INAOE at CLEF 2006: Experiments in Spanish Question Answering

Antonio Juárez-Gonzalez, Alberto Téllez-Valero, Claudia Denicia-Carral
Manuel Montes-y-Gómez, Luis Villaseñor-Pineda

Laboratorio de Tecnologías del Lenguaje
Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Mexico.
{antjug, albertotellezv, cdenicia, mmontesg, villasen}@inaoep.mx

Abstract

This paper describes the system developed by the Language Technologies Lab at INAOE for the Spanish Question Answering task at CLEF 2006. The presented system is centered in a full data-driven architecture that uses machine learning and text mining techniques to identify the most probable answers to factoid and definition questions respectively. Its major quality is that it mainly relies on the use of lexical information and avoids applying any complex language processing resource such as named entity classifiers, parsers or ontologies. Experimental results show that the proposed architecture can be a practical solution for monolingual question answering reaching an answer precision as high as 51%.

1 Introduction

Current information requirements claim for efficient mechanisms capable of interact with users in a more natural way. Question Answering (QA) systems has been proposed as a feasible option for the creation of such mechanisms [1]. Recent developments in QA use a variety of linguistic resources to help in understanding the questions and the documents. The most common linguistic resources include: part-of-speech taggers, parsers, named entity extractors, dictionaries, and WordNet [2, 3, 4, 5]. Despite of the promising results of these approaches, they have two main inconveniences. On the one hand, the construction of such linguistic resources is a very complex task. On the other hand, their performance rates are usually not optimal.

In this paper we present a QA system that allows answering factoid and definition questions. This system is based on a full data-driven approach that requires a minimum knowledge about the lexicon and the syntax of the specified language. It is basically supported on the idea that the questions and their answers are commonly expressed using the same set of words. Therefore, it simply uses lexical information to identify the relevant document passages and to extract the candidate answers.

The prototype presented this year by our group continues our last year work [6]: it is also based on a lexical full data-driven approach. However, it presents two important modifications. First, it applies a supervised approach instead of a statistical method for answering factoid questions. Second, it answers definition questions by applying lexical patterns that were automatically constructed rather manually defined.

The following sections give some details on the proposed system. In particular, section 2 describes the method for answering factoid questions, section 3 explains the method for answering definition questions, and section 4 discusses the results achieved by our system in the Spanish Question Answering task.

2 Answering Factoid Questions

Figure 1 shows the general process for answering factoid questions. It considers three main modules: *passage retrieval*, where the passages with the major probability to contain the answer are recovered from the document collection; *question classification*, where the type of expected answer is determined; and *answer extraction*, where candidate answers are selected using a machine-learning approach, and the final answer recommendation of the system is produced. The following sections describe each of these modules.

2.1 Passage Retrieval

The passage retrieval (PR) method is specially suited for the QA task [7]. It allows retrieving the passages with the highest probability to contain the answer instead of simply recover the passages sharing a subset of words with the question.

Given a user question, the PR method finds the passages with the relevant terms (non-stopwords) using a classical information retrieval technique based on the vector space model. Then, it measures the similarity between

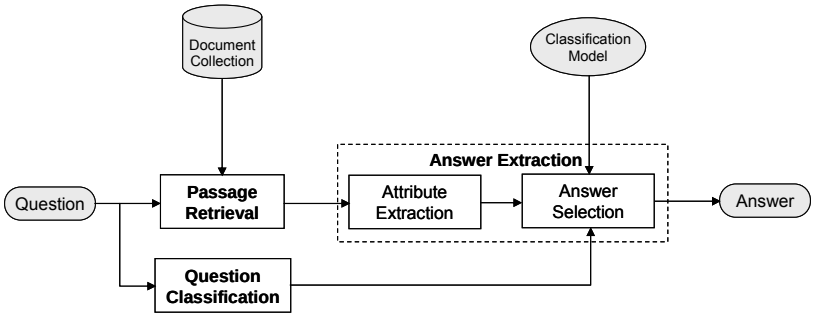

Figure 1. Process for answering factoid questions

the *n*-gram sets of the passages and the user question in order to obtain the new weights for the passages. The weight of a passage is related to the largest *n*-gram structure of the question that can be found in the passage itself. The larger the *n*-gram structure, the greater the weight of the passage. Finally, it returns to the user the passages with the new weights.

3.1.1 Similarity measure

The similarity between a passage *d* and a question *q* is defined by (1).

$$sim(d,q) = \frac{\sum_{j=1}^{n} \sum_{\forall x \in Q_j} h(x(j), D_j)}{\sum_{j=1}^{n} \sum_{\forall x \in Q_j} h(x(j), Q_j)} \tag{1}$$

Where *sim*(*d*, *q*) is a function which measures the similarity of the set of *n*-grams of the question *q* with the set of *n*-grams of the passage *d*. Q_j is the set of *j*-grams that are generated from the question *q* and D_j is the set of *j*-grams of the passage *d*. That is, Q_1 will contain the question unigrams whereas D_1 will contain the passage unigrams, Q_2 and D_2 will contain the question and passage bigrams respectively, and so on until Q_n and D_n. In both cases, *n* is the number of question terms.

The result of (1) is equal to 1 if the longest *n*-gram of the question is in the set of passage *n*-grams.

The function $h(x(j), D_j)$ measures the relevance of the *j*-gram $x(j)$ with respect to the set of passage *j*-grams, whereas the function $h(x(j), Q_j)$ is a factor of normalization[1]. The function *h* assigns a weight to every question *n*-gram as defined in (2).

$$h(x(j), D_j) = \begin{cases} \sum_{k=1}^{j} w_{\hat{x}_k(1)} & \text{if } x(j) \in D_j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Where the notation $\hat{x}_k(1)$ indicates the *k*-th unigram included in the *j*-gram *x*, and $w_{\hat{x}_k(1)}$ specifies the associated weight to this unigram. This weight gives an incentive to the terms –unigrams– that appear rarely in the document collection. Moreover, this weight should also discriminate the relevant terms against those (e.g. stopwords) which often occur in the document collection.

The weight of a unigram is calculated by (3):

$$w_{\hat{x}_k(1)} = 1 - \frac{\log(n_{\hat{x}_k(1)})}{1 + \log(N)} \tag{3}$$

Where $n_{\hat{x}_k(1)}$ is the number of passages in which appears the unigram $\hat{x}_k(1)$, and *N* is the total number of passages in the collection. We assume that the stopwords occur in every passage (i.e., *n* takes the value of *N*). For instance, if the term appears once in the passage collection, its weight will be equal to 1 (the maximum weight), whereas if the term is a stopword, then its weight will be the lowest.

[1] We introduce the notation *x*(*n*) for the sake of simplicity. In this case *x*(*n*) indicates the *n*-gram *x* of size *n*.

2.2 Question Classification

This module is responsible of the definition of the semantic class of the answer expected to respond to the given question. The idea is to know in advance the type of the expected answer in order to reduce the searching space to only those information fragments related this specific semantic class.

Our prototype implements this module following a direct approach based on regular expressions. It only considers three general semantic classes for the type of expected answer: date, quantity and name (i.e., a proper noun).

2.3 Answer Extraction

Answer extraction aims to establish the best answer for a given question. It is based on a supervised machine learning approach. It consists of two main modules, one for attribute extraction and other one for answer selection.

Attribute extraction. First, the set of recovered passages are processed. The purpose is to identify all text fragments related to the semantic class of the expected answer. This process is done using a set of regular expression that allows identifying proper names, dates and quantities. Each identified text fragment is considered a "candidate answer".

In a second step, the lexical context of each candidate answer is analyzed with the aim of constructing its formal representation. In particular, each candidate answer is represented by a set of 17 attributes, clustered in the following groups:

1. Attributes that describe the complexity of the question. For instance, the length of the question (number of non-stopwords).
2. Attributes that measure the similarity between the context of the candidate answer and the given question. Basically, these attributes considers the number of common words, word lemmas and named entities (proper names) between the context of the candidate answer and the question. They also take into consideration the density of the question words in the answer context.
3. Attributes that indicate the relevance of the candidate answer in accordance with the set of recovered passages. For instance, the relative position of passage that contains the candidate answer as well as the redundancy of the answer in the whole set of passages.

Answer Selection. This module selects from the set of candidate answers the one with the maximum probability of being the correct answer. This selection is done by a machine learning method, in particular, by a Naïve Bayes classifier.

It is important to mention that the classification model (actually, we have three classifiers, one for each kind of answer) was constructed using as a training set the questions and documents from previous CLEFs.

3 Answering Definition Questions

Figure 2 shows the general scheme of our method for answering definition questions[2]. It consists of three main modules: a module for the discovery of definition patterns, a module for the construction of a general definition catalog, and a module for the extraction of the candidate answer. The following sections describe in detail these modules.

It is important to mention that this method is specially suited for answering definition questions as delimited in the CLEF. That is, questions asking for the position of a person, e.g., Who is Vicente Fox?, and for the description of concept, e.g., What is the CERN? or What is Linux?.

It is also important to notice that the processes of pattern discovery and catalog construction are done offline, while the answer extraction is done online, and that different to traditional QA approaches, the proposed method does not consider any module for document or passage retrieval.

3.1 Pattern Discovery

The module for pattern discovery uses a small set of concept-description pairs to collect from the Web an extended set of definition instances. Then, it applies a text mining method on the collected instances to discover a set of definition surface patterns. The idea is to capture the definition conventions through their repetition. This module considers two main subtasks:

[2] This method is an adaptation of the one previously proposed in [8].

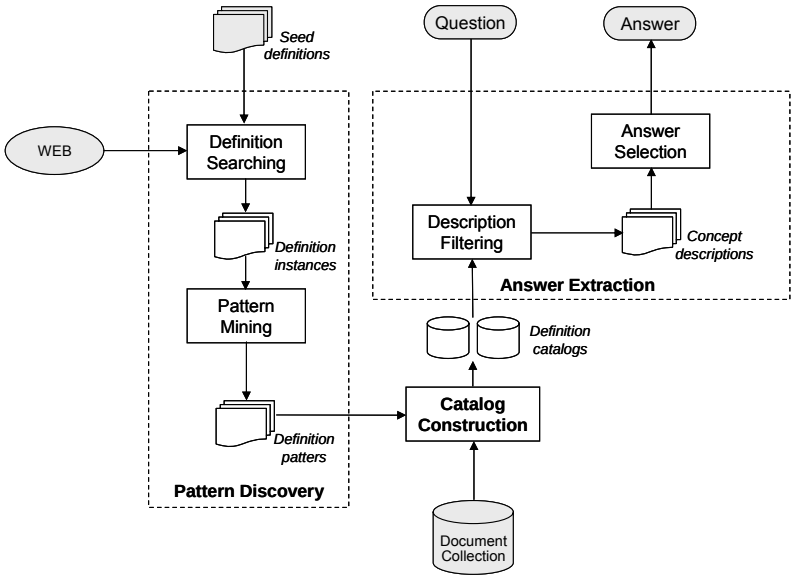

Figure 2. Process for answering definition questions

Definition searching. This task is triggered by a small set of empirically defined concept-description pairs. The pairs are used to retrieve a number of usage examples from the Web[3]. Each usage example represents a definition instance. To be relevant, a definition instance must contain the concept and its description in one single phrase.

Pattern mining. It is divided in three main steps: data preparation, data mining and pattern filtering. The purpose of the data preparation phase is to normalize the input data. It transforms all definition instances into the same format using special tags for the concepts and their descriptions. It also indicates with a special tag the concepts expressing proper names.

In the data mining phase, a sequence mining algorithm [9] is used to obtain all maximal frequent sequences of words, punctuation marks and tags from the set of definition instances. The sequences express lexicographic patterns highly related to concept definitions.

Finally, the pattern-filtering phase allows choosing the more discriminative patterns. It selects the patterns satisfying the following general regular expressions:

```
        <left-string> DESCRIPTION <middle-string> CONCEPT <right-string>
        <left-string> CONCEPT <middle-string> DESCRIPTION <right-string>
  <left-string> DESCRIPTION <middle-string> PROPER_NAME_CONCEPT <right-string>
  <left-string> PROPER_NAME_CONCEPT <middle-string> DESCRIPTION <right-string>
         <left-string> DESCRIPTION <middle-string> PROPER_NAME_CONCEPT
        PROPER_NAME_CONCEPT <middle-string> DESCRIPTION <right-string>
                <left-string> DESCRIPTION PROPER_NAME_CONCEPT
               PROPER_NAME_CONCEPT DESCRIPTION <right-string>
```

Figure 3 illustrates the information treatment through the pattern discovery process. The idea is to obtain several surface definition patterns starting up with a small set of concept-description example pairs. First, using a small set of concept description seeds, for instance, "*Wolfgang Clement – German Federal Minister of Economics and Labor*" and "*Vicente Fox – President of Mexico*", we obtained a set of definition instances. One example of these instances is "*…meeting between the Cuban leader and the president of Mexico, Vicente Fox.*". Then, the instances were normalized, and finally a sequence-mining algorithm was used to obtain some lexical patterns highly related to concept definitions. The figure shows two example patterns: "*, the <DESCRIPTION>, <CONCEPT>, says*" and "*the <DESCRIPTION> <PROPER_NAME_CONCEPT>*". It is important to notice that the discovered patterns may include words, punctuation marks as well as proper name tags as frontier elements.

[3] At present we are using Google for searching the Web.

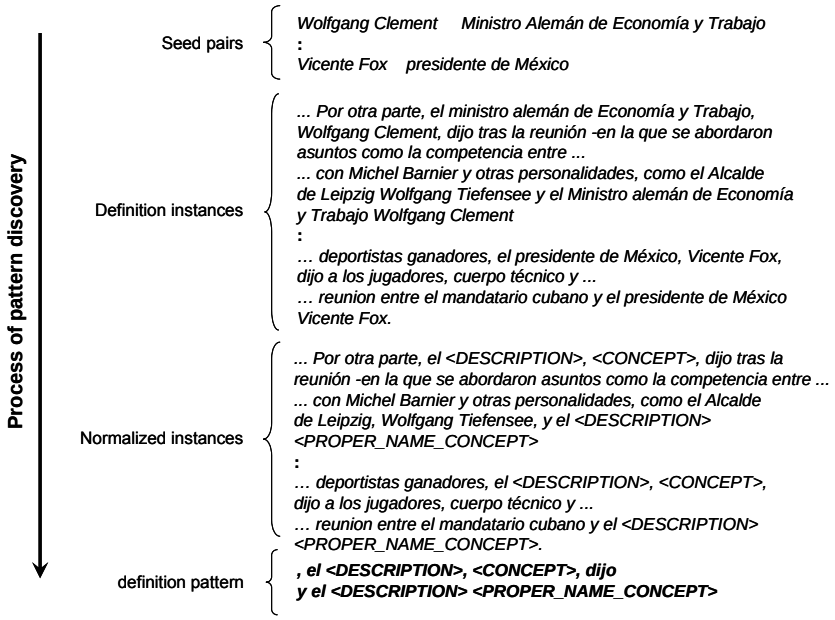

Figure 3. Data flow in the pattern discovery process

3.2 Catalog Construction

In this module, the definition patterns discovered in the previous stage (i.e., in the pattern discovery module) are applied over the target document collection. The result is a set of matched text segments that presumably contain a concept and its description. The definition catalog is created gathering all matched segments.

3.3 Answer Extraction

This module handles the extraction of the answer for a given definition question. Its purpose is to find the more adequate description for a requested concept from the definition catalog. The definition catalog may contain a huge diversity of information, including incomplete and incorrect descriptions for many concepts. However, it is expected that the correct information will be more abundant than the incorrect one. This expectation supports the idea of using a frequency criterion and a text mining technique to distinguish between the adequate and the improbable answers to a given question. This module considers the following steps:

Description filtering. Given a specific question, this procedure extracts from the definition catalog all descriptions corresponding to the requested concept. As we mentioned, these "presumable" descriptions may include incomplete and incorrect information. However, it is expected that many of them will contain, maybe as a substring, the required answer.

Answer selection. This process aims to detect a single answer to the given question from the set of extracted descriptions. It is divided in two main phases: data preparation and data mining.

The data preparation phase focuses on homogenizing the descriptions related to the requested concept. The main action is to convert these descriptions to a lower case format. In the data mining phase, a sequence mining algorithm [9] is used to obtain all maximal frequent word sequences from the set of descriptions. Then, the most frequent sequence is selected as the correct answer.

Figure 4 shows the process of answer extraction for the question "*Who is Diego Armando Maradona?*". First, we obtained all descriptions associated with the requested concept. It is clear that there are erroneous or incomplete descriptions (e.g. "*Argentina soccer team*"). However, most of them contain a partially satisfactory explanation of the concept. Actually, we detected correct descriptions such as "*captain of the Argentine soccer team*" and "*Argentine star*". Then, a mining process allowed detecting a set of maximal frequent sequences. Each sequence was considered a candidate answer. In this case, we detected three sequences: "*argentine*", "*captain of the Argentine soccer team*" and "*supposed overuse of Ephedrine by the star of the Argentine team*". Finally, the candidate answers were ranked based on the frequency of occurrence of its subsequences in the whole description set. In this way, we took advantage of the incomplete descriptions of the concept. The selected answer was

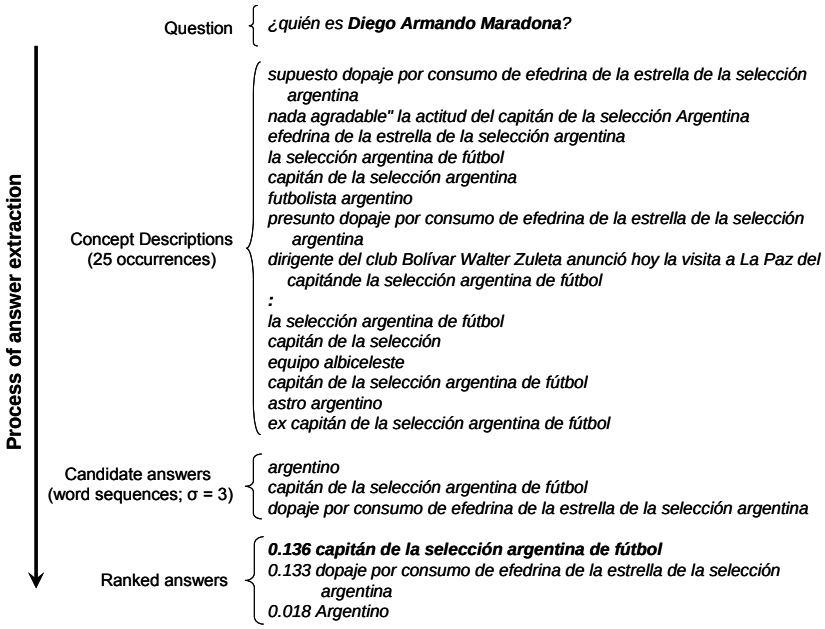

Figure 4. Data flow in the answer extraction process

"*captain of the Argentine national football soccer team*", since it was conformed from frequent subsequences such as "*captain of the*", "*soccer team*" and "*Argentine*".

It is important to clarify that a question may have several correct answers. In accordance with the CLEF, an answer is correct if there is a passage that supports it. Therefore, for the question at hand there are other correct answers such as "*ex capitán de la selección argentina de futbol*" and "*astro argentino*".

4 Evaluation Results

This section describes the experimental results related to our participation at QA@CLEF2006 monolingual track for Spanish. It is important to remember that this year the question type (e.g., factoid, definition, temporal or list) was not included as a data field on the question test file. Therefore, each participant had to automatically determine the kind of question.

Our system prototype, as described in the previous sections, only can deal with factoid and definition questions. In particular, from the 200 test questions, it treats 144 as factoid questions and the rest of them as definition questions. Table 1 details our results on answering factoid questions.

Table 1. Accuracy on answering factoid questions (run *inao061eses*)

	Overall Evaluation	**Evaluation by answer type**		
		Quantity	**Date**	**Name**
Right	59	13	9	37
Wrong	75	13	10	52
Inexact	2	0	1	1
Unsupported	8	0	5	3
Accuracy	**40.9%**	50%	36%	39.7%

On the other hand, the method for answering definition questions was used to respond 56 questions; from them 28 questions asked for the position of a person (*who questions*) and 28 asked for the description of a concept (*what questions*). Table 2 resumes the assessed results from this kind of questions.

Table 2. Accuracy on answering definition questions (run *inao061eses*)

	Overall Evaluation	**Person's Positions**	**Concept's Descriptions**
Right	43	19	24
Wrong	11	7	4
Inexact	1	1	0
Unsupported	1	1	0
Accuracy	**76.7%**	67.8%	85.7%

In addition to the outstanding results obtained by this method, it was very interesting to notice that it replies very exact answers most of the times. Nevertheless, it has the inconvenience of constructing an enormous definition catalog (1,772,918 for concept expansion and 3,525,632 for persons positions) containing a huge quantity of incorrect/incomplete registers. This characteristic was the origin of most of our wrong answers, since noisy information was more redundant that correct one.

Lastly, it is important to mention that the overall evaluation of this year exercise (51%) was 10-points over our last year result [8].

5 Conclusions and Future Work

This paper presented a question answering system that allows answering factoid and definition questions. This system is based on a lexical data-driven approach. Its main idea is that the questions and their answers are commonly expressed using almost the same set of words, and therefore, it simply uses lexical information to identify the relevant passages as well as the candidate answers.

The answer extraction for factoid questions is based on a machine learning method. Each candidate answer (uppercase word, date or quantity) is represented by a set of lexical attributes and a classifier determines the most probable answer for the given question. The method achieved good results, however it has two significant disadvantages: (*i*) it requires a lot of training data, (*ii*) the detection of the candidate answers is not always (not for all cases, nor for all languages) an easy –high precision– task.

On the other hand, the answer extraction for definition questions is based on a text mining approach. The proposed method uses a text mining technique (namely, a sequence mining algorithm) to discover a set of definition patterns from the Web as well as to determine –with finer precision– the answer to a given question. The achieved results were especially good, and they evidenced that a non-standard QA approach, which does not contemplate an IR phase, can be a good scheme for answering definitions questions.

As future work we plan to improve the final answer selection by applying an answer validation method. The purpose is to reduce the dependence of our current methods to the answer redundancy.

Acknowledgements. This work was done under partial support of CONACYT (Project Grants: 43990 and U39957-Y). We also like to thanks to the CLEF organizing committee as well as to the EFE agency for the resources provided.

References

1. Magnini B., Vallin A., Ayache C., Erbach G., Peñas A., de Rijke M., Rocha P., Simov K. and Sutcliffe R., *Overview of the CLEF 2004 Multilingual Question Answerig Track*, In Working notes for the Cross Language Evaluation Forum Workshop (CLEF 2004), Bath, UK, September 2004.
2. Roger S., Ferrández S., Ferrández A., Peral J., Llopis F., Aguilar A. and Tomás D., *AliQAn, Spanish QA System at CLEF-2005*, In Working notes for the Cross Language Evaluation Forum Workshop (CLEF 2005), Vienna, Austria, September 2005.
3. Gómez-Soriano J.M., Bisbal-Asensi E., Buscaldi D., Rosso P. and Sanchos-Arnal E., *Monolingual and Cross-language QA using a QA-oriented Passage Retrieval System*, In Working notes for the Cross Language Evaluation Forum Workshop (CLEF 2005), Vienna, Austria, September 2005.

4. de-Pablo-Sánchez C., González-Ledesma A., Martinez-Fernández J.L., Guirao J.M., Martinez P. and Moreno A., *MIRACLE's 2005 Approach to Cross-Lingual Question Answering*, In Working notes for the Cross Language Evaluation Forum Workshop (CLEF 2005), Vienna, Austria, September 2005.
5. Ferrés D. Kanaan S., González E., Ageno Al, Rodríguez H. and Turmo J., *The TALP-QA System for Spanish at CLEF-2005*, In Working notes for the Cross Language Evaluation Forum Workshop (CLEF 2005), Vienna, Austria, September 2005.
6. Montes-y-Gómez M., Villaseñor-Pineda L., Pérez-Coutiño M., Gómez-Soriano J.M., Sanchis-Arnal E. and Rosso P., *INAOE-UPV Joint Participation at CLEF 2005: Experiments in Monolingual Question Answering*, In Working notes for the Cross Language Evaluation Forum Workshop (CLEF 2005), Vienna, Austria, September 2005.
7. Gómez-Soriano J.M., Montes-y-Gómez M., Sanchis-Arnal E., Villaseñor-Pineda L. and Rosso P., *Language Independent Passage Retrieval for Question Answering*, In Proceedings for the Fourth Mexican International Conference on Artificial Intelligence (MICAI 2005), Monterrey, Nuevo León, México, November 2005.
8. Denicia-Carral C., Montes-y-Gómez M., Villaseñor-Pineda L. and García-Hernández R., *A Text Mining Approach for Definition Question Answering*, to appear in Proceedings for the Fifth International Conference on Natural Language Processing (FinTal 2006), Turku, Finland, August 2006.
9. García-Hernández R., Martínez-Trinidad F. and Carrasco-Ochoa A., *A New Algorithm for Fast Discovery of Maximal Sequential Patterns in a Document Collection*, in Proceedings for the Seventh International Conference on Computational Linguistics and text Processing (CICLing 2006), Mexico City, Mexico, February 2006.