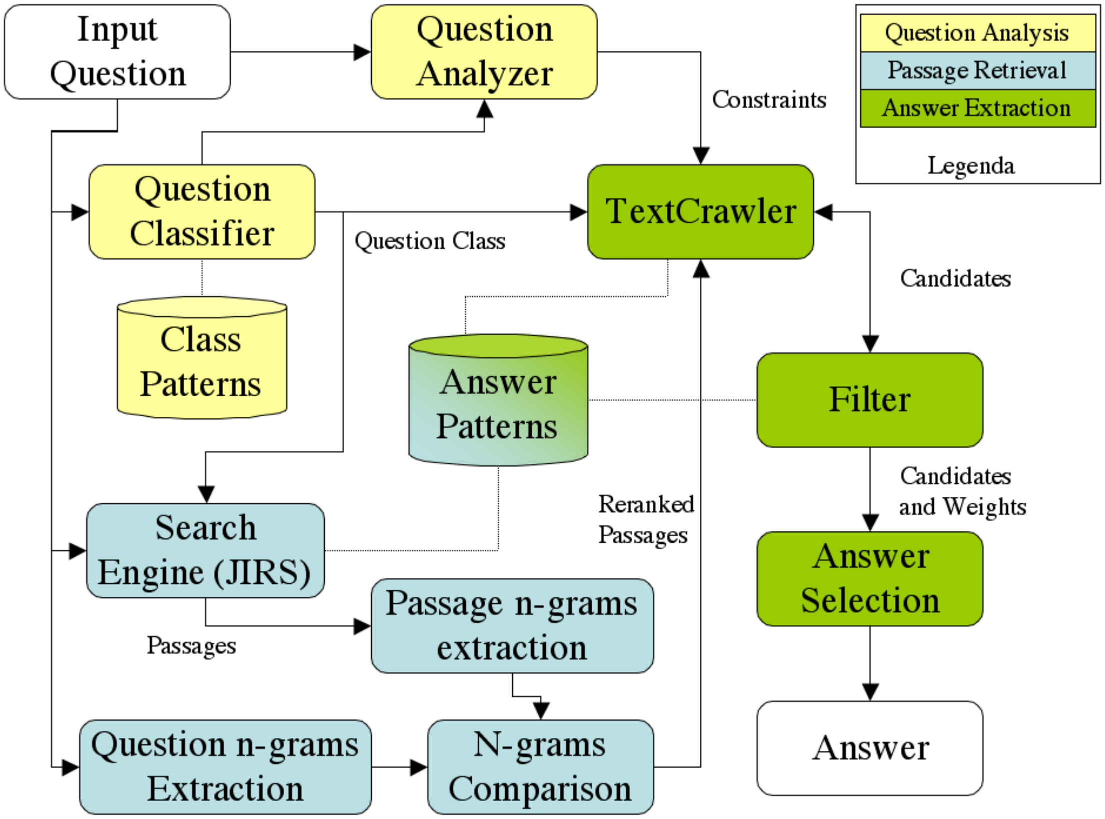

Figure 1: Diagram of the QA system

Given a user question, this will be handed over to the *Question Analysis* module, which is composed by a *Question Analyzer* that extracts some constraints to be used in the answer extraction phase, and by a *Question Classifier* that determines the class of the input question. At the same time, the question is passed to the *Passage Retrieval* module, which generates the passages used by the *Answer Extraction* (AE) module together with the information collected in the question analysis phase in order to extract the final answer.

2.1 Question Analysis Module

This module obtains both the expected answer type (or *class*) and some constraints from the question. Question classification is a crucial step of the processing since the Answer Extraction module uses a different strategy depending on the expected answer type; as reported by Moldovan et al. [4], errors in this phase account for the 36.4% of the total number of errors in Question Answering.

The different answer types that can be treated by our system are shown in Table 1. We introduced the “FIRSTNAME” subcategory for “NAME” type questions, because we defined a pattern for this kind of questions in the AE module. We also specialized the “DEFINITION” class into three subcategories: “PERSON”, “ORGANIZATION” and “OBJECT”, which was introduced this year (e.g.: *What is a router?*). With respect to CLEF 2005, the Question Classifier does not use a SVM classifier.

Table 1: QC pattern classification categories.

L0	L1	L2
NAME	ACRONYM	
	PERSON	
	TITLE	
	FIRSTNAME	
	LOCATION	COUNTRY
		CITY
		GEOGRAPHICAL
DEFINITION	PERSON	
	ORGANIZATION	
	OBJECT	
DATE	DAY	
	MONTH	
	YEAR	
	WEEKDAY	
QUANTITY	MONEY	
	DIMENSION	
	AGE	

Each category is defined by one or more patterns written as regular expressions. The questions that do not match any defined pattern are labeled with *OTHER*. If a question matches more than one pattern, it is assigned the label of the longest matching pattern (i.e., we consider longest patterns to be less generic than shorter ones).

The Question Analyzer has the purpose of identifying the constraints to be used in the AE phase. These constraints are made by sequences of words extracted from the POS-tagged query by means of POS patterns and rules. For instance, any sequence of nouns (such as *ozone hole*) is considered as a relevant pattern. The POS-taggers used were the SVMtool[2] for English and Spanish, and the TreeTagger[3] for Italian and French.

There are two classes of constraints: a *target* constraint, which is the word of the question that should appear closest to the answer string in a passage, and zero or more *contextual* constraints, keeping the information that has to be included in the retrieved passage in order to have a chance of success in extracting the correct answer. For example, in the following question: “*Dónde se celebraron los Juegos Olímpicos de Invierno de 1994?*” (*Where did the Winter Olympic games of 1994 take place?*) *celebraron* is the target constraint, while *Juegos Olímpicos de Invierno* and *1994* are the contextual constraints. There is always only one target constraint for each question, but the number of contextual constraint is not fixed. For instance, in “*Quién es Neil Armstrong?*” the target constraint is *Neil Armstrong* but there are no contextual constraints.

We did not implement

[2] http://www.lsi.upc.edu/ nlp/SVMTool/

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

2.2 Passage Retrieval Module (JIRS)

Passages with the relevant terms (i.e., without stopwords) are found by the Search Engine using the classical IR system. This year, the module was modified in order to rank better the passages which contain an answer pattern matching the question type. Therefore, this module is not as language-independent as in 2005 because it uses informations from the Question Classifier and the patterns used in the Answer Extraction phase.

Sets of unigrams, bigrams, ..., n-grams are extracted from the extended passages and from the user question. In both cases, n will be the number of question terms. With the n-gram sets of the passages and the user question we will make a comparison in order to obtain the weight of each passage. The weight of a passage will be heavier if the passage contains greater n-gram structures of the question.

For instance, if we ask "*Who is the President of Mexico?*" the system could retrieve two passages: one with the expression "*... **Vicente Fox** is the President of Mexico...*", and the other one with the expression "*...**Giorgio Napolitano** is the President of Italy...*". Of course, the first passage must have more importance because it contains the 5-gram "*is the President of Mexico*", whereas the second passage only contains the 4-gram "*is the President of* ", since the "*is the President of Italy*" 5-gram is not in the original question. To calculate the weight of n-grams of every passage, first the greatest relevance of n-gram in the passage is identify and we assign to this a weight equal to the sum of all term weights. Next, other n-grams less relevant are searched. These n-grams are not composed by terms of found n-grams. The weight of these n-grams will be the sum of all their weight terms divided by two, in order to avoid that its weight will be the same of the complete n-gram. The weight of every term comes fixed by (refeqtermweight):

$$w_k = 1 - \frac{\log(n_k)}{1 + \log(N)} \quad . \tag{1}$$

Where n_k is the number of passages in which the associated term to the weight w_k appears and N is the number of system passages. We make the assumption that stopwords occur in every passage (i.e., n_k takes the value of N). For instance, if the term appears once in the passage collection, its weight will be equal to 1 (the greatest weight). Whereas if it is a stopword its weight will be the lowest.

Depending on the style used to submit a question, sometimes a term unrelated to the question can obtain a greater weight than those assigned to the Named Entities (NEs), such as names of persons, organizations and places, dates. The NEs are the most important terms of the question and it does not make sense to return passages which do not contain them. Therefore, the equation (1) has been changed in order to give more weight to the NE than the rest of question terms and so to force its presence in the first passages of the ranking. In order to identify a NE a natural language processing is not used. We assumed that in most questions the NEs start with either an uppercase letter or they are a number. Once the terms are weighted, these are normalized for the sum of all terms are equal to 1.

JIRS can be obtained at the following URL: *http://leto.dsic.upv.es:8080/jirs*.

2.3 Answer Extraction

The input of this module is constituted by the n passages returned by the PR module and the constraints (including the expected type of the answer) obtained through the *Question Analysis* module. A *TextCrawler* is instantiated for each of the n passages with a set of patterns for the expected type of the answer and a pre-processed version of the passage text. For CLEF 2006, we corrected some errors in the patterns and we also introduced new ones.

Some patterns can be used for all languages; for instance, when looking for proper names, the pattern is the same for all languages. The pre-processing of passage text consists in separating all the punctuation characters from the words and in stripping off the annotations of the passage. It is important to keep the punctuation symbols because we observed that they usually offer important clues for the individuation of the answer (this is true especially for *definition* questions):

for instance, it is more frequent to observe a passage containing "*The president of Italy, Giorgio Napolitano*" than one containing "*The president of Italy IS Giorgio Napolitano*" ; moreover, movie and book titles are often put between apices.

The positions of the passages in which occur the constraints are marked before passing them to the TextCrawlers. A difference with 2005 is that now we do not use the Levenshtein-based spell-checker to compare strings in this phase now.

The TextCrawler begins its work by searching all the passage's substrings matching the expected answer pattern. Then a weight is assigned to each found substring s, depending on the positions of s with respect to the constraints, if s does not include any of the constraint words. If in the passage are present both the target constraint and one or more of the contextual constraints, then the product of the weights obtained for every constraint is used; otherwise, it is used only the weight obtained for the constraints found in the passage.

The *Filter* module takes advantage of a mini knowledge base in order to discard the candidate answers which do not match with an allowed pattern or that do match with a forbidden pattern. For instance, a list of country names in the four languages has been included in the knowledge base in order to filter country names when looking for countries. When the Filter module rejects a candidate, the TextCrawler provide it with the next best-weighted candidate, if there is one.

Finally, when all TextCrawlers end their analysis of the text, the *Answer Selection* module selects the answer to be returned by the system. The following strategies apply:

- Simple voting (SV): The returned answer corresponds to the candidate that occurs most frequently as passage candidate.
- Weighted voting (WV): Each vote is multiplied for the weight assigned to the candidate by the TextCrawler and for the passage weight as returned by the PR module.
- Maximum weight (MW): The candidate with the highest weight and occurring in the best ranked passage is returned.
- Double voting (DV): As simple voting, but taking into account the second best candidates of each passage.
- Top (TOP): The candidate elected by the best weighted passage is returned.

This year we used the Confidence Weighted Score (CWS) to select the answer to be returned to the system, relying on the fact that in 2005 our system was the one returning the best values for CWS [6]. For each candidate answer we calculated the CWS by dividing the number of strategies giving the same answer by the total number of strategies (5), multiplied for other measures depending on the number of passages returned (n_p/N, where N is the maximum number of passages that can be returned by the PR module and n_p is the number of passages actually returned) and the averaged passage weight. The final answer returned by the system is the one with the best CWS. Our system always return only one answer (or NIL), although 2006 rules allowed to return more answers per question. The weighting of NIL answers is slightly different, since is obtained as $1 - n_p/N$ if $n_p > 0$, 0 elsewhere.

The snippet for answer justification is obtained from the portion of text surrounding the first occurrence of the answer string. The snippet size is always 300 characters (150 before and 150 after the answer) + the number of characters of the answer string.

3 Experiments and Results

We submitted two runs for each of the following monolingual task: Spanish, Italian and French. The first runs (labelled *upv_061*) use the system with JIRS as PR engine, whereas for the other runs we used Lucene, adapted to the QA task with the implementation of a weighting scheme that privileges long passages and is similar to the word-overlap scheme of the MITRE system [2]. In Table 2 we show the overall accuracy obtained in all the runs.

Table 2: Accuracy results for the submitted runs. Overall: overall accuracy, factoid: accuracy over factoid questions; definition: accuracy over definition questions; nil: precision over nil questions (correctly answered nil/times returned nil); CWS: confidence-weighted score.

task	run	overall	factoid	definition	nil	CWS
es-es	upv_061	36.84%	34.25%	47.62%	0.33	0.225
	upv_062	30.00%	27.40%	40.48%	0.32	0.148
it-it	upv_061	28.19%	28.47%	26.83%	0.23	0.123
	upv_062	28.19%	27.78%	29.27%	0.23	0.132
fr-fr	upv_061	31.58%	31.08%	33.33%	0.36	0.163
	upv_062	24.74%	26.35%	19.05%	0.18	0.108

With respect to 2005, the overall accuracy increased by $\sim 3\%$ in Spanish and Italian, and by $\sim 7\%$ in French. We suppose that the improvement in French is due to the fact that the target collection was larger this year. Spanish is still the language in which we obtain the best results, even if we are not sure about the reason: a possibility is that this can be due to the better quality of the POS-tagger used in the analysis phase for the Spanish language.

We obtained an improvement over the 2005 system in factoid questions, but also worse results in definition ones, probably because of the introduction of the *object* definitions.

The JIRS-based systems performed better than the Lucene-based ones in Spanish and French, whereas in Italian they obtained almost the same results. The difference in the CWS values obtained in both Spanish and French is consistent and weights in favour of JIRS. This prove that the quality of passages returned by JIRS for these two languages is considerably better.

4 Conclusions and Further Work

We obtained a slight improvement over the results of our 2005 system. This is consistent with the small amount of modifications introduced, principally because of the new rules defined for the 2006 CLEF QA task. The most interesting result is that JIRS demonstrated to be more effective for the QA task than a standard search engine such as Lucene in two languages over three. Our further works on the QUASAR system will concern the implementation of a specialized strategy for definition questions, and probably a major revision of the Answer Extraction module.

Acknowledgments

We would like to thank R2D2 CICYT (TIC2003-07158-C04-03) and ICT EU-India (ALA/95/23/2003/077-054) research projects.

References

[1] Lili Aunimo, Reeta Kuuskoski, and Juha Makkonen. Cross-language question answering at the university of helsinki. In *Workshop of the Cross-Lingual Evaluation Forum (CLEF 2004)*, Bath, UK, 2004.

[2] Marc Light, Gideon S. Mann, Ellen Riloff, and Eric Breck. Analyses for elucidating current question answering technology. *Nat. Lang. Eng.*, 7(4):325–342, 2001.

[3] Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. Multilingual question/answering: the DIOGENE system. In *The 10th Text REtrieval Conference*, 2001.

[4] Dan Moldovan, Marius Pasca, Sanda Harabagiu, and Mihai Surdeanu. Performance issues and error analysis in an open-domain question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, New York, USA, 2003.

[5] Gunther Neumann and Bogdan Sacaleanu. Experiments on robust nl question interpretation and multi-layered document annotation for a cross-language question/answering system. In *Workshop of the Cross-Lingual Evaluation Forum (CLEF 2004)*, Bath, UK, 2004.

[6] Alessandro Vallin, Danilo Giampiccolo, Lili Aunimo, Christelle Ayache, Petya Osenova, Anselmo Peas, Maarten de Rijke, Bogdan Sacaleanu, Diana Santos, and Richard Sutcliffe. Overview of the clef 2005 multilingual question answering track. In *CLEF 2005 Proceedings*, 2005.

[7] José L. Vicedo, Ruben Izquierdo, Fernando Llopis, and Rafael Munoz. Question answering in spanish. In *Workshop of the Cross-Lingual Evaluation Forum (CLEF 2003)*, Trondheim, Norway, 2003.