The UPV at QA@CLEF 2007

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

August 18, 2007

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 2007 edition of the CLEF Question Answering task. We participated in the Spanish monolingual task only. A series of technical difficulties prevented us from completing all the tasks we subscribed. Moreover, an unnoticed indexing error resulted in the system working only on the Wikipedia collection. The system is a slightly revised version of the one we used in 2006. It was modified in order to comply with the 2007 guidelines, especially with regard to anaphora resolution, tackled with a web based anaphora resolution module. We obtained an accuracy of 11.5% despite the problems we experienced.

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, Anaphora Resolution

1 Introduction

QUASAR (QUestion AnSwering And Retrieval) is the name we gave to our Question Answering (QA) system. It is based on the JIRS Passage Retrieval (PR) system [3], specifically oriented to this task. JIRS does not use any knowledege about the lexicon and the syntax of the target language, therefore it can be considered as a language-independent PR system. The system we used this year differs slightly from the one used in 2007. Its major improvement has been the insertion of an Anaphora Resolution module in order to comply with the guidelines of CLEF QA 2007. As evidenced in [6], the correct resolution of anaphora is crucial and allows to improve accuracy of more than 10% with respect to a system that does not implement any method for anaphora resolution. The web is an important resource for QA [4] and has been already used to solve anaphora in texts [5, 1]. We took into account these works in order to build a web-based Anaphora Resolution module.

In the next section, we describe the structure and the building blocks of our QA system, including the new Anaphora Resolution module. In section 3 we discuss the results of QUASAR in the 2007 CLEF QA task.

2 System Architecture

In Fig.1 we show the architecture of the system used by our group at the CLEF QA 2007.

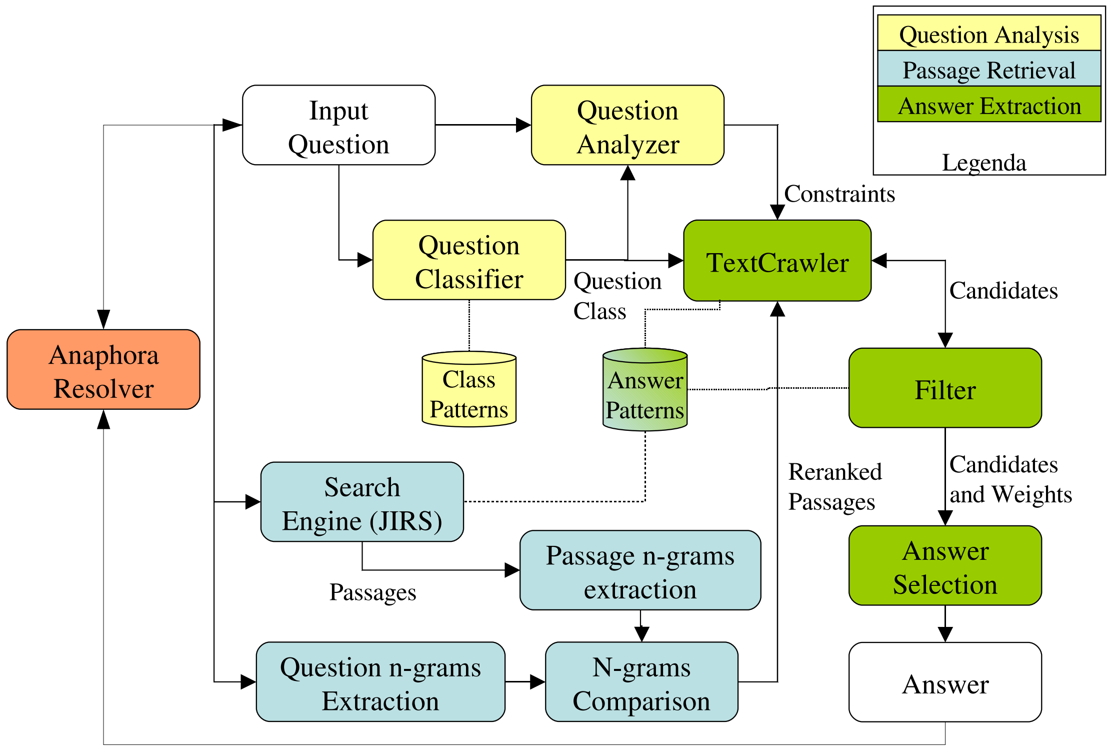

Figure 1: Diagram of the QA system

The user question is first examined by the *Anaphora Resolver*. This module will pass the question to QUASAR in order to obtain the answer that will be used in the following questions of the same group (i.e., questions over the same topic) for the anaphora resolution. The AR hands over the eventually reformulated question 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 Anaphora Resolution Module

The anaphora resolution module is the first or last module of the system (depending on the viewpoint). It carries out its work in 5 basic steps:

- Step 1: Using patterns in this step the module induces the different entity names, temporal and number expressions occuring in the first question of a group.
- Step 2 : For each question of the group the module replaces the entity names which occured in the first question (which is most likely to contain the target) and occur only partially the others. For instance if the first question was “Cuál era el aforo del Estadio Santiago Bernabéu en los años 80?” and another question within the same group was “Quién es el dueño del estadio?”, “estadio” in the second question is replaced by “Estadio Santiago Bernabéu”.
- Step 3: In this step the module focuses on pronominal anaphora and uses a web count to decide on weither to replace the anaphora with the answer of the previous question or with one of the entity names occuring in the first question of the group. For instance, if the first question was “Cómo se llama la mujer de Bill Gates?” and another question is “En qué universidad estudiaba él cuando creó Microsoft?” the module would check the Web counts of both “Bill Gates creó Microsoft” and “Melinda Gates creó Microsoft” and then replace the anaphora which whould change the second question to “En qué universidad estudiaba Bill Gates cuando creó Microsoft?”. However, in the Spanish language pronominal anaphora tend to be absent in the sentence for instance “En que año se fundó?” and this kind of questions are handled in Step 5 (see below).
- Step 4: The other type of anaphora left to resolve are the possessive anaphora. Similarily to the previous step the module decides on how to change the question using web counts. For instance, if we take the same example mentioned in Step 2 and we say that a third question was “Cuánto dinero se gastó durante su ampliación entre 2001 y 2006?”, the module would check the web counts of both “ampliación del Estadio Santiago Bernabéu” and “ampliación del Real Madrid Club de Fútbol” and change the question in order to become “Cuánto dinero se gastó durante ampliación del Estadio Santiago Bernabéu entre 2001 y 2006?”.
- Step 5: For the questions which have not been changed during any of the previous steps the module adds at the end of the question the entity name which has been found in the first question.

2.2 Question Analysis Module

This module obtains both the expected answer type (or *class*) and some constraints from the question. The different answer types that can be treated by our system are the same of 2006 [2]. Each category is defined by one or more patterns written as regular expressions. If a question matches more than one pattern, it is assigned the label of the longest matching pattern, on the basis of the assumption that longest patterns are less generic than shorter ones. Questions that do not match any defined pattern are labeled with *OTHER*. The patterns have been slightly modified since our 2006 participation.

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, mostly language-dependent. For instance, in English, a pair adjective - noun (such as *Russian president*) is considered as a relevant pattern. The POS-taggers used were the SVMtool[1] for English and Spanish, and the TreeTagger[2] 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. There is always only one target constraint for each question, but the number of contextual constraint is not fixed.

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

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

2.3 Passage Retrieval Module (JIRS)

Passages with the relevant terms (i.e., without stopwords) are found by the Search Engine using the classical IR system. 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. A complete description of JIRS is outside the scope of this paper and can be found in [3]. JIRS can be obtained at the following URL: *http://leto.dsic.upv.es:8080/jirs*.

2.4 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. The positions of the passages in which occur the constraints are marked before passing them to the text analyzers (*TextCrawlers*). One of these analyzers 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.

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. 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.

We used the Confidence Weighted Score (CWS) to select the answer to be returned to the system. 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

This year we experienced some technical difficulties during the test period, making us able only to submit one run for the Spanish monolingual task. Moreover, we realized after the submission that JIRS indexed only a part of the collection, specifically the Wikipedia snippet. The results is that we obtained only an accuracy of 11.5%. The reason of the poor performance is due partly to the fact that the patterns were defined for a collections of news and not for an encyclopedia, and partly to the fact that many questions had their answer in the news collection. We obtained 54 NIL answers, more than the 25% of the total number of questions, and only one time correctly.

4 Conclusions and Further Work

Our experience in the CLEF QA 2007 exercise was disappointing in terms of results, however we managed to develop an anaphora resolution module based on the web. The performance of this module cannot be evaluated from the results obtained at the CLEF, therefore we are considering to perform more experiments with the correct collections in order to improve the system.

Acknowledgments

We would like to thank the TIN2006-15265-C06-04 research project for partially supporting this work.

References

[1] R. Bunescu. Associative anaphora resolution: A web-based approach. In *EACL Workshop on the Computational Treatment of Anaphora*, Budapest, Hungary, 2003.

[2] Davide Buscaldi, José Manuel Gómez, Paolo Rosso, and Emilio Sanchis. The upv at qa@clef 2006. In Carol Peters, editor, *CLEF 2006 Working Notes*, Alicante, Spain, 2006.

[3] José Manuel Gómez, Manuel Montes, Emilio Sanchis, and Paolo Rosso. A passage retrieval system for multilingual question answering. In *8th International Conference of Text, Speech and Dialogue 2005 (TSD'05)*, pages 443–450, Karlovy Vary, Czech Republic, 2005.

[4] Jimmy Lin. The web as a resource for question answering: Perspectives and challenges. In *Language Resource and Evaluation Conference (LREC 2002)*, Las Palmas, Spain, 2002.

[5] K. Markert, N. Modjeska, and M. Nissim. Using the web for nominal anaphora resolution. In *EACL Workshop on the Computational Treatment of Anaphora*, Budapest, Hungary, 2003.

[6] José L. Vicedo and Antonio Ferrández. Importance of pronominal anaphora resolution in question answering systems. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 555–562, Morristown, NJ, USA, 2000. Association for Computational Linguistics.