COLE at CLEF 2004: Rapid prototyping of a QA system for Spanish

Enrique Méndez Díaz
Jesús Vilares Ferro
David Cabrero Souto
Departamento de Computación
Universidade da Coruña
Campus de Elviña s/n
15071 La Coruña (Spain)
{jvilares,cabrero}@udc.es
http://www.grupocole.org

Abstract

This is our third participation in CLEF, this time in the Spanish monolingual Question Answering track. We have continued applying Natural Language Processing techniques for single word conflation. Our approach for Question Answering is based on complex pattern matching either over forms, part-of-speech tags or lemmas of the words involved.

1 Introduction

In past editions of CLEF, our research group has participated in the Spanish monolingual Information Retrieval (IR) track [23, 22], applying Natural Language Processing (NLP) techniques to conflate the documents to be indexed. In these past participations, our main premise has been the simplicity, motivated by the lack of freely available linguistic resources for Spanish such as large tagged corpora, treebanks or advanced lexicons.

This year, in this our first participation in the Spanish monolingual Question Answering (QA) track, our premise keeps being the same in order to get a valid prototype which will be improved by continuous refinements. As usual, in our QA system we have identified three tasks: analysis of the question, retrieval of the passages of the documents related to the question and identification of the exact fragment of the document that constitutes the answer. Thus, this paper should be read as a progress report. Our research in QA is in an early stage and much work has to be done. It should be remarked that another serious drawback is the lack of freely available linguistic resources for Spanish.

This article is outlined as follows. Section 2 introduces the NLP techniques we have used in our prototype. After that, section 3 describes the overall design of the prototype and then the different modules of the system are described in subsequent sections: the analysis of questions, the information retrieval module and the answer delimitation process are detailed in subsections 3.1, 3.2 and 3.3 respectively. Finally, our conclusions and future work are presented in section 3.3.

2 NLP processing

In this section we introduce the NLP techniques used as basis for our QA prototype, focused in dealing with inflectional variation. Both in Information Retrieval and Query Answering systems, one of the major limitations we have to deal with is the *linguistic variation* of natural languages [4]. When managing this type of phenomena, the employment of Natural Language Processing techniques becomes feasible. This has been our working hypothesis since our research group started its work on Spanish Information

Retrieval time ago, and it keeps being our working hypothesis now we have started working on Spanish Query Answering.

Our proposal in this our first participation in the Spanish QA track, consists on the employment of lemmatization for solving the *inflectional variation* of documents instead of classical approaches such as stemming.

The effectiveness of *stemming* is dependent on the morphology of the language, this way, when processing languages with complex morphology and a high number of irregularities the performance of stemmers becomes irregular [4, 7]. In the case of Spanish, there exist inflectional modifications at multiple levels (gender and number for nouns and adjectives, and person, mood, time and tense for verbs) and with many irregularities [24]: for nouns and adjectives, more than 20 variation groups for gender inflection and more than 10 variation groups for number inflection have been identified; for verbs, 3 regular groups and almost 40 irregular groups have been identified, each group containing more than 100 inflected forms. This level of complexity cannot be managed only through stemming. Moreover, stemming can also cause problems for NLP systems by causing the loss of information needed in further processing [16], as in the case of Query Answering.

This way, lemmatization shows itself as an advisable alternative to stemming, since it can manage properly these complex phenomena of Spanish with no losses of information. The encouraging results obtained in the Spanish monolingual IR track [23, 22] support this choice.

The lemmatization process is performed in two steps: a first phase of preprocessing and a second phase of part-of-speech tagging and lemmatization, properly speaking.

2.1 Preprocessing

One of the most important prior tasks in NLP is *text segmentation*, the task of dividing a text into linguistically meaningful units —words (*tokenization*) and sentences (*sentence segmentation*)—, since the words and sentences identified at this stage are the fundamental units passed to further processing stages, such as part-of-speech taggers, Information Retrieval systems, Question Answering systems, etc [18]. Nevertheless, this stage is often obviated in many current applications, which assume that input texts are already segmented correctly in *tokens* or high level information units. This working hypothesis is not realistic due to the heterogeneous nature of the application texts and their sources, and it results in erroneous behaviors during further processing.

This way, preprocessing is an indispensable task in practice, and it can involve processes which are much more complex than the simple identification of the different sentences in the text and each of their individual components. For this reason, we have developed a linguistically-motivated preprocessor module for Spanish [10, 5] in order to perform tasks such as format conversion, tokenization, sentence segmentation, morphological pretagging, contraction splitting, separation of enclitic pronouns from verbal stems, expression identification, numeral identification and proper noun recognition.

2.2 Tagging and lemmatization

Once the text has been preprocessed, the output generated by our preprocessor —the words and sentences which form the text— is then taken as input by our tagger-lemmatizer, `MrTagoo` [8], although any similar high-performance tool could be used instead. `MrTagoo` is based on a second order Hidden Markov Model (HMM), whose elements and procedures of estimation of parameters are based on Brant's work [6], and also incorporates certain capabilities which led to its use in our system. Such capabilities include a very efficient structure for storage and search —based on finite-state automata [9]—, management of unknown words, the possibility of integrating external dictionaries in the probabilistic frame defined by the HMM [11], and the possibility of managing ambiguous segmentations [12]

Nevertheless, these kind of tools are very sensitive to spelling errors, as, for example, in the case of sentences written completely in uppercase —e.g., news headlines and subsection headings—, which cannot be correctly managed by the preprocessor and tagger modules. For this reason, when documents are processed in order to be indexed, the initial output of the tagger is processed by an *uppercase-to-lowercase* module [23] in order to process uppercase sentences, converting them to lowercase and restoring the diacritical marks when necessary.

3 Architecture

The overall architecture of our prototype is composed of three main modules: question processing, realted passage retrieval and answer extraction. The first module analyzes the query obtaining a list of keywords, then the next module takes that list and performs a mostly conventional information retrieval process obtaining a list of paragraphs expected to contain the answer. Finally, the last module takes such paragraphs and extracts the answer from them. At the first stages of the prototype we are focusing on question processing and information retrieval for serveral reasons:

- Simplicity is a premise.
- Once the system is capable of returning to the user a paragraph contaning the right answer, the average user will find the system satisfactory.
- If you cannot find the paragraph containing the answer, you cannot extract it.

3.1 Question Processing

For question processing we are usign some kind of simplified shallow parsing [3]. This parsing is made at two levels: lemmatization and pattern matching. The result of the pattern matching phase is a list of keywords to be used in order to search relevant documents.

This way, the first step of the process consists on tagging and lemmatazing the question usign our *preprocessor* and out tagger-lemmatizer, Mr Taggo, as it has been previously described in section 2. Once the question has been tagged and lemmatized, the keyword selection process is performed by means of pattern matching. In a previous study we have identified different categories of questions, such as:

```
> Quién ser ... ? / Who be ... ?
> Quién ... ?     / Who ... ?
> Dónde ... ?     / Where ... ?
```

Each category has associated a list of patterns composed of tags and/or words. For each taggged question, the system goes through this list of patterns till one of them matches and the keywords matched are extracted.

Our first prototype uses all the keywords extracted from the query. This approach showed a poor performance, since only in 25% of the cases leaded to the retrieval of paragraphs containing the answer. To overcome this problem, our next prototype will reduce the specificity of the querys by removing useless elements from the list of keywords.

3.2 Passage retrieval

At this stage of the process, the system performs a mostly conventional IR task on the set of available documents in order to retrive the portions of documents supposed to contain the answer. As usual this requires that the documents were indexed before the system becomes operative.

In order to identify the candidate documents which are relevant to a given question in which we will look for the answer, a *Passage Retrieval* (PR) approach has been used [14, 17] in order to delimit not only the relevant document but also the relevant portion of text. This way, documents are splitted into passages made up by three sentences, with an overlap factor of two sentences[1].We found that using passage retrieval instead of document retrieval overcomes two main disadvantages:

- The search engine would find as relevants documents that containts most of the keywords of our query, even when those keywords are sparse in the document. That situation probably means that the document does not contain the answer. On the other hand, when keywords are close enough the answer will be eventually in the same part of the document.
- It is more difficult to extract the answer from a document than from a small part of it.

[1]That is, first passage contains sentence 1 to 3, second passage contains from sentence 2 to 4, and so on.

As in our previous contributions to CLEF IR Spanish Monolingual Track [23, 22], text is conflated through lemmatization in order to solve the problems derived from inflection in Spanish. This way, once text has been tagged and lemmatized, the lemmas of the *content words* [13] —nouns, verbs and adjectives— are extracted to be indexed, since they contain the main semantics of the text [13, 15]. Before indexing, the terms obtained are converted to lowercase and their spelling signs are eliminated in order to reduce typographical errors.

The resulting conflated text is indexed using the probabilistic engine ZPrise [1], employing the Okapi BM25 weight scheme [19] with the constants defined in [20] for Spanish ($b = 0.5$, $k_1 = 2$). The stopword list used was obtained by lemmatizing the content words of the Spanish stopword list provided with the well-known indexing engine SMART [2].

3.3 Answer extraction

The answer extraction module takes the list of paragrahs retrived by the previuos module and tries to extract the answer to the question formulated by the user. Currently, this module is quite naive and simply tries to find a coherent answer near the keywords extracted from the question. Work is in progress in order to improve this module. We achive this goal we intend to develop several methods to extract the answer. Each method will select some answer candidates and a vote system will be used to choose the best one. The methods currently scheduled are:

- At first module determine the answer type and use that information to select the probable answer.
- Use word distances. A suitable implemtation [21] is in progress.

Conclusions and future directions

We have built a small prototype using the tools created for IR tasks and new ones specifically developed for QA tasks. As expected for an early prototype, it is far from optimal, showing an irregular performance. However we find the desgin architecture good enough. Regarding to the modules, futher experimentation with the question processign module has showed that our approach works fine, but new improvements are desirable. Regarding to the passage retrieval module, NLP techniques proved quite usefull once again. Finally, the answer extraction module needs futher research and new approaches in order to get satisfactory results. Also a new approach, based on the employment of a locality-based retriebal model [21] is being considered in order to locate the relevant portion of the docuement with a higher degree of precision.

Acknowledgements

The research reported in this article has been partially supported by Ministerio de Ciencia y Tecnología (HF2002-81), FPU grants of Secretaría de Estado de Educación y Universidades (AP2001-2545), Xunta de Galicia (PGIDIT02PXIB30501PR and PGIDIT02SIN01E) and Universidade da Coruña.

References

[1] `http://www.itl.nist.gov/iaui/894.02/works/papers/zp2/zp2.html` (site visited August 2004).

[2] `ftp://ftp.cs.cornell.edu/pub/smart` (site visited August 2004).

[3] Steven Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1997.

[4] Avi Arampatzis, Th. P. van der Weide, P. van Bommel, and C.H.A. Koster. Linguistically-motivated information retrieval. In *Encyclopedia of Library and Information Science*, volume 69, pages 201–222. Marcel Dekker, Inc, New York-Basel, 2000.

[5] Fco. Mario Barcala, Jesús Vilares, Miguel A. Alonso, Jorge Graña, and Manuel Vilares. Tokenization and proper noun recognition for information retrieval. In *3rd International Workshop on Natural Language and Information Systems (NLIS 2002), September 2-3, 2002. Aix-en-Provence, France*, Los Alamitos, California, USA, 2002. IEEE Computer Society Press.

[6] Thorsten Brants. TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP'2000)*, Seattle, WA., 2000.

[7] Carlos G. Figuerola, Raquel Gómez, Angel F. Zazo Rodríguez, and José Luis Alonso Berrocal. Stemming in Spanish: A first approach to its impact on information retrieval. In Carol Peters, editor, *Results of the CLEF 2001 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2001 Workshop, 3 September, Darmstadt, Germany*, 2001.

[8] Jorge Graña. *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*. PhD thesis, Departamento de Computación, Universidade da Coruña, A Coruña, Spain, December 2000.

[9] Jorge Graña, Fco. Mario Barcala, and Miguel A. Alonso. Compilation methods of minimal acyclic automata for large dictionaries. In Bruce W. Watson and Derick Wood, editors, *Proc. of the 6th Conference on Implementations and Applications of Automata (CIAA 2001)*, pages 116–129, Pretoria, South Africa, July 2001.

[10] Jorge Graña, Fco. Mario Barcala, and Jesús Vilares. Formal methods of tokenization for part-of-speech tagging. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 240–249. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[11] Jorge Graña, Jean-Cédric Chappelier, and Manuel Vilares. Integrating external dictionaries into stochastic part-of-speech taggers. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nocolov, and Nokolai Nikolov, editors, *EuroConference Recent Advances in Natural Language Processing. Proceedings*, pages 122–128, Tzigov Chark, Bulgaria, 2001.

[12] Jorge Graña Gil, Miguel A. Alonso Pardo, and Manuel Vilares Ferro. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. In P. Sojka, I. Kopeček, and K. Pala, editors, *Text, Speech and Dialogue*, volume 2448 of *Lecture Notes in Computer Science*, pages 3–10. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[13] Christian Jacquemin and Evelyne Tzoukermann. NLP for term variant extraction: synergy between morphology, lexicon and syntax. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, volume 7 of *Text, Speech and Language Technology*, pages 25–74. Kluwer Academic Publishers, Dordrecht/Boston/London, 1999.

[14] M. Kaszkiel and J. Zobel. Effective ranking with arbitrary passages. *Journal of the American Society of Information Science*, 52(4):344–364, 2001.

[15] Cornelis H. A. Koster. Head/modifier frames for information retrieval. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2945 of *Lecture Notes in Computer Science*, pages 420–432. Springer-Verlag, Berlin-Heidelberg-New York, 2004.

[16] Gerald Kowalski. *Information Retrieval Systems: Theory and Implementation*. The Kluwer international series on Information Retrieval. Kluwer Academic Publishers, Boston-Dordrecht-London, 1997.

[17] Fernando Llopis, José L. Vicedo, and Antonio Ferrández. IR-n system at CLEF-2002. In Carol Peters, Martin Braschler, Julio Gonzalo, and Martin Kluck, editors, *Advances in Cross-Language Information Retrieval*, volume 2785 of *Lecture Notes in Computer Science*, pages 291–300. Springer-Verlag, Berlin-Heidelberg-New York, 2003.

[18] David D. Palmer. *Handbook of Natural Language Processing*, chapter Tokenisation and Sentence Segmentation. Marcel Dekker, Inc., New York & Basel, 2000.

[19] S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In E. M. Voorhees and D. K. Harman, editors, *NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8)*, pages 151–162, Gaithersburg, MD, USA, 2000. Department of Commerce, National Institute of Standards and Technology.

[20] Jacques Savoy. Report on CLEF-2002 Experiments: Combining Multiple Sources of Evidence. In Carol Peters, editor, *Results of the CLEF 2002 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2002 Workshop, 19-20 September, Rome, Italy*, pages 31–46, 2002.

[21] Jesús Vilares and Miguel A. Alonso. Dealing with syntatic variation through a locality-based approach. In *To be published*, Lecture Notes in Computer Science. Springer-Verlag, Berlin-Heidelberg-New York, 2004.

[22] Jesús Vilares, Miguel A. Alonso, and Francisco J. Ribadas. COLE experiments at CLEF 2003 Spanish monolingual track. In Carol Peters, Martin Braschler, Julio Gonzalo, and Martin Kluck, editors, *Advances in Cross-Language Information Retrieval*, Lecture Notes in Computer Science. Springer-Verlag, Berlin-Heidelberg-New York, 2004.

[23] Jesús Vilares, Miguel A. Alonso, Francisco J. Ribadas, and Manuel Vilares. COLE experiments at CLEF 2002 Spanish monolingual track. In *Advances in Cross-Language Information Retrieval*, volume 2785 of *Lecture Notes in Computer Science*, pages 265–278. Springer-Verlag, Berlin-Heidelberg-New York, 2003.

[24] Manuel Vilares, Jorge Graña, and Pilar Alvariño. Finite-state morphology and formal verification. *Journal of Natural Language Engineering, special issue on Extended Finite State Models of Language*, 3(4):303–304, 1997.