miraQA: Initial experiments in Question Answering

C. de Pablo[1], J.L. Martínez-Fernández[1,4], P. Martínez[1], J. Villena[3,4], A. M. García-Serrano[2], J. M. Goñi[5] and J. C. González[4]

[1] Advanced Databases Group, Computer Science Department, Universidad Carlos III de Madrid,
Avda. Universidad 30, 28911 Leganés, Madrid, Spain
{cdepablo, jlmferna, pmf}@uc3m.es

[2] Artificial Intelligence Department, Universidad Politécnica de Madrid.
Campus de Montegancedo s/n, Boadilla del Monte 28660, Spain
{agarcia, vmendez, sgonzalez, mcastagnone}@isys.dia.fi.upm.es

[3] Department of Telematic Engineering, Universidad Carlos III de Madrid,
Avda. Universidad 30, 28911 Leganés, Madrid, Spain
jvillena@it.uc3m.es

[4] DAEDALUS – Data, Decisiond and Language, S.A.
Centro de Empresas "La Arboleda", Ctra. N-III km. 7,300 Madrid 28031, Spain
{jgonzalez, jmartinez, jvillena}@daedalus.es

[5] Department of Mathematics Applied to Information Techmologies,
E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid,
Avda. Ciudad Universitaria s/n,
28040 Madrid, Spain
jmg@mat.upm.es

Abstract

We present the miraQA system that constitutes MIRACLE first experience in Question Answering for monolingual Spanish and has been developed for QA@CLEF 2004. The architecture of the system is described and details of our approach to Statistical Answer Extraction based on Hidden Markov Models are presented. One run that uses last year question set for training purposes has been submitted. The results are presented together with ideas for improvement.

Introduction

Question Answering has received a lot of attention during the last years due to the advances in IR and NLP. As in other applications, the bulk of the research has mainly been centered around English, while perhaps, one of the most interesting applications of QA systems could be in cross and multilingual scenarios. Access to concrete quality information in a language that is not spoken or just poorly understood could be advantageous in lots of situations. QA@CLEF has encouraged the development of QA systems in other languages than English and in crosslingual scenarios.

QA systems are complex because of the number of different modules that they use, and the need for a good integration between them. Even in the case when questions are expecting a simple fact or a short definition as an answer, the requirement of more precise information has entailed the use of language and domain specific modules. On the other hand, some other approaches relying on data-intensive [3], machine learning and statistical techniques have achieved wide spread and relative success. Moreover, the interest of these approaches for multilingual QA systems lies on the possibilitiy of adapt them quickly to other target languages.

In this paper we present our first approach to the QA task. As we have not taken part before in any of the QA evaluation forums, most of the work has been spent in putting together a system from available resources. So far, the system we present is targeted only to the monolingual Spanish task. The system explores the use of Hidden Markov Models for Answer Extraction and uses Google to collect training data. The results show that further improvements in the method and appropiate tuning is needed but it remains promising. We expect to continue working on this system to enhance its results and inspect the suitability of the approach for different languages.

System Description

miraQA, the system developed for QA@CLEF 2004 by the MIRACLE group represents our first attempt to face Question Answering . As Spanish is our mother tongue, we have developed the system for the monolingual Spanish task, where the group is familiar with available tools. Despite the system was developed for Spanish, we had in mind that it should be easily adapted to other target languages. For that reason, we have explored the potential of statistical models for Answer Extraction. Besides, most of the tools that we are using, like POS taggers or partial parsers, are available for almost every other european language.

The general architecture of miraQA system for QA follows the classical structure in three modules and is presented in the following figure:

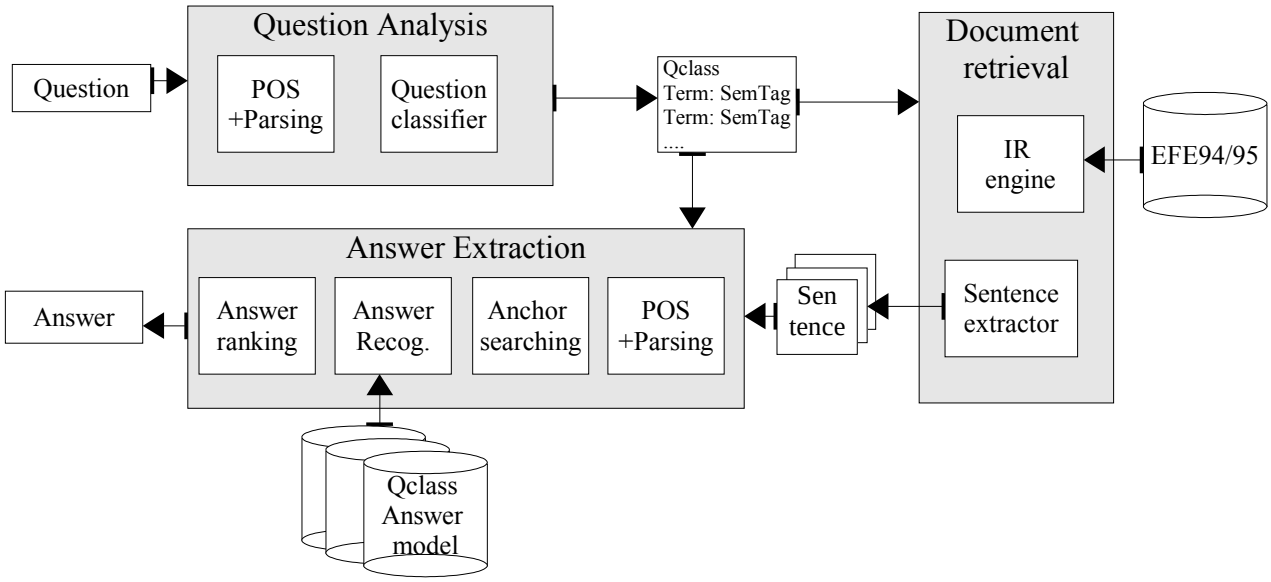

Figure 1: miraQA arquitecture

A fourth module for Answer Evaluation would be required to address this year novelty of providing a confidence measure for every answer. Although we appreciate the usefulness of that feature for the final user, we have not been able to include that module in our system due to time constraints.

In addition to the QA system, we have also developed another system to train the used Hidden Markov Models in our answer extraction phase. The system uses questions and answers to build queries that are posed to Google. Snippets of the results are extracted and used to build a model for the co-ocurrence of question terms and answers. In order to build the models we have used CLEF 2003 evaluation question set.

Question analysis

This module classifies the questions according to a manual taxonomy shown in Table 1 and composed of 17 classes. The taxonomy was decided considering mainly answer types. For some of them we decided to split or clonflate the classes depending on the frequency of appearance question-answer in QA@CLEF 2003 evaluation set. Questions are partially analysed using ms-tools [2]. We used MACO tagger and TACAT parser (slightly modified to avoid attacchment of PP chunks). Once the questions are partially parsed, a set of simple rules is applied to classify the questions determining its type, the type of the answer that is expected and assign a set of semantic tags to some of the chunks according to the relations they have with the answer. A simple example for the question "*¿Cual es la capital de Croacia?"* is shown in Figure 2:

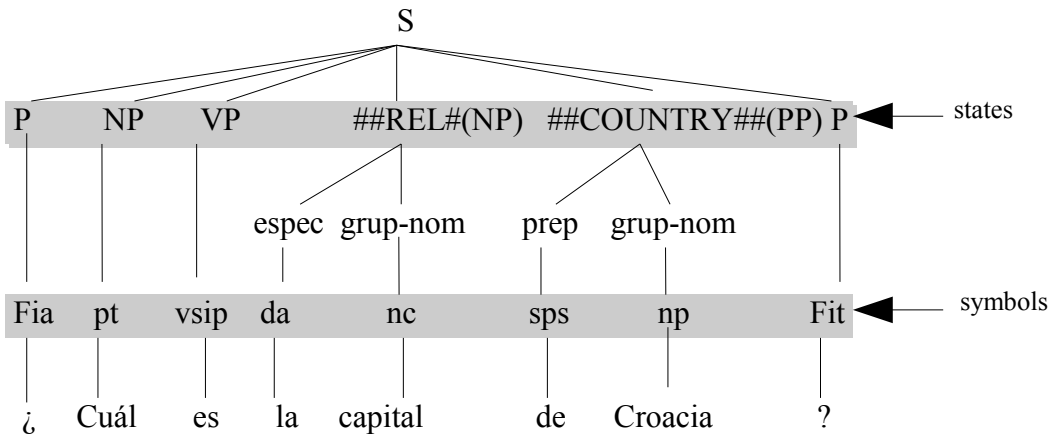

Figure 2: Question Analysis Example

Name	Time	Location	Cause
Person	Year	Country	Manner
Group	Month	City_0	Definition
Count	Day	City_ 1	Quantity
Rest			

Table 1 Question answer classes

Document retrieval

The IR module retrieves the top most relevant documents for a query and extracts the sentences that contain any of the words expressed in a query. After the question is analyzed, words that have a semantic tag assigned, are used in the query. For robustness purposes, the semantic tags are scanned again to remove stopwords and a query with all the terms is built and given to the IR engine. Our system uses Xapian [10] probabilistic engine to search for the most relevant documents. The last step consists on tokenizing the document using Daedalus Tokenizer [4] to extract those sentences that contain any of the words or stems that appeared in the query. The system assigns two scores to every sentence, the relevance measure provided by Xapian to the document and another figure proportional to the number of terms that were found in the sentence.

Answer extraction

The answer extraction module uses a statistical approach to answer pinpointing that is based on a syntactic-semantic context model of the answer built for any of the question-answer types. The following operations are performed:

1. *Parsing and Anchor Searching*. The sentences provided by the IR modules containing terms from the questions are parsed in a similar way as questions and training sentences using the ms-tools. Once parsed, the chunks containing the question terms are substituted by their semantic tags and constitute what we have called anchor terms. Finally, sentences are chunked in pieces that form a window of words around anchor terms and passed to the next module.
2. *Answer Recognition.* Pieces built in this way are passed to the answer extraction module that uses the HMM model. A variant of the N-best recognition strategy is used to identify the most probable sequence of states that originated the POS sequence and identifies an answer as the sequence of words that has been generated from the answer state. The recognition algorithm is guided by the semantic information in order to find a path that passes through the answer state. Besides, the algorithm provides a score for every path computed as the sum of the log of the probabilities for that path and sequence.
3. *Ranking*. Candidate answers are conflated when they present small differences in the outer form due to stopwords, for example. Finally, the candidate answers are ranked attending to a

weighted score that takes into account the score of the document, the sentence, the path followed in recognition and their lengths.

Training of Answer Context

Models that are used in the answer extraction phase are previously trained from examples. For the training of the models we used the question-answer set provided for QA at CLEF 2003. Questions are analyzed in the same way that they are in the main QA system. Question terms and the answers strings are combined in queries that we send to Google using the Google API. Snippets for the top 100 results are retrieved and stored to build the model. They are splitted in sentences, analyzed and chunks having questions and answer terms are retagged. The tag is either the semantic class assigned to that term in the question or the answer tag (##ANSWER##). Only sentences containing the answer and at least one of the other semantic tags are selected to train the model.

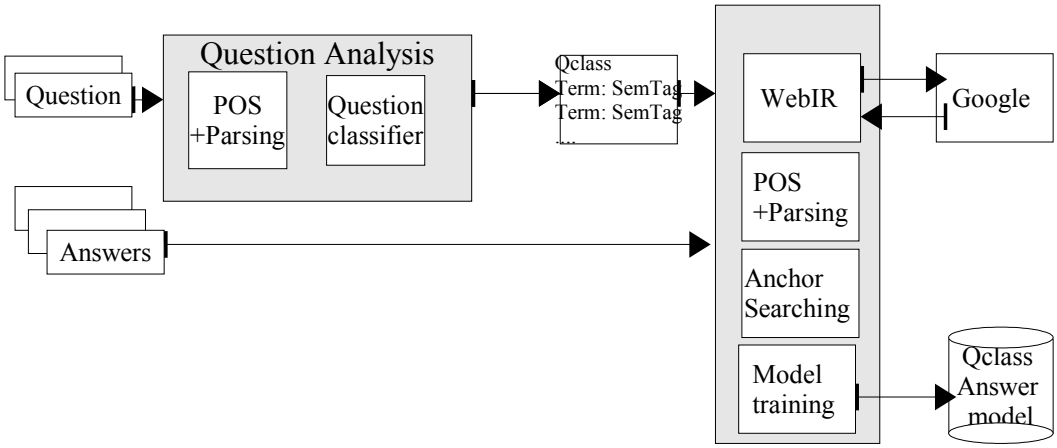

Figure 3: HMM Training subsystem

The machine that we built to extract answers is a Hidden Markov Model in wich the states are the syntactic-semantic tags assigned to the chunks while the emitted symbols are the POS tags assigned to the classes. To estimate the transition and emission probabilities we have counted the frequencies of the bigrams for POS-POS and POS-CHUNKS. In order to account for states or symbols that were not seen in the Google sentences we have used the simple add-one smoothing technique. We build a model like this one for every question-answer type that uses a closed set of one to three semantic tags.

Results analysis

We submitted one run for the monolingual Spanish task (mira041eses) that provides one exact answer to every question. Our system is unable to compute the confidence measure and we limited us to assign the default value of 0. There are two main kinds of questions, factoid and definition and we have tried the same approach for both of them. Besides, the question set contains some questions whose answer could not be found in the document corpus and the valid answer in that case is the NIL string.

The results obtained for our run mira041eses are outlined in Table 2

Question Type	***Right***	***Wrong***	***IneXact***	***Unsupported***
Factoid	18	157	4	1
Definition	0	17	3	0
TOTAL	18	174	7	1

Table 2 Result form mira041eses

Results are fairly low if we compare them with other systems. We attribute these bad results to the fact that the system is in a very early stage of development and tuning. We have obtained several conclusions from the

analysis of correct and wrong answers that will guide our future work. The extraction algorithm is working better for factoid questions than definitional. Obviously, among factoid questions results are also better for certain question-answer classes (DATE,NAME...) which are found often in the training set of questions. This is remarkable as the algorithm extract answers of the proper type even if they are incorrect. We were aware that such effect could appear as the amount of questions in each of the question-answers classes were unevenly distributed in QA@CLEF 2003 question set. There were specially few questions that we could classify as definitions wich diminish the amount of training data available and therefore the accuracy of the probabilities. Another fact noteworthy in our HMM algorithm is that is somewhat greedy when trying to identify answer and in that case shows some preference for words appearing near anchor terms . Finally, the algorithm is actually doing two jobs at a time as it identifies answers and, in some way, analyses entities according to patterns that were present in training answers of the same kind.

Another important source of errors in our system is induced by the document retrieval process and the way we posed questions and score documents. In our system all the terms that have been assigned a semantic tag will be used in queries and as anchors. Some terms are not very discriminating, specially if they are considered against proper names, and therefore lot of noisy documents are retrieved. As well, the simple scoring schema that we used for sentences (one token-one point) contributes to mask some of the useful fragments.

Errors are also generated by the question classification step as it is unable to handle some of the new surface forms introduced in this year question set. For that reason a catch all classification was also defined and used as a ragbag, but results were not expected to be good for that class.

The evaluation also provides results for the percentage of NIL answers that we have returned. In our case we returned 74 NIL answers and only 11 of them were correct (14.86%). NIL values were returned when the process did not provided any answer and their high value is due to the chaining of the other problems mentioned above.

Future work

Several lines for further research are open along with the deficiencies that we have detected in our system. We are also intending to extend the same approach to other languages both in the source and the target language. Some attempts to address different language for the question have already been done by translating questions, but the low quality of the translations would have obligued us to extend the set of question patterns or to develop correction mechanisms. These problems as well as the errors caused by new questions not addressed in our schema are claiming for a more robust approach to question classification and analysis.

One of the most straightful improvements we should introcuce in miraQA is a module for specific answer type recognition that address Named Entity Recognition but also other common answer types as dates, time, amounts, etc. With such extension the answer extraction task would be reduce to identify proper units based on context. We believe that with this improvement the method would be able to reduced the inexact ratio and address short definitional questions.

Results show that for the answer extraction mechanism to work properly, a thorough training is needed. We are already carrying out experiments to determine the amount of training data that would be needed in order to improve recognition results. We would likely need to acquire or generate larger question-answer corpus. Several improvements in the learning and recognition machine would be definetily beneficial and therefore several extensions to Hidden Markov Model and other statistical finite state approaches are under study, as well as more effective methods for learning the structure and parameters of these machines.

Besides the previous improvements a more careful look at the interfaces and dependencies between the different subsystems is also needed. In that sense, the main work involves developing better strategies to query de document database and retrieve the most meaningful passages. We also need to estimate more precisely the contribution of any of the modules and elaborate a method to combine this information in a succesful measure of answer confidence as this would greatly increase the acceptance of QA systems by the final user.

Finally, as we have stated from the beginning, we believe that an statistical approach could be practical from a software engineering approach and would allow the rapid development of baseline QA systems for different

languages and domains.

Acknowledgements

This work has been partially supported by the projects OmniPaper (European Union, 5th Framework Programme for Research and Technological Development, IST-2001-32174) and MIRACLE (Regional Government of Madrid, Regional Plan for Research, 07T/0055/2003).

References

1. Abney S., Collins M., Singhal A., **Answer Extraction** Applied Natural Language Processing (ANLP): Proceedings of the Conference. 2000.
2. Atserias J., J. Carmona, I. Castellón, S. Cervell, M. Civit, L. Màrquez, M.A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé and J. Turmo **Morphosyntactic Analysis and Parsing of Unrestricted Spanish Text.** Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC'98). Granada, Spain, 1998.
3. Baeza-Yates R. Ribeiro-Neto B. (1999) **Modern Information Retrieval**. Addison Wesley.
4. Brill E. Lin J. Banco M, Dumais S, Ng A. (2001) **Data-Intensive Question Answering**. *Proceedings of TREC 2001*
5. Daedalus Website: http;//www.daedalus.es
6. Jurafsky D. Martin J.H. (2000) **Speech and Language Processing**. Prentice Hall.
7. Manning C, Schütze H. (1999) **Foundations of Statistical Natural Language Processing**.. MIT Press
8. Magnini B., Romagnoli S., Vallin A., Herrera J.,Peñas A., Peinado V, Verdejo F and de Rijke M. **The Multiple Language Question Answering Track at CLEF 2003**
9. Vicedo J.L. (2003) **Recuperando información de alta precisión. Los sistemas de Búsqueda de Respuestas**. Phd Thesis. Universidad de Alicante.
10. Xapian Website: http://www.xapian.org/