The Use of Lexical Context in Question Answering for Spanish

M. Pérez-Coutiño, T. Solorio, M. Montes-y-Gómez[†],
A. López-López and L. Villaseñor-Pineda

Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Luis Enrique Erro No. 1, Sta Ma Tonantzintla, 72840, Puebla, Pue, México.
{mapco,thamy,mmontesg,allopez,villasen}@inaoep.mx

Abstract. This paper describes the prototype developed by the Language Technologies Laboratory at INAOE for Spanish monolingual QA evaluation task at CLEF 2004. Our approach is centered in the use of context at a lexical level in order to identify possible answers to factoid questions. Such method is supported by an alternative one based on pattern recognition in order to identify candidate answers to definition questions. The methods applied at different stages of the system and prototype architecture for question answering are described. The paper shows and discusses the results achieved with this approach.

Keywords: Question Answering for Spanish, Lexical Context, Natural Language Processing.

1 Introduction

Question Answering (QA) systems has become an alternative to traditional information retrieval systems because of its capability to provide concise answers to questions stated by the user in natural language. This fact, along with the inclusion of QA evaluation as part of the Text Retrieval Conference (TREC)[1] in 1999, and recently [6] in Multilingual Question Answering as part of the Cross Language Evaluation Forum (CLEF)[2], have arisen a promising and increasing research field.

The Multilingual Question Answering evaluation track at CLEF 2004 is similar to last year edition. For each subtask, participants are provided with 200 questions requiring short answers. Some questions may not have known answer, and systems should be able to recognize them. However there are some important differences, this year answers included fact based instances or definitions, and systems must return exactly one response per question, and up to two runs.

Our laboratory has developed a prototype system for Spanish monolingual QA task. Two important things to consider about it are, a) this is our first QA prototype and has been developed from scratch, and b) this the first time that our laboratory participates in an evaluation forum.

The prototype described in this document relies in the fact that several approaches of QA systems like [8, 13, 4, 10] use named entities at different stages of the system in order to find a candidate answer. Generally speaking, the use of named entities is performed at the final stages of the system, i.e., either in the passage selection or as a discriminator in order to select a candidate answer at the final stage. Another interesting approach is the use of *Predictive Annotation* which was first presented at TREC-8 by Prager et al. [8]. One meaningful characteristic of this approach is the indexing of anticipated semantic types, identifying the semantic type of the answer sought by the question, and extracting the best matching entity in candidate answer passages. In their approach, the authors used no more than simple pattern matching to get the entities. Our prototype was developed to process both, questions and source documents in Spanish. Our system is based on approach just described but differs in the following: i) Semantic classes' identification relies in the preprocessing of the whole document collection by a POS tagger that simultaneously works as named entity recognizer and classifier. ii) The indexing stage takes as item the lexical context associated to each single named entity contained in every document of the collection. iii) The searching stage selects as candidate answers those named entities whose lexical contexts match better the context of the question. iv) At the final stage, candidate answers are compared against a second set of candidates gathered from the Internet. v) Final answers are selected based on a set of relevance measures which encompass all the information collected in the searching process.

The rest of this paper is organized as follows; section two describes the architecture and functionality of the system; section three details the process of question processing; section four details the process of indexing; section five shows the process of searching; section six describe the process of answer selection; section seven discusses the results achieved by the system; and finally section eight exposes our conclusions and discusses further work.

[†] This work was done while visiting the Dept. of Information Systems and Computation Polytechnic University of Valencia, Spain.

[1] http://trec.nist.gov/

[2] http://clef-qa.itc.it/

2 System Overview

The system adjusts to a typical QA system architecture [14]. Figure 1 shows the main blocks of the system. The system could be divided into the following stages: *question processing*, which involves the extraction of named entities and lexical context in the question, as well as question classification to define the semantic class of the answer expected to respond to the question; *indexing*, where a preprocessing of the supporting document collection is done, building the representation of each document that become the searching space to find candidate answers to the question; *searching*, where a set of candidate answers is obtained from the index and the Internet, (here candidate answers are classified by a machine learning algorithm, and provides information to perform different weighting schemes); and finally *answer selection* where candidate answers are ranked and the final answer recommendation of the system is produced. Next sections describe each of these stages.

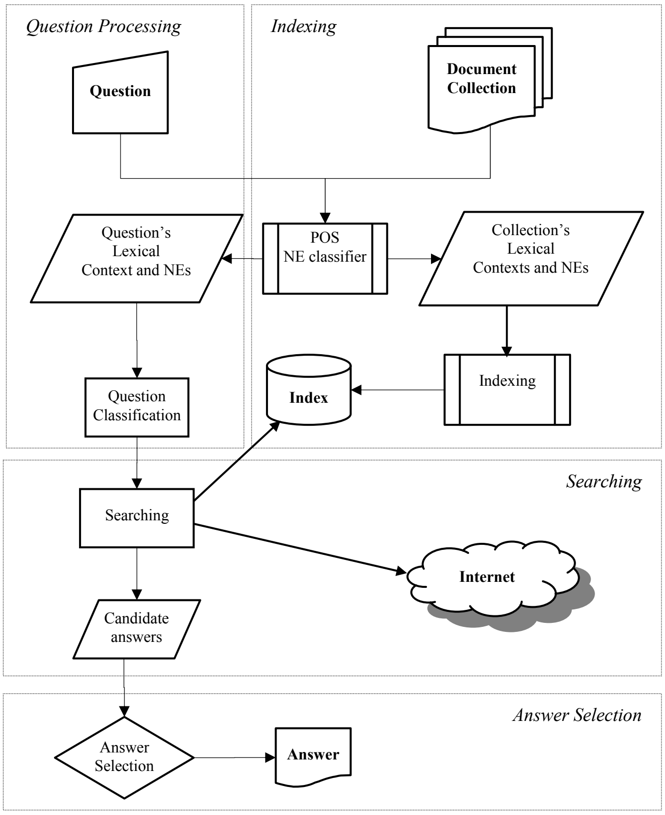

Figure 1. Block diagram of the system.
There are four stages: question processing, indexing, searching and answer selection.

3 Question Processing

MACO [3] is a POS tagger and lemmatizer capable of recognizing and classifying named entities (NEs). The possible categories for NEs are the following: person, organization, geographic place, date, quantity and miscellaneous. In order to reduce the possible candidate answers provided by our system we perform a question classification process. The purpose of this classification is to match each question with one of the six named entities provided by MACO.

We use a straightforward approach, where the attributes for the learning task are the prefixes of the words in the question and additional information acquired by an Internet search engine.

The procedure for gathering this information from Internet is first we use a set of heuristics in order to extract from the question the first noun word or words *w*. We then employ a search engine, in this case Google, submitting queries using the word *w* in combination with the five possible semantic classes. For instance, for the question *Who is the President of the French Republic?* President is extracted as the noun in the question using our heuristics, and run 5 queries in the search engine, one for each possible class. The queries take the following forms:

- "President is a person"
- "President is a place"
- "President is a date"
- "President is a measure"
- "President is an organization"

For each query *(q_i)* the heuristic takes the number of results *(Cr_i)* returned by Google and normalizes them according to equation 1. This means that for each question, the summatory of their five performed queries is 1. Normalized values *($Iw(q_i)$)* are taken as attributes values for the learning algorithm. As it can be seen is a very direct approach, but experimental evaluations showed that this information gathered from Internet is quite useful [11].

$$Iw(q_i) = Cr_i \Big/ \sum_{i=0}^{n} Cr_i \qquad \text{Equation 1.}$$

The machine learning technique used was Support Vector Machines [12] implemented in WEKA [15]. Section 7 discusses the performance of question classification process.

4 Indexing

Each document in the collection is modeled by the system as a factual text object whose content refers to several named entities even when it is focused on a central topic. As mentioned, named entities could be one of these objects: persons, organizations, locations, dates, quantities and miscellaneous. The model assumes that the named entities are strongly related to their lexical context, especially to nouns (subjects) and verbs (actions). Thus, a document can be seen as a set of entities and their contexts. For details about the document model we refer the reader to [7]. In order to obtain the representation of the documents, the system begins preprocessing each document with MACO, where this process is performed off-line. Once the document collection has been tagged, the system extracts the lexical contexts associated to named entities. The context considered for this experiment consists of the four verbs or nouns, both at the left and right of its corresponding NE (table 1 shows a sample). The final step in the indexing stage is the storage of the extracted contexts, populating a relational database[3] which preserves several relations between each named entity, its semantic class, associated contexts, and the documents where they appeared. In other words, the index is an adaptation of the well knows inverted file structure used in several information retrieval systems.

Table 1 Context associated to named entity "CFC".
Verbs and common nouns in cursive are gathered from a preprocessing with a POS tagger.

<DOCNO>EFE19941219-11009</DOCNO>
… Los CFC son usados en los productos anticongelantes, de insuflación y como *refrigerantes*, que *tienen* al *cloro* como un *ingrediente* común. "Los CFC *son* los *responsables* del *agujero* de la *capa* de ozono",…

5 Searching

The search engine developed for the system and the searching process differ in several aspects from traditional search engines. This process relies on two information sources: first the information gathered from question processing, i.e., the expected semantic class of the answer to the question, and the named entities and lexical context of the question; and second, the index of named entities, contexts and documents created during indexing.

5.1 Searching Algorithm

With the document representation, all the name entities mentioned in a given document can be known beforehand. Thus, the name entities from the question become key elements in order to define the document set more likely to provide the answer. For instance, in the question *"¿Dónde se entregan los Oscar?"*, the named entity "Oscar" narrows the set of documents to only those containing such name entity. At the same time, another assumption is that the context in the neighborhood of the answer has to be similar to the lexical context of the question. Once more, from the question of the example, the fragment *"...reciben esta noche, en la sexagésimasexta edición de los Oscar, el homenaje de Hollywood…"* contains a lexical context next to the answer which is similar to that of the question.

[3] Due to performance constraints, the index has been distributed over a cluster of 5 CPUs.

Following is the algorithm in detail:

1. Identify the set of relevant documents according to the named entities in the question.
2. Retrieve all contexts in each relevant document.
3. Compute the similarity between question context and those obtained in step 2.
    - 3.1. Preserve only those contexts whose associated named entity corresponds to the semantic class of the question.
    - 3.2. Compute a similarity function based on frequencies to perform further ranking and answer selection. This function is based on the number of question's named entities found in each pair *(NE,Context)* retrieved and the number of similar terms in both contexts.
4. Rank the candidate named entities in decreasing order of similarity.
5. Store similarity and named entity classification information (step 3.2) for next stage.

6 Answer Selection

Analyzing the output from the local index we find out that we had a lot of possible answers with the same values for similarity and named entity classification information. Thus, we develop a method for selecting the final possible answer based on answers retrieved from Internet and automated classification of answers using a bagged ensemble of J48 [15].

The final answer presented by our system was selected by calculating the intersection among words between the local index candidate answers and the answers provided by the Internet search. We consider the candidate answer with highest intersection value to be more likely to be the correct answer. However, in some cases all the candidate answers have the same intersection values. In this case we selected from the candidates the first one classified by the learning algorithm as belonging to the positive class. When no positive answer was found among the candidates for a question, then we selected the first candidate answer with highest value from the local index.

The following sections briefly describe the Internet search and the answer classification processes.

6.1 Internet Searching

As mention earlier, at the final stage, the system uses information from the Internet in order to get more evidence of the possible accuracy of each candidate answer. From the perspective of the overall system, Internet searching occurs simultaneously to the local search. This subsection reviews the process involved in such task.

The module used at this step was originally developed at our laboratory to research the effectiveness of a statistical approach to web question answering in Spanish. Such approach lies in the concept of redundancy in the web, i.e, the module applies a several transformations in order to convert the question into a typical query and then this query along to some query reformulations are sent to a search engine with the hypothesis that the answer would be contained –several times– in the snippets retrieved by the search engine[4]. The selection of candidate answers from Internet is based on computing all the n-grams, from unigrams to pentagrams, as possible answers to the given question. Then, using some statistical criteria the n-grams are ranked by decreasing likelihood of being the correct answer. The top ten are used to validate the candidates gathered from the local searching process.

6.2 Answer Classification

Discriminating among possible answers was posed as a learning problem. Our goal was to train a learning algorithm capable of selecting from a set of possible candidates the answer that most likely satisfies the question. We selected as features the values computed by the local indexing. We use five attributes: 1) the number of times the possible answer was labeled as the entity class of the question; 2) the number of times the possible entity appeared labeled as a different entity class; 3) number of words in common in the context of the possible answer and the context of the question, excluding named entities; 4) the number of entities that matched the entities in the question, and 5) the frequency of the possible answer along the whole collection of documents. With these attributes, we then trained a bagged ensemble of classifiers using as base learning algorithm the rule induction algorithm J48 [9].

In this work we build the ensemble using the bagging technique which consists of manipulating the training set [1].

Given that we had available only one small set of questions, we evaluate the classification process in two parts. We divided the set of questions into two subgroups of the same size and performed two runs. In each run, we trained on one half and tested on the other.

[4] The search engine used by this module is Google (http://www.google.com)

6.3 Answering Definitions

Due to the length and elements in a definition answer, our approach considers the treatment of these questions as a special case. In order to reach accurate definition answers, we have implemented a set of heuristics able to find patterns like those described in [10]. Table 2 shows some samples of applying such heuristics.

The heuristics are based in punctuation and some stopwords (articles, pronouns and prepositions) which provide evidence for identification of pairs *<Answer><Name>*. Thus could be easily gathered by regular expressions.

Table 2 Examples of definition questions and their answers.

Question	Text fragment containing the answer
¿Quién es Arabella Kiesbauer?	…otra carta-bomba dirigida, al parecer, a una conocida periodista austriaca de raza negra, *Arabella Kiesbauer*, y que fue enviada desde Austria...
¿Qué es UNICEF?	Naciones Unidas, 3 ene (EFE).- El Fondo de las Naciones Unidas para la Infancia (*UNICEF*), formuló hoy, lunes, una petición…
¿Quién es Andrew Lack?	…Tanto es así, que el presidente del departamento de noticias de la cadena NBC, *Andrew Lack*, confesó en una entrevista…

7 Evaluation

We participate in the evaluation with two runs. The first one *inao041eses* was gathered applying all components of the system, while our second run *inao042eses* didn't make use of heuristics for definition answers. Table 3 shows our results.

Table 3 Results of submitted runs.

Run	*inao041eses*	*inao042eses*
Right	45	37
Wrong	145	152
ineXact	5	6
Unsupported	5	5
Overall Accuracy	22.50%	18.50%
Factoid Questions	19.44%	17.78%
Definition Questions	50%	25%
"NIL" Accuracy	19.61%	21.74%

Results show that overall performance of the system with evaluation questions was over 60% lower than training results. A preliminary analysis of the errors has let us note some obvious mistakes like the length and elements considered as part of the context (which couldn't be fixed before questions' release due to time constraints). However there are some other issues to take into account besides those errors, for instance, question classification. Figure 2 shows the accuracy of the classifier, from a total of 200 questions, the classifier only can assign an accurate semantic class to 157 questions, which represents a precision of 78.5%. Besides classifier performance, searching and candidate answers selection were also very low, only 29.41% of questions right classified as person were answered, 63.63 % of organizations, 39.10% of localities, 37.50% of dates, 28.57% of quantities and 18.18% of miscellaneous were answered.

We have begun a detailed analysis looking for inconsistencies in the overall approach, as well as programming bugs. The initial step is to get an improved configuration of the POS tagger and NE classifier (MACO) in order to label the corpus and rebuild our indexes (databases) with a non restricted version of document model, i.e. without pre-established elements and length in the context. Thus repeat some experiments with a refinement method for candidates and answer selection.

8 Conclusions

This work has presented a lexical-context approach for QA in Spanish. The strength of this work lies in the model used for the source documents. The identification and annotation in advance of named entities and their associated contexts serves as key information in order to select possible answers to a given factoid question. On the other hand, the discrimination of candidate answers is a complex task that requires more research and experimentation of different methods. In this work we have experimented with the merging of evidence coming from three main sources: a ranked list of candidate answers gathered by a similarity measure, answer classification by a bagged ensemble of classifiers, and a set of candidate answers gathered from the Internet. Further work includes exploring the inclusion of more information as part of the context, the refinement of the semantic classes for questions and named entities, and the improvement of answer selection methodology.

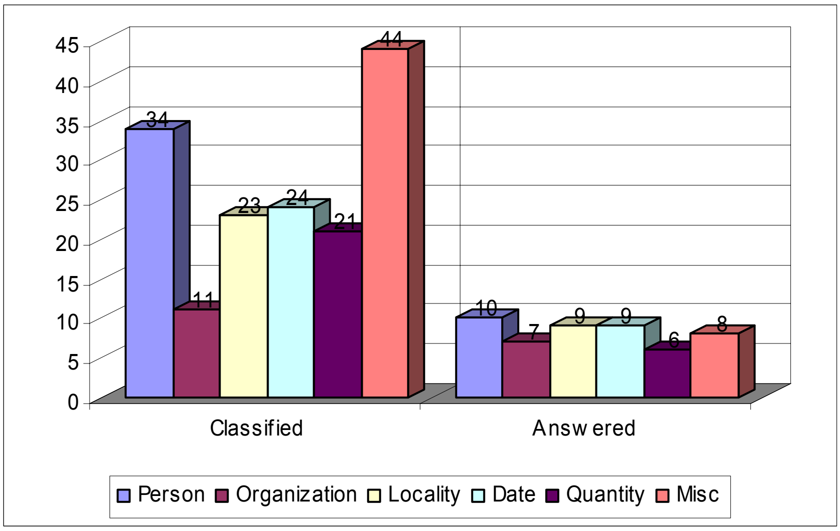

Figure 2. Question classifier accuracy. Numbers in data labels refers to total number of questions classified or answered.

Acknowledgements. This work was done under partial support of CONACYT (Project Grants U39957-Y and 43990), SNI-Mexico, and the Human Language Technologies Laboratory of INAOE. We also like to thanks to the CLEF as well as EFE agency for the resources provided.

References

1. Breiman L. *Bagging predictors. Machine Learning*, 24(2):123-140, 1996.
2. Burger, J. et al. *Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A).* NIST 2001.
3. Carreras, X. and Padró, L. *A Flexible Distributed Architecture for Natural Language Analyzers*. In Proceedings of the LREC'02, Las Palmas de Gran Canaria, Spain, 2002.
4. Cowie J., et al., *Automatic Question Answering*, Proceedings of the International Conference on Multimedia Information Retrieval (RIAO 2000)., 2000.
5. Hirshman L. and Gaizauskas R. *Natural Language Question Answering: The View from Here*, Natural Language Engineering 7, 2001.
6. Magnini B., Romagnoli S., Vallin A., Herrera J., Peñas A., Peinado V., Verdejo F. and Rijke M. *The Multiple Language Question Answering Track at CLEF 2003*. CLEF 2003 Workshop, Springer-Verlag.
7. Pérez-Coutiño M., Solorio T., Montes-y-Gómez M., López-López A. and Villaseñor-Pineda L., *Toward a Document Model for Question Answering Systems.* In Advances in Web Intelligence. LNAI3034 Springer-Verlag 2004.
8. Prager J., Radev D., Brown E., Coden A. and Samn V. *The Use of Predictive Annotation for Question Answering in TREC8*. NIST 1999.
9. Quinlan J. R. *C4.5: Programs for machine learning.* 1993. San Mateo, CA: Morgan Kaufmann.
10. Ravichandran D. and Hovy E. *Learning Surface Text Patterns for a Question Answering System*. In ACL Conference, 2002.

11. Solorio T., Pérez-Coutiño M., Montes-y-Gómez M., Villaseñor-Pineda L., and López-López A. 2004. *A language independent method for question classification*. In COLING-04. 2004. Switzerland.
12. Vapnik, V. *The Nature of Statistical Learning Theory*, Springer, 1995.
13. Vicedo, J.L., Izquierdo R., Llopis F. and Muñoz R., *Question Answering in Spanish*. CLEF 2003 Workshop, Springer-Verlag.
14. Vicedo, J.L., Rodríguez, H., Peñas, A. and Massot, M. Los sistemas de Búsqueda de Respuestas desde una perspectiva actual. Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural, n.31, 2003.
15. Witten H. and Frank E. 1999. *Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann.