The MIRACLE Team at the CLEF 2008 Multilingual Question Answering Track

Ángel Martínez-González[2], César de Pablo-Sánchez[1], Concepción Polo-Bayo[2],
María Teresa Vicente-Díez[1], Paloma Martínez-Fernández[1], José Luís Martínez-Fernández[1,2]
[1] Universidad Carlos III de Madrid
[2] DAEDALUS - Data, Decisions and Language, S.A.

amartinez@daedalus.es, cdepablo@inf.uc3m.es, cpolo@daedalus.es,
tvicente@inf.uc3m.es, pmf@inf.uc3m.es, jmartinez@daedalus.es

Abstract

The MIRACLE team is a consortium formed by three universities from Madrid, (Universidad Politécnica de Madrid, Universidad Autónoma de Madrid and Universidad Carlos III de Madrid) and DAEADALUS, a small and medium size enterprise (SME). The MIRACLE team participated in the monolingual Spanish and cross-language French to Spanish subtasks at QA@CLEF 2008. For the Spanish subtask, we used an almost completely rebuilt version of our system, designed with the aim of flexibly combining information sources and linguistic annotators for different languages. To allow easy development for new languages, most of the modules don’t make any language dependent assumptions. This language dependent knowledge is encapsulated in a rule language developed within the MIRACLE team. By the time of submitting the runs, work on the new version was still ongoing, so we consider the results as a partial test of the possibilities of the new architecture. Subsystems for other languages were not yet available, so we tried a very simple approach for the French to Spanish subtask: questions were translated to Spanish with Babylon, and the output of this translation was fed into our system. The results were an accuracy of 16% for the monolingual Spanish task and 5% for the cross-language task.

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

Keywords

Question Answering, Spanish, multisource, multilingual, rule, rule engine, correference, temporal expressions, timex.

1. Introduction

During the last year the MIRACLE QA system has gone through a major redesign and reimplementation that is still not finished. The main rationale of the new design is to flexibly combine heterogeneous information sources and linguistic annotation tools in a multilingual environment. In order to allow easy development for new languages, most of the modules don’t make any language dependent assumptions. This language dependent knowledge is encapsulated in a rule language developed within the MIRACLE team.

By the time of submitting the runs, work on the new version implementation was still ongoing, so we consider it as a partial test of the possibilities of the new architecture. For the monolingual Spanish task, the MIRACLE team submitted two runs. We sent one main run with our system’s best configuration, and another one where we test how the systems performs when varying the number of documents returned by the Information Retrieval subsystem. Although the modules for languages other than Spanish were not ready for this year’s participation, we also took part in the French to Spanish task, with a very simple strategy. We just translated the questions from French to Spanish and fed our system with the results.

This paper is structured as follows. The next section describes the system architecture with special attention paid to three novelties: the rule language, the enhanced topic detection and the temporal expression analysis module. Section 3 presents and briefly analyzes the results. Conclusions and directions of future work follow in section 4.

2. System architecture

This year the MIRACLE team took part in QA@CLEF with a system that has been rebuilt almost from scratch. The main objectives of the new design are:

- An architecture that can more easily be extended to Question Answering in languages other than Spanish. Though it was not ready for this year's CLEF, an English version of the system has been developed in parallel with the Spanish one.
- A system that can work simultaneously with several document collections in a flexible way. A new collection that has been processed offline can be hot-added to the running system by just changing the parameters of some configuration file. Answers from different collections are not anymore extracted independently and then combined, but they are managed by the same extraction module. Nevertheless, collection specific extraction rules can also be specified.
- Enhanced linguistic processing with a more detailed question analysis, the integration of temporal expression analyzer and the use of a rule engine in all the modules that require some symbolical decision taking, so that more complex rules can be created and modified in a faster way.
- The system is required to be a web application that provides answers in real time.

The system architecture is presented in figure 1. It has more or less the same structure as most state-of-the-art QA systems. Several modules (Question Classification, Time Expression Analyzer, Answer Extraction and Topic Tracking) rely on a rule engine that we developed for this purpose. A brief introduction to the rule language is given in section 2.1. The strategy followed to deal with multilinguality was to gather all the language dependent knowledge in the rules, so that all the other modules (with the exception of the language processor) are language independent.

The main difference with previous versions of the system is that there are no separate streams for the EFE Newswire and Wikipedia Collection. Instead of that, the system is now ready for several collections to be considered jointly. Each source is assigned a confidence value and also source specific extraction rules can be added to the system.

The system modules are:

- *Linguistic Analysis:* the architecture allows several tokenizers and linguistic annotators for each language to be cascaded. For Spanish, we used Daedalus STILUS [3] analyzer that provides tokenization, sentence detection and token analysis. Token analyses include detailed art of speech analysis, lemmatization and semantic information. For English, a more heterogeneous set of tools was used that includes Charniak parser [2], LingPipe Statistical Name Entity Recognizer [6], Wordnet [15] for lemmatization and self-created dictionaries for named entity annotation. Any other module that needs a linguistic processor for a language can get it without dealing with all those details using a Factory Design Pattern.
- *Time Expression Analyzer*: a component that analyzes and normalizes temporal expressions has also been integrated into the system. It is described in section 2.3.
- *Question Classification*: as an output of this module, the following values are determined for the question: focus, topic, question type, expected answer entity and a boolean feature telling whether the answer should be a list.

  The value of the expected entity is taken quite directly from the entity tags used by the STILUS tokenizer. STILUS uses a multilevel named entity hierarchy, which in turn is inspired in the one developed by Sekine [11]. This hierarchy has been used to tag by hand a large number of entities in STILUS dictionaries. This hand tagging of the resources is a very labor-intensive task, which is still in process.

Question	Expected entity	Abstraction levels
¿Qué es Opel? (What is Opel?)	INDUSTRIAL_COMPANY	class or subclass
¿Cómo se llaman las líneas aéreas de Niki Lauda? (What is the name of Niki Lauda's airlines?)	SERVICE_COMPANY	instance
¿Qué empresa tiene a Bibendum como mascota? (Which company has Bibendum as mascot?)	COMPANY	instance

Table 1: Expected entity for sample questions

Though Sekine's hierarchy was originally thought for tagging instances (which roughly correspond to proper nouns), STILUS resources apply it in a novel way to tag common nouns, marking them as classes or subclasses. For example, "*Opel*" is tagged as an instance of "ORGANIZATION-

>COMPANY->INDUSTRIAL_COMPANY", while "*líneas_aéreas*" ("*airlines*") is tagged as a subclass of "ORGANIZATION->COMPANY->SERVICE_COMPANY" and the word "*empresa*" ("*company*") is tagged as the class "ORGANIZATION->COMPANY". These tags help the rules in the question classification module to determine the expected entity as shown in table 1. For example, in the case of the question "*¿Qué es Opel?*" ("*What is Opel?*"), the expected answer is a class or subclass of "INDUSTRIAL_COMPANY". And for the question "*¿Cómo se llaman las líneas aéreas de Niki Lauda?*" ("*What is the name of Niki Lauda's airlines?*") an instance of SERVICE_COMPANY is needed.

Note that the expected entity feature is used not only for factoid questions, but also for definition questions. In a question like "*¿Quién era Edgar P. Jacobs?*" ("*Who was Edgar P. Jacobs?*"), it's interesting for other modules to know that the answer will probably contain words such as "*writer*" or "*artist*", which are tagged as subclasses of "PERSON".

- *Topic and correference tracker*: we have enhanced our topic candidate generator of previous years, by analyzing referring expressions in the follow-up questions, which often signal a change of topic inside the question group. This is explained in more detail in section 2.2.

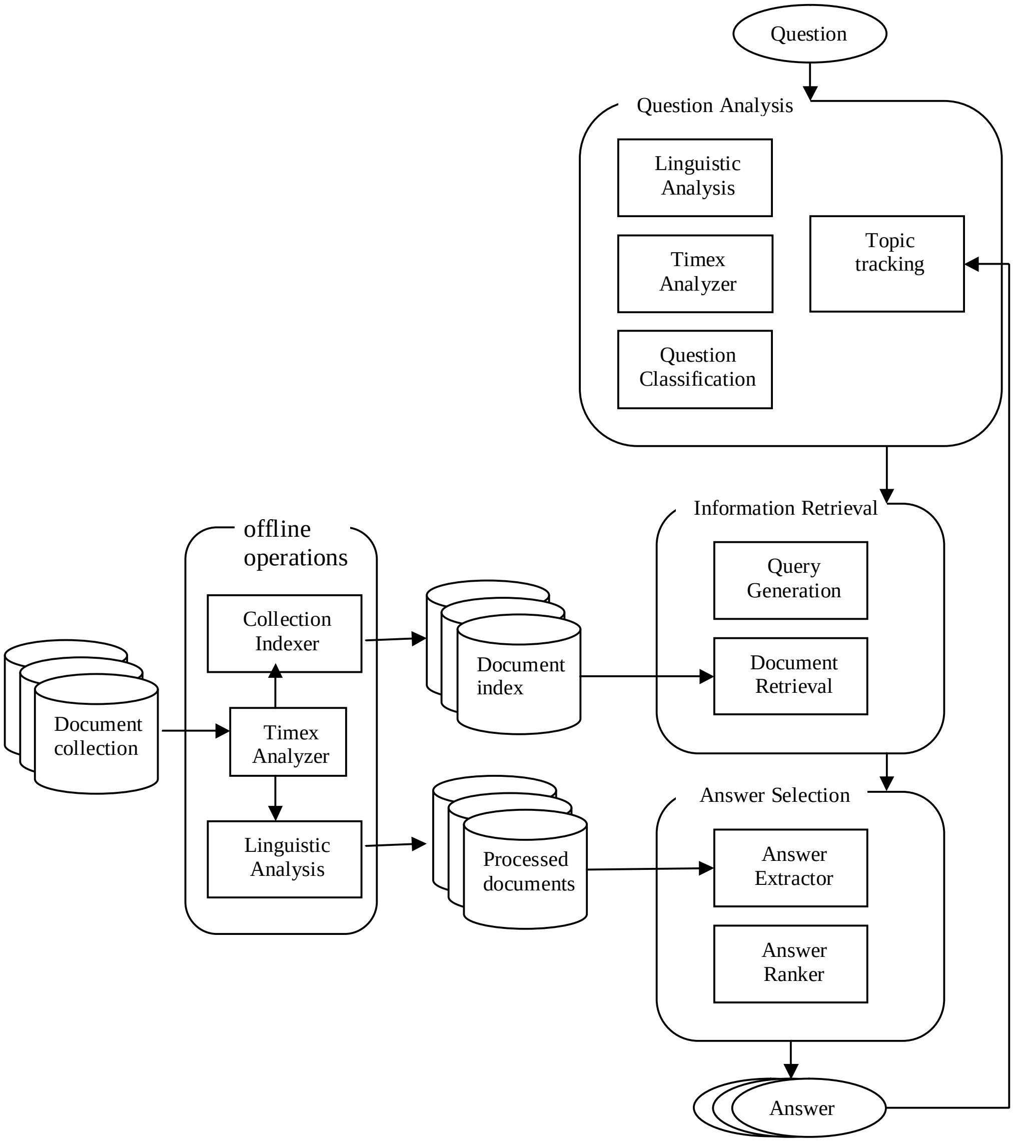

Figure 1: MIRACLE 2008 system architecture

- *Query generation and Document Retrieval*: Lucene was introduced for QA@CLEF 2007 as the Information Retrieval Engine and this year we kept it for this task. In this module, a query in Lucene syntax is generated and the most relevant documents according to cosine similarity ranking are retrieved.
- *Answer Extractor*: a ruled based approach that detects patterns for each type of question is used. The Sentence Selection Module present just before the Answer Extractor in the pipeline of previous versions of the system has disappeared and its functionality has been merged into the Answer Extractor. The reason of this is that many rules of the Answer Extractor had to be duplicated in the Sentence Selector, to ensure that the sentence containing that pattern reached the input of the Extractor.

  One of the improvement needs identified after our last year's participation in CLEF [8] was in the extraction of definitions from Wikipedia. This year we made an special effort to write rules for definitions; some of the heuristics used to recognize definitions include selecting Wikipedia articles whose title matches the question focus, giving priority to the first sentences of each document and searching for patterns that include the focus, expressions such as "*es*" ("*is*"), "*son*" ("*are*"), "*se denomina*" ("*is called*") and entities of the expected type. As discussed in section 3 of this document, a substantial improvement for definition questions was achieved with this approach.
- *Answer Ranker*: sorts the answer candidates according to some ranking formula. For this years runs, the following elements were taken into account:
  - Number of named entities in the support sentence that are compatible with the expected entity. An entity is said to be compatible with another if both are equal or the first is a child of the second in the hierarchy.
  - Number of term lemmas in the support sentence that also appear in the query.
  - Number of named entities in the support sentence that also appear in the question.

  The values mentioned above are normalized in order not to favour answers contained in longer sentences. Besides, terms of the support sentence that are closer to the extracted answer are likely to be more related to it. To reflect this, the values are weighted with a factor that reflects term proximity in the support sentence and ranges between 0 (if the term is more than 9 words away from the extracted part of the sentence) and 10 (if the term belongs to the extracted answer).

2.1. Rule engine

In previous versions of the system a rule engine was used for question classification. This approach was found very useful to separate decisions related to symbolical linguistic knowledge from the rest of the code, therefore, rules were introduced in other parts of the system such as Answer Extraction, Temporal Expression Analysis and Topic Tracking. Rules in this language are preprocessed to generate Java code. New rule predicates and actions to expand the rule language can also be defined handily in Java. For the particular case of Answer Extraction, we have found the rules suitable to incrementally introduce the quite different heuristics necessary to cope with heterogeneous sources and different question types.

```
BEGIN_RULE
   WORD_FORM(0, "¿") AND
   EXISTENTIAL_LEMMA(POS_FIRST_EXISTENTIAL_ANALYSIS(Tag_WHPronoun),
                     "quién")
THEN
   ADD_EXPECTED_ENTITY("PERSON")
END_RULE
```

Figure 2: Example of rule

The rules have a left part that expresses a pattern and a right part specifying the actions to be taken each time the pattern is found. The pattern is not necessarily only lexical; it can also have a syntactic or semantic component. While the search engine is working, there is always a current sentence, for which all the patterns are tested. Depending on the module where the engine is running, this current sentence will simply be the question or each one of the selected document sentences. Figure 2 shows a simplified example of rule. All it does is assign the expected entity "PERSON" to question starting with the interrogation sign "¿" and whose first interrogative pronoun is "*quién*". The rule language includes three types of constructs:

- *Predicates* (for example, EXISTENTIAL_LEMMA): return a boolean value and are used to check some linguistic feature such as word form, lemma, syntactical or semantic information. This predicates can be combined in the left part of the rule with boolean operators of conjunction, disjunction and negation. It has to be taken into account that in the case of ambiguity a word might have more than one analysis, so predicates may require that all the analyses of a word satisfy a condition (universal test) o just that one of them does (existential text). Other context data such as the current document title or collection identifier can also be tested.

- *Actions* to be taken if the rule fires (such as ADD_EXPECTED_ENTITY): these actions form the right side of the rule. They assign a type to the question, extract a part of a sentence as an answer, calculate the normalized form of a date, etc.
- *Position Functions* (for example, POS_FIRST_EXISTENTIAL_ANALYSIS): these functions return a position in the current sentence and are used as arguments to predicates and to actions. They give the language a higher order of expressiveness. Positions can also be kept in temporary variables.

In our design, the rules are supposed to be the only language dependent part of the code. Another principle we have found useful about the rules, is that the right side of the rules should perform all the suitable actions considering the linguistic knowledge obtained by the tests on the left side of the rule. A different design, for example with only one action taken by each rule, would lead to more complexity and redundancy. This idea of avoiding redundancy is the reason why the Sentence Selector is not present in our new system as a module separated from the Answer Extractor (as explained above in the architecture outline).

2.2. Topic detection and correference tracking

In this year evaluation, there were a large number of questions groups. In these groups the topic is presented in the first question and the following questions are related to this topic. The guidelines restrict the topic to any kind of entity or event introduced in the first question or the answer to this question. In contrast, an analysis of previous year topics reveal that sometimes the topic can change within a group. In Spanish, a topic shift is naturally introduced by the use of a referring expression that recalls a different entity or event. This is a similar but simplified view of the theory of centering (Grosz et al [4]). The example of table 2 is from group 2011 in CLEF 2007 topic set.

QUESTION	ANSWER	REFERENT LIST	REFERRING EXPRESSIONS	TOPIC
¿Quién fue Hermann Emil Fischer?	químico alemán			Hermann Emil Fischer
¿Que premio recibió en 1902?	Premio Nobel de Quimica	R1 = (Hermann Emil Fischer, químico alemán)	E1 (ellipsis)	E1=R1= (Hermann Emil Fischer, químico alemán)

¿Quién recibió el Premio Nobel de Literatura ese año?	Theodor Mommsen	R1 = (Herman Emil Fischer, químico alemán)
		R2 = Premio Nobel de Química,
		R3 =1902
			E2= ese año
			E3= Premio Nobel de Literatura
				E2 = R3 = 1902

Table 2: Example of topic tracking for a question group

In our previous participation [8] we implemented a rule based system for topic identification that considered candidates among the topic, the focus, the candidate answers and other constituents of the first question. The different candidates were selected based on syntactic heuristics and reordered depending on factors like the semantic type of the expected answer and the usual structure of an information seeking dialogue. For example, numbers, quantities and dates are uncommon as topics a priori when considered against persons or locations.

We have enhanced our topic tracking module by analyzing the follow-up question and tracking the use of referring expressions. Most referring relations in Spanish questions are realized by ellipsis and in those cases the a priori selection works well. In contrast when an explicit referring expression is introduced it usually signals a topic shift that reflects a reordering in the set of candidate referents. We have implemented rules that are able to track the most common cases in questions and answer dialogues: definite noun phrases (using determiners and articles), pronouns and named entities. Rare cases like epithets or verb nominalizations have been so far ignored. Once the candidate referring expression has been selected the next step consists in solving their coreferent. For each candidate pair of referent and referring expressions we calculate if they satisfy some agreement constraints. We have implemented five different constraints based on the linguistic information that is available after analysis: number, genre, lemma, semantic type and acronym expansion. A candidate pair that satisfies more constraints than the a priori best rated candidate for co-reference could be promoted if the score is much higher. So far the weights have been adjusted manually using previous examples and counterexamples. This accounts for the most common co-reference phenomena in QA dialogues although some others are also feasible like partitive, meronymy or collective correference and subject to future work ([12] and [5]).

2.3. Temporal expression analyzer

A precise analysis of temporal expressions is of vital importance both for questions about time ("*In what year did The Red Baron get the Blue Max?*") and for questions with some time restriction ("*Which city did Charlemagne capture in 778?*").

A temporal expression extractor and normalizer, which had been developed within the MIRACLE team ([13] and [14]), has been enhanced and integrated into our QA system for this year. The basis of the system is a set of recognition rules that defines a Finite State Grammar. For the definition of this grammar, an exhaustive study of the temporal expressions that appear in Spanish texts was necessary. The defined patterns include both absolute expressions, which are completely defined by themselves, and relative expressions which make reference to some other time that has to be known in order to be completely determined. Furthermore, both phrases that refer to time points or to intervals are considered. This latter classification is independent of the former, so we can have absolute time points ("*25/12/2007*"), relative time points ("*ayer*"/ "*yesterday*"), absolute intervals ("*entre 2000 y 2003*"/ "*between 2000 and 2003*") and relative intervals ("desde mayo hasta junio"/ "*from May to June*"). To define the normalized output value the international standard ISO 8601 (2004) for representation of dates and times is used. For the resolution of relative temporal expressions, some reference date is necessary. Though in some cases the reference date should be deduced from the context, for the integration in the QA system a simpler approach was followed and the document's date of creation is always the reference.

Input	Description	Resolution rule	Reference date	Normalized output

El 31 de diciembre de 2005	[ART|PREP]? DAY PREP MONTH_NAME PREP YYYY	Day =toDD (DAY)
		Month=toMM(MONTH_NAME)
		Year=YYYY
			NA	2005-12-31

mañana	DEICTIC_UNIT	Day=getDD(Creation_Time)+1
		Month=getMM(Creation_Time)
		Year=getYYYY(Creation_Time)
			2008-06-01	2008-06-02

Entre mayo y agosto	PREP
	MONTH_NAME1
	CONJ
	MONTH_NAME2
		Year1=getYYYY(Creation_Time)
		Month1=getMM(MONTH_NAME1)
		Day1=1
		Year2=getYYYY(Creation_Time)
		Month2=getMM(MONTH_NAME2)
		Day2=getLastDay(Month2)
			2008-06-01	2008-05-01
				2008-08-31

Table 3: Example of date recognition and normalization

The Temporal Expression Analyzer is integrated into the QA system at two levels:

- At the Information Retrieval level: Temporal expressions are normalized in the indexes generated from the document collections and in the queries generated from the questions. This way recall is increased.
- At a more symbolical level, the rules for answer extraction can use predicates that check whether a given token is a time expression and, in that case, which normalized value it has. For time restriction checking, a basic temporal inference mechanism has been developed, based on the principle of inclusion of a time point or interval in another interval.

3. Results

3.1. Submitted runs

Three runs were submitted by the MIRACLE team this year. Two monolingual runs for Spanish and a cross-language one for the French to Spanish subtask. For Spanish, we sent a main run using the system tuned as we thought it would yield best results. It is described in tables 4 and 5.

Name	Right	Wrong	Inexact	Unsupp.
mira081eses	32	156	3	9

Table 4: Judged answers for the main Spanish run

Factoids	Lists	Definitions	NIL Returned	Temporally Restricted
11.180 %	0.000 %	73.684 %	16.667 %	4.762 %

Table 5: Accuracy by question type for the main Spanish run

The results of the two other runs are summarized in table 6. Considering that two runs can be sent for each language, we wanted to employ the second Spanish run to test the variation in the system performance when changing some configuration parameter. We chose the maximum number of documents returned by the Information Retrieval module, setting it to 40 instead of the default value 20. As we expected, the result was worse, but not very significantly.

Finally, as explained in previous sections, the system for this year was developed with multilinguality in mind, but only the Spanish part was ready by the time of submitting the runs. Nevertheless, we decided to send a run with French as query language using a very simple approach: we translated the questions with Babylon [1] and just fed the Spanish System with the translation, with the non-surprising poor results shown in the second row of table 6.

Name	Right	Wrong	Inexact	Unsupp.
mira082eses	29	159	3	9
mira081fres	10	185	2	3

Table 6: Results for runs other than the main Spanish run

3.2. Error analysis

In this section the results for our main Spanish task are analyzed. These results can be considered as disappointing, as they suppose a very small improvement from last year (from an accuracy of 15% to 16%). We consider that the main reason of this is the lack of time to complete the development and tuning of the system after fully rewriting it this year. For example, no rules for the extraction of lists were added to the system before the deadline, this explains that there are no right answers of this kind. We have dealt with list questions in previous years so there is no technical reason that explains this absence but for lack of time (or a failure in task planning). Therefore, we consider the results as a partial test of the possibilities of the new architecture at an intermediate stage of development.

The modest improvement can also be partially attributed to the greater difficulty of the question set. According to an evaluation we have done with this year's system on 2007 questions, a 22% accuracy was obtained, compared to 15% of last year's system on the same question set. The new set had many more group questions, 110 instead of 50. And also some rather tricky questions were included: "*¿Quién es el guardameta de la selección española de baloncesto?*" ("*Who is the goalkeeper of the Spanish basketball national team?*").

On the other hand, table 5 shows a great improvement for definition questions, with an accuracy of 73,684%. This was one of the main weaknesses in our last year's participation [8]. The number of definition questions fell from 32 to 19.

4. Conclusions and future work

In the discussion about the results of the previous section, an analysis of the influence of each module on the errors is missing. We are currently working on an evaluation framework that lets as measure the performance of module independently. This framework will also allow easily putting together different configurations of the system, with different implementation of one module or different setup parameters, and testing the overall performance of the system.

The other main focus of work of the Miracle Team for next year is to introduce some more sophisticated logic description of the meaning of questions that goes beyond question focus and topic, probably using RDF. This semantic representation shall be compared with a similar analysis for document sentences, so that reasoning is possible with the aid of some of the available high-level open-domain ontologies ([10] and [16]).

Though this year's results in QA CLEF don't seem very promising, we consider it as an intermediate evaluation of an unfinished system. And we still keep our confidence the novelties we have introduced this year will yield fruit once we have the time to tune and debug the system.

Acknowledgements

This work has been partially supported by the Autonomous Community of Madrid by means of the project MAVIR (S-0505/TIC/000267) and by the Spanish Ministry of Education by means of the project BRAVO (TIN2007-67407-C3-01)

References

[1] Babylon website. http://www.babylon.com/, visited 18/08/2008.

[2] Charniak, E. A maximum-entropy-inspired parser. In Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics. Seattle, Washington, pp. 132-139. 2000.

[3] Daedalus. STILUS website. http://www.daedalus.es, visited 18/08/2008.

[4] Grosz, B.J.; Joshi, A.K.; and Weinstein, S. Centering. A framework for modeling the local coherence of discourse. Computational Linguistics 21 (2), 203-225, 1995.

[5] Jurafsky, D.; and Martin, J. H. Speech and Language Processing. Second edition. Chapter 21, 2008.

[6] LingPipe (Java libraries for the linguistic analysis of human language): On line http://www.alias-i.com/lingpipe/, visited 18/08/2008.

[7] Lucene webpage. http://lucene.apache.org/, visited 18/08/2008.

[8] de Pablo-Sánchez, C.; Martínez-Fernández, J. L.; González-Ledesma, A.; Samy, D.; Moreno, A.; Martínez, P.; and Al-Jumaily, H. MIRACLE Question Answering System for Spanish at CLEF 2007. In Working Notes of CLEF 2007, Budapest.

[9] de Pablo-Sánchez, C.; Gonzalez-Ledesma, A.; Moreno-Sandoval, A.; and Vicente-Díez, M.T. MIRACLE experiments in QA@CLEF 2006 in spanish: main task, real-time QA and exploratory QA using wikipedia (WiQA). 2007.

[10] Saias, J.; and Quaresma, P. The senso question answering approach to Portuguese qa@clef-2007. In Proceedings of CLEF – Cross Language Evaluation Forum, Budapest, Hungary, September 2007.

[11] Sekine, S. Sekine's extended named entity hierarchy. On line http://nlp.cs.nyu.edu/ene/, visited 18/08/2008.

[12] Vicedo, J. L.; and Ferrández, A. Correference in Q & A. In Advances in Open Domain Question Answering (edited by Strzalkowski, Tomek and Sanda Harabagiu), volume 32 of Text, Speech and Language Technology, pp. 71–96. Dordrecht: Springer, 2006.

[13] Vicente-Díez, M.T., de Pablo-Sánchez, C. y Martinez, P. 2007. Evaluacion de un Sistema de Reconocimiento y Normalización de Expresiones Temporales en Español. En Actas del XXIII Congreso de la Sociedad Española de Procesamiento de Lenguaje Natural (SEPLN 2007), páginas 113-120. Sevilla, (Spain). Septiembre 2007.

[14] Vicente-Díez, M.T., Samy, D. y Martínez, P. 2008. An Empirical Approach to a Preliminary Successful Identification and Resolution of Temporal Expressions in Spanish News Corpora. En Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). Marrakech, (Morocco). Mayo 2008.

[15] WordNet, a lexical database for the English language, http://wordnet.princeton.edu/, visited 18/08/2008.

[16] Zajac, R. Towards ontological question answering. In Proceedings of the workshop on ARABIC language processing: status and prospects, p.1-7, July 06, 2001, Toulouse, France.