Adapting IBQAS to work with text transciptions in QAst Task: IBQAst*

M. Pardiño, J.M. Gómez, H. Llorens, R. Muñoz-Terol, B. Navarro-Colorado, E. Saquete, P. Martínez-Barco, P. Moreda and M. Palomar
Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información
Natural Language Processing and Information Systems Group Department of Software and Computing Systems
University of Alicante, Spain
maria,jmgomez,hllorens,rafamt,borja,stela,patricio,moreda,mpalomar@dlsi.ua.es

August 15, 2008

Abstract

This paper shows the results of adapting a modular domain English QA system (called IBQAS, whose initials correspond to Interchangeable Blocks Question Answering System) to work with both manual and automatic text transcriptions. This system provides a generic and modular framework using an approach based on the recognition of named entities as a method of extracting answers.

The system architecture follows the general methodology of QA systems incorporating the modules detailed below: analysis of the question, information retrieval and extraction of the answer. In the analysis phase of the system, we extracted the type of question or type of answer expected, keywords and focus. Next, we used JIRS, a traditional Passage Retrieval system which is able to find structures in questions using n-gram models, for the information retrieval process. Finally, we selected the potential answers and those with higher scores were given as result.

1 Introduction

In this paper we will explain the adaptation to the CLEF 2008 QAST (Question Answering on Speech Transcription) track of the Question Answering (QA) system IBQAS previously developed by the University of Alicante [14] [13], and we will report our official evaluation results in the frame of this CLEF 2008 QAST track.

In order to perform the first participation of the University of Alicante in the CLEF 2008 QAST track, only the Question Answering process over manual and automatic transcriptions of European Parliament Plenary sessions in English (EPPS English corpus) has been carried out. So, the goal of the QAST process is to extract the correct answer to factual and definition questions over these recordings from the European Parliament. Nevertheless, we will only deal here with factual questions.

With the aim of performing the QAST process, the applied QA system is functionally structured in three QA tasks: question analysis, retrieval of relevant passages from automatic and

*This paper has been partially supported by the Spanish government, project TIN-2006-15265-C06-01 and project GV06028, by the framework of the project QALL-ME, which is a 6th Framework Research Programme of the European Union (EU), contract number: FP6-IST-033860, and by the Generalitat Valenciana throught the research grant BFPI/2008/093.

manual speech transcripts and answer extraction.

The application of this QA system to the QAST process is explained in the following sections of the paper. So, next section presents the state-of-the-art of the systems that also perform QAST process. Section number three details the core of the QAST-based system and its application to the QAST process. Section number four shows the results obtained by the system according to the CLEF 2008 QAST evaluation track. Finally, the last section details the conclusions and further works.

2 QAST background

More concretely, in the state of the art for QAST task in CLEF 2007 there are five main systems:

- University of Catalonia [16]: This research group participated with two systems in the four sub-tasks. Its main feature was that the systems made minimal use of syntactic analysis and used a data-driven query relaxation algorithm to extract the best answer context from the input question. The difference between the two systems was that one was tailored for manual transcripts, while the other was tailored for automatic transcripts. In all four sub-tasks they obtained the best performance with the system that was initially designed for manual transcripts. Although the system designed for automatic transcripts performed worse than expected, this approach is a good long-term research direction because it is the only one of the two systems developed that can truly address the specific phenomena of automatic transcripts. Their best performing runs have TOP1 scores that range from 0.21 (on automatic transcripts with WER of 38%) to 0.51 (on manual transcripts).

- LIMSI participation [18]: This group presented two different QA systems based on a complete and multilevel analysis of both queries and documents. The main changes between both systems were the replacement of the large set of hand-made rules by the automatic generation of a research descriptor, and the addition of an efficient scoring of the candidate answers. The evaluation of the systems showed that, on transcribed lectures, the best Accuracy result was 39% for manual and 21.3% for automatic, and, on transcribed meetings was 24% for manual and 18.3% for automatic.

- AnswerFinder [2]: This contribution was centered on a study of Named Entity (NE) recognition on speech transcripts, and how such NE recognition impacts on the accuracy of the final question answering system. The NE recognizer (AFNER) of the AnswerFinder question-answering project was ported to the types of answer expected in the QAst track. They participated in all QAst sub-tasks with two runs per task. Their conclusions were that the small training corpus and the presence of annotation errors in the AMI corpus made the machine learning component of AFNER ineffective. Nevertheless, the system was second (out of three participants) in one of the QAst subtasks (Task 3) with 19.77% accuracy for the second run.

- Tokyo Institute of Technology [3] presented a QAST system based on non-linguistic, data-driven approach with a noisy channel model. This system had two modules: the first one is a IR system with a sentence-based retrieval approach. The corpus was pre-processed with simple text processing: fillers and pauses removing, etc. The second module was the answer extractor: the best answer was extracted with the maximum probability based on Bayes' rules. The system was fourth in the QAst subtasks 1 with 0.20% of MRR.

- Finally, as previous papers, Neumann and Wang [15] adapted a previous open-domain QA system to the specific task of QA in speech transcription: QAst-v1. This system was developed for factual questions, and it was based on a NER system. They pre-process the speech transcription corpus with automatic annotation of sentence boundaries, chunk structures (based on dependency analysis) and Named Entities. Furthermore, they analyzed questions with shallow dependency structures, NE recognition and expected answer type. For the location of candidates answers they used redundancy, filtered by the correspondence between the named entities of possible answers and the expected answer type. Ill-formed candidates answers were deleted by manually specified rules. They run the system in two subtasks: T1 and T2. In the first one achieved 0.15 accuracy, and in the second one 0.09.

3 Description of the System

This work shows the results of adapting a modular domain English QA system IBQAS based on the proposal of Pizzato [17] to work with text transcripts both manual and automatic. This system provides a generic and modular framework using an approach based on the recognition of named entities as a method of extracting answers.

The system architecture follows the general methodology of QA systems incorporating the modules detailed below: analysis of the question, information retrieval and extraction of the answer.

In the analysis phase of the system, we extracted the type of question or type of answer expected (by means of patterns like a question type taxonomy previously defined), keywords (verb main phrases without nominal stopwords, denials) and focus (it describes the type of answer expected when it is not possible to infer it from the interrogative particle, but it rarely appears in the sentence where the answer is, so it should be removed from the list of keywords).

For the information retrieval process, we did not use the IR module incorporated in the original IBQAS but we adapted JIRS. JIRS is an IR system able to find structures in question using n-gram models. It uses a traditional Passage Retrieval system and searches each n-gram of the question in the retrieved passages. Afterwards, it rates them depending on quantity and weight of the n-grams of these passages.

Finally, relevant documents are filtered and potential answers are extracted from them (using Lingpipe to recognize location, person and organization entities; TERSEO for temporal expressions and patterns to recognize other types of entities such as numeric entities, languages ...). The last step consists on scoring and sorting the responses obtained to select several of them according to the distance between each response and the keywords, as well as the mutual information of the bigrams and trigrams of the passages.

3.1 Question Analysis

The importance of this module relies on the fact that the success of all the other parts of the system depends on it. Its goal is to extract any relevant information from the question. More specifically, the presented system extracts the question type, the focus and the keywords (see figure 1).

3.1.1 Question Type

The question type indicates the expected answer type. For instance, the question *¿Where is the Eiffel Tower?* is expecting a location as answer. In this way, the goal of this section is to classify the questions among the different types, given a taxonomy. Several taxonomies have been developed with this purpose. Normally, these taxonomies are hierarchically organized offering different granularities (coarse or fine) [11, 8]. Regarding the classification method, two main types can be considered. On the one hand, methods based on Regular Expressions (REs) or patterns[11, 7] are faster but have a lower recall and a higher development cost. On the other hand, methods based on machine learning [8] are slower but offer, normally, a higher recall and a lower development cost.

The presented system uses a set of REs in order to determine the question type. The implemented patterns represent an extension of the ones used by Molla in AnswerFinder[12]. The REs

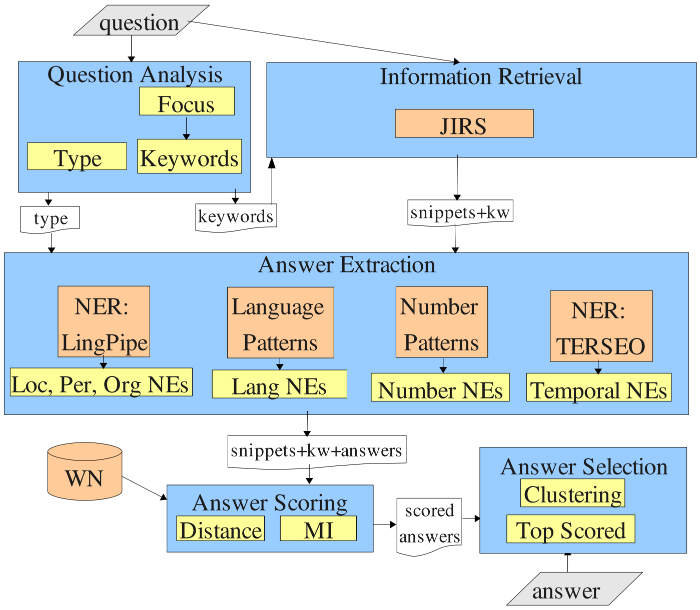

Figure 1: Configuration of our system

are ordered by relevance. Therefore, if one question matches with more than one RE the system returns the first one, that is, the most relevant one. Table 1 shows the taxonomy used that is based on Li and Roth taxonomy[8].

Table 1: Coarse grain question type taxonomy

Question Type Taxonomy	
Person (Human)	Temporal
Number	Organization
Location	Language
Definition	Unknown

3.1.2 Question Keywords

The relevant words of a question are considered keywords. The presented system considers as keywords:

- Main verb of the question
- Nominal phrases, once stopwords are removed (the, of, etc.)
- Negations

As implementation, a syntactical analysis of dependencies is carried out using MINIPAR [1] [9] in order to select keywords:

1. Using MINIPAR information, the system extracts:

[1] http://www.cs.ualberta.ca/~lindek/minipar.htm

- Subject Nucleus
- Object Nucleus
- Nominal Phrases Nucleus
- Main verb Nucleus
- Negations

2. Each item of the previous list is extended with:
   - Expressions related through prepositional modifiers
   - Expressions related through subordination modifiers
3. Duplicate entries are removed from keyword list
4. Focus is omitted from keyword list

MINIPAR detects multiwords and the presented system takes advantage of this feature treating them as single words.

3.1.3 Question Focus

Question focus describes the expected answer type when this type can not be deduced from the question word, that is, in What/Which type questions. Furthermore, it is an expression that normally does not appear in sentences containing the answer. The method to detect the question focus consists in extracting the expression that follows or is related to the question word[6]. The presented system uses this method to detect question focus and it omits it from the question keywords list.

3.2 Description of the JAVA Information Retrieval System

JAVA Information Retrieval System (JIRS) is an IR system specially adapted to retrieve passages. Our Passage Retrieval (PR) system is based on searching the question structures rather than just the keywords, and it makes a comparison between them. JIRS uses a traditional search engine as the first step and then searches all possible n-grams of the question in the retrieved passages and rates them depending on the number and the weight of the n-grams that appeared in these passages. The system architecture is shown in Fig. 2.

JIRS is based on searching the heaviest n-grams (i.e., those with the greatest weight) instead of the longest one using the *Distance Density n-gram* model [5]. With this model, the final similarity is obtained by multiplying the n-gram weight by a distance factor that takes into account the distance with respect to the heaviest n-gram. Therefore, the similarity value depends on the density of question terms in the passage, and it is calculated as the sum of all n-gram weights, multiplied by the distance factor and divided by the sum of all term weights of the question. The equation we have used is the following:

$$Sim(p,q) = \frac{1}{\sum_{i=1}^{n} w_i} \cdot \sum_{\forall x \in \hat{P}} h(x) \frac{1}{d(x, x_{max})} \tag{1}$$

Let p be the set of n-grams composed by passage terms and $\hat{Q}$ be the set of n-grams of p composed only by question terms. Therefore, we define $\hat{P} = \{x_1, x_2, ..., x_M\}$ as a sorted subset of $\hat{Q}$ that fulfills the following conditions:

1. $\forall x_i \in \hat{P} : h(x_i) \geq h(x_{i+1}) \quad i \in \{1, 2, ..., M-1\}$
2. $\forall x, y \in \hat{P} : x \neq y \Rightarrow T(x) \bigcap T(y) = \emptyset$
3. $\min_{x \in \hat{P}} h(x) \geq \max_{y \in \hat{Q} - \hat{P}} h(y)$

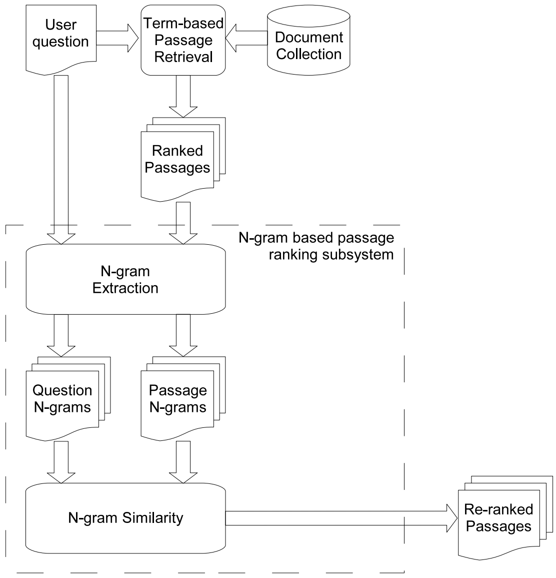

Figure 2: The main architecture of the JIRS n-gram based PR system

where $T(x)$ is the set of terms of the n-gram x, and $h(x)$ is the function which measure the n-gram weight and it is defined by the Equation (2):

$$h(x) = \sum_{k=1}^{j} w_k \tag{2}$$

where $w_1, w_2, ..., w_j$ are the term weights of the j-gram $x = t_1 t_2 ... t_j$. These weights should penalize the terms that appear frequently in the document collection (e.g. stopwords) and promote the relevant words (i.e. the question terms that are of crucial importance to retrieve a relevant passage). The following function was introduced to assign the weight to a term:

$$w_k = 1 - \frac{\log(n_k)}{1 + \log(N)} \tag{3}$$

where n_k is the number of passages in which the term t_k appears, and N is the number of system passages. We assume that stopwords occur in every passage (i.e., n_k takes the value of N). For instance, if the term t_k occurs only once in the passage collection, its weight will be equal to 1 (the greatest weight). However, if it is a stopword, its weight will be the lowest one.

The importance of the term proximity weighting has already been addressed in IR [20] which describes how lexical cohesion between query terms in documents might be used in document ranking. From our perspective, the simplest measure of distance between two n-grams could be defined as the number of terms between them. Nevertheless, this function has the disadvantage that it grows linearly and, therefore, the weight of the n-gram decreases too fast with respect to its distance from the heaviest n-gram. In order to address this issue, we use a logarithmic distance instead of the linear one. The distance function we have used is the following:

$$d(x, x_{max}) = 1 + k \cdot \ln(1 + L) \tag{4}$$

where L is the number of terms between the n-gram x_{max} (x_{max} is the n-gram with the maximum weight calculated in the Equation (2)) and the n-gram x of the passage. We have introduced the k constant to adjust the importance of the distance in the similarity equation. In previous experiments, we have determined that the best score for this value is 0.1. The other added constants are used to avoid the infinities when L is equal to 0.

3.3 Answer Extraction

The aim of this module is to determinate which parts of the selected information are potential answers for the question formulated by the user. The main difference between this module and the previous one is that here we are looking at concrete pieces of information, exact answers. Once extracted the feasible answers for the question, they are scored and reranked with the aim of selecting one of them as the final answer.

Figure 1 shows the architecture used by the system in this module. The implementation is divided into three main processes:

3.3.1 Candidate Answer Detection

The presented system uses Named Entity Recognizers (NERs) in order to detect candidate answers. To do this, the system uses different NERs to detect and classify different Named Entity (NE) types according to different question types.

1. **Named Entity Recognition**: Used NERs.
   - *Location, Person and Organization (Lingpipe).* To be able to detect answers for *Person, Organization or Location* type questions, the presented system uses Lingpipe[2] by Carpenter y Baldwin. Lingpipe was evaluated with ConLL 2002 data obtaining a 77.29% F1.
   - *Temporal Expressions (TERSEO).* Due to the spareness of temporal expression recognition feature in most NERs, a specialized NER to recognize and normalize temporal expressions was used. The presented system uses TERSEO[3] [19] to do this.
   - *Other entities.* In order to detect other types of entities like *Number*, the system uses specific patterns of intern and extern evidences following the steps of Mikheev, Moens y Grover [10].
2. **Candidate Answer Selection Process**: Knowing the question type, and therefore the entity type of the answer the system follows the following steps.
   (a) It executes the corresponding NER over the information obtained by the IR module.
   (b) Once entities are tagged, it removes every entity of non searched types.
   (c) It omits entities corresponding to keywords.
   (d) It applies filters in order to remove punctuation symbols, blanks or stopwords[4] tagged as candidate answers.

3.3.2 Candidate Answer Scoring and Reranking

The presented system uses different techniques to score and rerank obtained answers. The scoring techniques used are:

1. **Answer-Keywords Distance** This method assumes that the closer the keywords are from candidate answers the better the method is. Formula 5 shows the way this distance is calculated.

$$D(A) = \sum_{i=1}^{n} \frac{\delta(A, f_i) - 1}{n} \tag{5}$$

   Defining $\delta(a, b)$ as the number of words between a and b, and $F = f1, f2, ..., fn$ as the list of keywords, distance function D for a concrete candidate answer A, is defined as shown in

[2] http://www.alias-i.com/lingpipe/
[3] http://gplsi.dlsi.ua.es/∼stela/TERSEO/
[4] http://dev.mysql.com/doc/refman/5.0/en/fulltext-stopwords.html

formula 5.

2. **Mutual Information and WordNet relationships**

   As an additional scoring method, the presented system uses Mutual Information (MI)[4] of bigrams and trigrams increasing the score of the answers that has a high MI in direct relationship with the keywords or in indirect relationship through WordNet (WN) with the keywords. $MI(a, b)$ of two words, a bigram, a and b is calculated as shown in formula 6.

$$I(a, b) = \log \frac{P(a, b)}{P(a)P(b)} \tag{6}$$

   The probability of an n-gram is calculated as described in formula 7.

$$P(x) = \frac{freq(x)}{total\ n - grams} \tag{7}$$

   The probability P of a concrete n-gram x is calculated by dividing its frequency in the text $freq(x)$ by the total of n-grams in the text $total\ n - grams$.

   The presented system calculates the MI of the bigrams and trigrams once stopwords[4] are removed from the text. Only n-grams repeated more than 5 times are taken into account in order to smooth the imprecision caused by the spareness of data in MI [1].

   The relevant n-grams are considered using the following steps:

   (a) Remove n-grams composed only of keywords.

   (b) Remove n-grams not containing any of the candidate answers.

   (c) The rest are scored as follows:
      - Number of keywords they contain.
      - Number of words in the n-gram that have any relationship with keywords. To perform this the system uses WN information.

   (d) Additional Score: $AS = (MI * (numkws + numrels + 1))/20$

3. **Answer Clustering**

   Before determining which are the best answers, the presented system carries out an *Answer Clustering* process. Equal answers or overlapped answers are grouped adding its scores to the most scored one. Once an ordered list of grouped and rescored answers is obtained, the final answers are those with the highest score.

4 Experiments and Results

Finally, we present the results obtained with our system in Qast task. We sent one manual run (for task T4a) and three automatic runs, one for each existing automatic transcriptions (for task T4b) all working with EPPS English corpus. The results are shown in Tables 2 and 3.
Specifically, Tables 2 shows the number of wrong answers (W), unsupported answers (U), inexact answers (X) and right answers(R) in the first answer returned for each question given to our system.

Such us we expected, the best results have been obtained with the manual transcription. This is due to the fact that this transcription has fewer errors than automatic transcriptions because most of the problems have been checked manually.

Task	Run_ID	# Questions	#R	#X	#U	#W	#NIL	#NIL=R
T4a	ali1_t4a.txt (manual)	100	20	2	0	78	34	6
T4b	ali1_t4b_a.txt (automatic)	100	7	0	0	93	42	5
T4b	ali1_t4b_b.txt (automatic)	100	10	0	0	90	35	6
T4b	ali1_t4b_c.txt (automatic)	100	11	3	0	86	65	7

Table 2: Results obtained by our system at QAst

Table 3 shows the comparative of the values obtained for MRR and Accuracy with the manual run and with the three automatic runs. MRR is the Mean Reciprocal Rank measures, that is to say how well ranked is the right answer in the list of 5 possible answers in average, while Accuracy is the fraction of correct answers ranked in the first position in the list of 5 possible answers.
With regard to the automatic transcripts, the best results were obtained with the transcript C versus transcription B (although these are very close to those of B) and transcription C (these are slightly smaller).

Task	Run_ID	# Questions	# Correct _answers (all ranks)	MRR	Accuracy(%)	avg # answers per question
T4a	ali1_t4a.txt (manual)	100	36	0.27	20.0	2.8
T4b	ali1_t4b_a.txt (automatic)	100	16	0.10	7.0	2.6
T4b	ali1_t4b_b.txt (automatic)	100	16	0.12	10.0	3.1
T4b	ali1_t4b_c.txt (automatic)	100	14	0.12	11.0	1.9

Table 3: Results obtained by our system at QAst

In addition, to explain the results obtained, we must not forget the problems arisen in the development of this work. On the one hand, the small size of the corpus, and hence, the consequent low redundancy in them, made difficult to adapt our system. On the other hand, the existence of broad types of questions made not possible to cover them in our system (we only dealt with factual questions).

5 Conclusions

In our first participation in QAst, we have adapted a generic and modular QA system to work with text transcriptions. We want to highlight that its results are above expectation because we did not use any specific resource to deal with automatic transcriptions. Despite using a generalist system, the results are not discouraging. Nevertheless, we want to compare our results with those obtained by the rest of the participants to be able to give an opinion. In the future, we hope to obtain a better system capable of answering questions from the task in a more precise way and we wish to measure the improvements we introduce in our system compared to the state-of-art at the moment.

References

[1] Kenneth W. Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[2] S. Cassidy D. Mollá and M. van Zaanen. Answerfinder at qast 2007: Named entity recognition for qa on speech transcripts. In *QAst at CLEF*, 2007.

[3] M. Heie E.W.D. Whittaker, J.R. Novak and S. Furi. Clef 2007 question answering experiments at tokyo institute of technology. In *QAst at CLEF*, 2007.

[4] Robert Fano. *Transmission of Information.* MIT Press, Cambridge, Massachusetts, 1961.

[5] José Manuel Gómez, Manuel Montes-Gómez, Emilio Sanchis, Luis Villaseñor-Pineda, and Paolo Rosso. Language independent passage retrieval for question answering. In *Fourth Mexican International Conference on Artificial Intelligence MICAI 2005*, Lecture Notes in Computer Science, pages 816–823, Monterrey, Mexico, 2005. Springer-Verlag GmbH.

[6] Sanda M. Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. Falcon: Boosting knowledge for answer engines. In *TREC-9*, 2000.

[7] Eduard H. Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question answering in webclopedia. In *TREC-9*, 2000.

[8] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 1–7, Taipei, Taiwan, August 2002. Association for Computational Linguistics (ACL).

[9] Dekang Lin. Dependency-based evaluation of minipar. In *Workshop on the Evaluation of Parsing Systems*, 1998.

[10] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *EACL*, pages 1–8, 1999.

[11] Dan Moldovan, Sanda M. Harabagiu, Marius Pasca, Rada Mihalcea, Richard Goodrum, Roxana Girju, and Vasile Rus. Lasso: A tool for surfing the answer net. In *Proceedings of The 8th Text REtrieval Conference (TREC 1999)*, Gaithersburg, Maryland, November 1999. National Institute of Standards and Technology (NIST).

[12] D. Mollá-Aliod. AnswerFinder in TREC 2003. In *Proceedings of the 12th Text REtrieval Conference (TREC 2003)*, 2003.

[13] P. Moreda, H. Llorens, E. Saquete, and M. Palomar. Automatic generalization of a qa answer extraction module based on semantic roles. In *Proceedings of the 11th edition of the Ibero-American Conference on Artificial Intelligence*, page To be published, 2008.

[14] P. Moreda, H. Llorens, E. Saquete, and M. Palomar. The influence of semantic roles in qa: A comparative analysis. In *Actas del XXIV Congreso de la SEPLN*, page To be published, 2008.

[15] G. Neumann and R. Wang. Dfki-lt at qast 2007: Adapting qa components to mine answers in speech trancripts. In *QAst at CLEF*, 2007.

[16] J. Turmo P. Comas and M. Surdeanu. Robust question answering for speech transcripts using minimal syntactic analysis. In *QAst at CLEF*, 2007.

[17] L. A. Sangoi Pizzato and D. Mollá-Aliod. Extracting Exact Answers using a Meta Question answering System. In *Proceedings of the Australasian Language Technology Workshop 2005 (ALTW05)*, Sidney, Australia, December 2005.

[18] G. Adda S. Rosset, O. Galibert and E. Bilinski. The limsi participation to the qast track. In *QAst at CLEF*, 2007.

[19] E. Saquete. *Temporal information Resolution and its application to Temporal Question Answering.* Phd, Departamento de Lenguages y Sistemas Informáticos. Universidad de Alicante, June 2005.

[20] Olga Vechtomova and Murat Karamuftuoglu. Lexical cohesion and term proximity in document ranking. *Inf. Process. Manage.*, 44(4):1485–1502, 2008.