A Grammatical Approach to the Extraction of Index Terms*

Jesús Vilares and **Miguel A. Alonso**
Universidade da Coruña
Campus de Elviña s/n
15071 La Coruña
{jvilares,alonso}@udc.es

Abstract

The extraction of the keywords that characterize each document in a given collection is one of the most important components of an Information Retrieval system. In this article, we propose to apply shallow parsing, implemented by means of cascades of finite-state transducers, to extract complex index terms based on an approximate grammar of Spanish. The effectiveness of the index terms extracted has been evaluated through the CLEF collection.

1 Introduction

Our previous works (Vilares *et al.* 01; Vilares *et al.* 02) have showed the feasibility of the employment of Natural Language Processing (NLP) techniques in Spanish Information Retrieval (IR) in order to manage word-level linguistic variation due to inflection and derivation. The next logical step consists of applying phrase-level analysis techniques in order to, on the one hand, manage the *syntactic variation*, and on the other hand, to obtain more precise index terms. Nevertheless, at this point, we have to face the problems derived from the high computational cost of parsing. With the purpose of maintaining lineal complexity with respect to the length of the text to be analyzed, we have discarded the employment of full parsing, opting for applying *shallow parsing* techniques, also looking for more robustness. During the parsing process we will extract the syntactic dependencies of the text, using the words involved in such dependencies as index terms.

Given a context-free grammar and an input string, the syntactic trees of height k generated by a parser can be obtained by means of k layers of finite-state transducers: the first layer obtains the nodes labeled by non-terminals corresponding to left-hand sides of productions that only contain terminals on their right-hand side; the second layer obtains those nodes which only involve terminal symbols and those non-terminal symbols generated on the previous layer; and so on. Of course, we are limiting the height of the trees, thus limiting the kind of syntactic structures we can recognize. Nevertheless, this kind of shallow parsing (Abney 97) has shown itself to be useful in several NLP application fields, particularly in Information Extraction. Its application in IR has not been deeply studied, and has often been limited to the analysis of simple noun phrases (Kraaij & Pohlmann 98; Hull *et al.* 97).

* Supported in part by Ministerio de Ciencia y Tecnología (TIC2000-0370-C02-01, HP2001-0044 y HF2002-81), FPU grants of Secretaría de Estado de Educación y Universidades, Xunta de Galicia (PGIDT01PXI10506PN, PGIDIT02PXIB30501PR and PGIDIT02SIN01E) and Universidade da Coruña.

2 Shallow parsing

Our system is based on a five layer architecture which takes as its input the output of a tagger-lemmatizer. In the rest of this section we will describe how each layer works. For this purpose, we will use as our notation context-free rules extended with classical regular expression operators: ? denoting optionality, * denoting repetition with optionality (0 or more times), and | for separating alternatives. In the same way, uppercase identifiers denote a set of terms, either preterminals (tags resulting from part-of-speech tagging) or elements of a given grammatical category. When requiring the presence of a concrete lemma, it will be indicated by using the `typewriter` font.

2.1 Layer 0

In order to minimize the noise generated during the subsequent parsing steps, certain constructions are preprocessed:

- *Quantity expressions and numerals in non-numerical format.* Expressions of the type *algo más de dos millones* (a little more than two million) are identified as *numeral phrases* ($NumP$).

- *Verbal expressions.* Some verbal expressions must be considered as a unit in order to sim-

plify the work of the upper layers. This way, the expression *tener en cuenta* (to take into account), for example, must be considered as a unit, synonym of the verb *considerar* (to consider), to avoid *en cuenta* being identified as a complement of the verb by the upper layers, impeding the correct identification of other complements of interest.

2.2 Layer 1

This layer consists of rules only containing tags and/or lemmas in its right-hand side. For the next layers to be able to extract syntactic dependency pairs, we will associate to the non-terminal in the left side of each rule, firstly, the lemma corresponding to the head of the phrase we are recognizing, and secondly, the tag with the appropriate morphosyntactic features. The notation employed for this inheritance mechanism is inspired in the notation employed when specifying the set of restrictions in feature structure-based grammars, as is now shown in the following rules.

This first rule allows us to identify sequences of adverbs (W), called *adverbial phrases* ($AdvP$). The last adverb will be considered the phrase head, so its lemma and its tag will be the lemma and tag of the non-terminal $AdvP$:

$$AdvP \rightarrow W^* \ W_1 \left\{ \begin{array}{l} AdvP.lem \doteq W_1.lem \\ AdvP.tag \doteq W_1.tag \end{array} \right.$$

The following set of rules allows us to identify *first level verbal groups* ($VG1$) corresponding to passive forms[1], whether simple tenses, e.g. *soy observado* (I am observed), or compound tenses[2], e.g. *he sido observado* (I have been observed). The first of these rules manages compound forms: the tag is taken from the auxiliary verb `haber` (to have), whereas the lemma is taken from the main verb, which must be a participle, the same as the auxiliary verb `ser` (to be). The second rule manages simple forms: the tag is obtained from the form of the auxiliary verb `ser`, whereas the lemma is taken from the main verb, again a participle.

$$VG1 \rightarrow V_1 \ V_2 \ V_3 \left\{ \begin{array}{l} VG1.lem \doteq V_3.lem \\ VG1.tag \doteq V_1.tag \\ VG1.voice \doteq \text{PASS} \\ V_1.lem \doteq \texttt{haber} \\ V_2.lem \doteq \texttt{ser} \\ V_2.tense \doteq \text{PART} \\ V_3.tense \doteq \text{PART} \end{array} \right.$$

$$VG1 \rightarrow V_1 \ V_2 \left\{ \begin{array}{l} VG1.lem \doteq V_2.lem \\ VG1.tag \doteq V_1.tag \\ VG1.voice \doteq \text{PASS} \\ V_1.lem \doteq \texttt{ser} \\ V_2.tense \doteq \text{PART} \end{array} \right.$$

[1] Constructed with the auxiliary verb `ser` (to be).

[2] Constructed with the auxiliary verb `haber` (to have).

Active forms, both compound and simple, are identified in a similar way.

2.3 Layer 2

Adjectival phrases and periphrastic verbal groups are processed in this layer. An adjectival phrase ($AdjP$) is that whose head is an adjective, which may be preceded by an adverbial phrase:

$$AdjP \rightarrow AdvP? \ A \left\{ \begin{array}{l} AdjP.lem \doteq A.lem \\ AdjP.tag \doteq A.tag \end{array} \right.$$

Second level verbal groups ($VG2$) are also managed in this layer, including periphrastic verbal groups. With respect to its structure, a periphrasis is generally formed by a conjugated auxiliary verb giving the inflection, a verb in a non-personal form (infinitive, gerund or participle) giving the main meaning, and an optional element (preposition or conjunction) linking both verbs.

Infinitive periphrases are identified using the following rule, which takes into account the possibility that the auxiliary verb may be followed by an enclitic pronoun (previously separated from the verb form by the tagger) when the first verb is reflexive. The tag is inherited from the auxiliary verb, while the lemma and the voice are inherited from the main verb:

$$VG2 \rightarrow VG1_1 \ (\texttt{me}|\texttt{te}|\texttt{se})? \ (\texttt{que}|\texttt{de}|\texttt{a})? \ VG1_2 \left\{ \begin{array}{l} VG2.lem \doteq VG1_2.lem \\ VG2.tag \doteq VG1_1.tag \\ VG2.voice \doteq VG1_2.voice \\ VG1_1.voice \doteq \text{ACT} \\ VG2_2.tense \doteq \text{INF} \end{array} \right.$$

Gerund and participle periphrases are managed in a similar way, whereas first level verbal groups which do not take part in any periphrastic group are promoted to second level verbal groups.

2.4 Layer 3

Noun phrases (NP) are processed in this layer. During the definition of the rules for their management, we have taken into account the possibility of their being preceded by a *partitive complement* (PC) such as `alguno de` (some of), `ninguno de` (none of), etc.

Following the head of the noun phrase, there may appear a modifier consisting of two adjectival phrases coordinated by a conjunction (Cc), or consisting of a sequence of one, two or even three adjectival phrases:

$$\begin{array}{lcl} AdjPostModif & \rightarrow & AdjP \ \ Cc \ \ AdjP \\ AdjPostModif & \rightarrow & AdjP \\ AdjPostModif & \rightarrow & AdjP \ \ AdjP \\ AdjPostModif & \rightarrow & AdjP \ \ AdjP \ \ AdjP \end{array}$$

The head of the noun phrase is formed by a common noun (N), an acronym or a proper noun;

its tag and lemma will decide the tag and lemma of the whole phrase. In the case of several candidates for head appearing, we will take the last one. The tag of the phrase may be modified in the presence of a partitive complement, because in this case, and in order to establish the concordances with other phrases, the number of the noun phrase will be inherited from the complement. For example, we must say *Cualquiera de ellos lo sabe*[3] but not **Cualquiera de ellos lo saben*[4].

Optionally, we may find one or more determiners (D) and an adjectival phrase before the head. The existence of adjectival post-modifiers is also optional, and thus we finally obtain the rule:

$$\begin{array}{rcl} NP & \rightarrow & PC? \\ & & D^* \ \ (AdjP \mid Number \mid NumP)? \\ & & (N \mid Acronym \mid Proper)^* \\ & & (N \mid Acronym \mid Proper)_1 \\ & & AdjPostModif? \\ & & \left\{ \begin{array}{l} NP.lem \doteq ()_1.lem \\ NP.tag \doteq ()_1.tag \\ NP.number \doteq PC.number \end{array} \right. \end{array}$$

2.5 Layer 4

The last layer is in charge of the identification of *prepositional phrases* (PP, $PPof$, $PPby$), those formed by a noun phrase (NP) preceded by a preposition (P). To make the extraction of dependencies easier, we will distinguish from the rest those phrases introduced by the prepositions `de` (of) and `por` (by), producing the following rules:

$$PPof \rightarrow P \ \ NP \left\{ \begin{array}{l} P.lem \doteq \texttt{de} \\ PP.lem \doteq NP.lem \\ PP.tag \doteq NP.tag \end{array} \right.$$

$$PPby \rightarrow P \ \ NP \left\{ \begin{array}{l} P.lem \doteq \texttt{por} \\ PP.lem \doteq NP.lem \\ PP.tag \doteq NP.tag \end{array} \right.$$

$$PP \rightarrow P \ \ NP \left\{ \begin{array}{l} PP.lem \doteq NP.lem \\ PP.tag \doteq NP.tag \end{array} \right.$$

3 Extraction of dependencies

In our system, the final goal of parsing is the extraction of pairs of words related through syntactic dependencies. This process is developed in two phases: a first phase of *identification of the syntactic roles* of the phrases identified during the analysis, and a second phase of *extraction of dependencies*, strictly speaking.

[3]Literally translated as *Any of them knows it*, which is incorrect in English but correct in Spanish.

[4]Literally translated as *Any of them know it*, which is correct in English but incorrect in Spanish.

The syntactic roles identified by the system, and the criteria used for it, are the following:

Prepositional noun complement. Due to the ambiguity in the attachment of prepositional phrases, we can not guarantee whether we are analyzing a complement of a noun or a complement of a verb, so we will only take into account the prepositional $PPof$ phrases introduced by `de` (of), because of their high reliability.

Subject. The closest noun phrase (NP) preceding a verbal group ($VG2$) will be considered its subject. We will also consider that those verbal groups whose head is a non-personal form (infinitive, gerund or participle) do not have a subject.

Attribute. In presence of a copulative verb, we will identify as its attribute that non-attached $AdjP$ or that head of a $NP/PPof$ closest to the verbal group.

Direct object. The closest NP after an active predicative $VG2$ will be considered as its object.

Agent. The closest $PPby$ following a passive predicative $VG2$ will be considered as its agent.

Prepositional verb complement. Due to the problem of the prepositional phrase attachment, we will only identify as a prepositional verb complement that prepositional phrase following the verb, closest to it, and previous to any attribute or verb complement identified before.

Once we have identified the syntactic roles of the phrases obtained by the parser, the next phase consists of the *extraction of the syntactic dependencies* existing between them. For this reason, the system will create the pairs formed by:

- A noun and each of its modifying adjectives.
- A noun and the head of its prepositional complement.
- The head of the subject and its predicative verb.
- The head of the subject and the head of the attribute. Copulative verbs are mere links, from a semantical point of view, so the dependency is directly established between the subject and the attribute.
- An active verb and the head of its object.
- A passive verb and the head of its agent.
- A predicative verb and the head of its prepositional complement.

- The head of the subject and the head of a prepositional complement of the verb, but only when it is copulative (because of its special behavior).

Once the dependencies have been extracted and conflated, they are employed as index terms. In our case, we have used a conflation technique based on the employment of morphological relations in order to improve the management of syntactic and morphosyntactic variation (Vilares *et al.* 02b; Jacquemin & Tzoukermann 99).

4 Evaluation

Our approach has been tested using the Spanish monolingual corpus of the 2001 and 2002 CLEF editions (Peters 02), composed of 215.738 news reports provided by EFE, a Spanish news agency. The 100 queries employed, from 41 to 140, consist of three fields: a brief *title* statement, a one-sentence *description*, and a more complex *narrative* specifying the relevance assessment criteria. All these three fields have been used in our experiments, but giving double relevance to the *title*, because it summarizes the basic semantics of the query. Documents were indexed with the vector-based engine SMART (Buckley 85), using the `atn-ntc` weighting scheme.

Previous experiments (Vilares *et al.* 02; Vilares *et al.* 02b) indicate that lemmatization is the best starting point for the development of NLP-based conflation methods for managing more complex linguistic variation phenomena. Thus, we will take lemmatization as our point of reference.

Table 1 shows the results obtained. The first column indicates the results for lemmatization (*lem*). The next columns, *sdx*, contain the results obtained when merging lemmatized simple terms and complex terms based on syntactic dependencies (*sd*), when the weight relation between simple and complex terms, x to 1, changes —i.e. when the weight of simple terms is multiplied by x. The column *opt* is formed by the best results obtained with *sd* for each parameter considered, which are also highlighted in bold. Finally, the column Δ shows the improvement of *opt* with respect to *lem*. The performance of the system for each of these techniques is measured using the parameters contained in each row: number of documents retrieved, number of relevant documents retrieved (5548 expected), R-precision, average precision (non-interpolated) for all relevant documents (averaged over queries), average document precision for all relevant documents (averaged over relevant documents), and precision at N documents retrieved.

As can be seen in column *sd1*, the direct employment of syntactic dependencies as index terms has led to a general decrease of the performance of the system. After examining the behavior of the system for each query, we inferred that the problem was caused by an over-balance of the weight of complex terms, which are much less frequent than simple terms and, therefore, with a much higher assigned weight. This situation leads to a growing instability of the system, because when undesired matchings of complex terms with non-relevant documents occurs, their assigned scores increase substantially, and their relevances grow excessively. At the same time, and also due to the same reason, when correct matchings between complex terms and relevant documents occur, we obtain a clear improvement of the results with respect to the employment of simple terms only. It can be argued that, according to this, we would expect similar results to those obtained only with simple terms. Nevertheless it should be noticed that complex term matchings are much less frequent than those for simple terms. Therefore, fortuitous matchings of complex terms are much more harmful than those for simple terms, whose effect tends to be weakened by the rest of the matchings.

In this way, we need to solve this over-balance of complex terms in order to minimize the negative effect of undesired matchings. To do so, we corrected the balance factor between the weights of simple and complex terms, decreasing the extra initial relevance assigned to complex terms, as is shown in the remaining *sdx* columns. The improvement obtained with this solution is immediate, particularly with respect to the precision in the first 15 documents retrieved and to the number of relevant documents retrieved (5220 with *lem*, 5214 with *sd1*, and 5250 with *sd2*).

As generally happens in IR, we can not talk about a best method for all situations. From a ranking point of view, *sd4*, in which the weights of simple terms are quadrupled, obtains the best results, also reaching the best recall (5252 relevant docs retrieved). Nevertheless, the best results for global performance measures[5] are obtained with

[5] R-precision, average precision (non-interpolated) for

	lem	*sd1*	*sd2*	*sd3*	*sd4*	*sd5*	*sd6*	*sd7*	*sd8*	*opt*	Δ
Documents	99k	99k	99k	99k	99k	99k	99k	99k	99k	- -	- -
Relevant (5548 expected)	5220	5214	5250	**5252**	**5252**	5248	5249	5244	5242	5252	32
R-precision	.5131	.4806	.5041	.5137	.5175	.5174	.5200	**.5203**	.5197	.5203	.0072
Non-interpolated precision	.5380	.5085	.5368	.5440	.5461	.5462	.5464	**.5472**	.5463	.5472	.0092
Document precision	.5924	.5489	.5860	.5974	.6013	.6025	**.6028**	.6026	.6020	.6028	.0104
Precision at 5 docs.	.6747	.6525	**.6909**	.6869	.6848	.6788	.6808	.6828	.6808	.6909	.0162
Precision at 10 docs.	.6010	.5859	.6091	.6192	**.6202**	.6192	.6192	.6172	.6152	.6202	.0192
Precision at 15 docs.	.5623	.5441	.5690	.5737	.5778	**.5791**	**.5791**	.5764	.5758	.5791	.0168
Precision at 20 docs.	.5374	.5040	.5298	.5328	.5354	.5343	.5384	**.5394**	.5384	.5394	.0020
Precision at 30 docs.	.4825	.4549	.4778	.4852	.4892	.4886	.4882	**.4896**	**.4896**	.4896	.0071
Precision at 100 docs.	.3067	.2873	.3017	.3070	.3084	**.3095**	.3087	.3089	.3083	.3095	.0028
Precision at 200 docs.	.2051	.1959	.2033	.2057	.2062	.2063	**.2067**	**.2067**	.2065	.2067	.0016
Precision at 500 docs.	.0997	.0980	.0997	.1001	.1004	**.1005**	**.1005**	**.1005**	**.1005**	.1005	.0008
Precision at 1000 docs.	.0527	.0527	.0530	**.0531**	**.0531**	.0530	.0530	.0530	.0529	.0531	.0004

Table 1: Experiments using the CLEF corpus

sd7, which uses a higher balance factor.

5 Conclusions and future work

Throughout this article we have proposed the employment of syntactic dependencies as complex index terms, in an attempt to solve the problems derived from syntactic and morphosyntactic linguistic variation, and, in this way, to obtain more precise terms. To extract such dependencies, we have developed a shallow parser for Spanish based on a cascade of finite-state transducers, which allows us to face the processing of big collections in a fast and robust way. The results we have obtained are encouraging, though our problem continues to be how to incorporate the syntactic information obtained with the parser in our indexes.

With respect to our future work, we expect that the application of automatically acquired *selections restrictions* (Gamallo *et al.* 01) using the texts themselves would let us improve the disambiguation capability of the system, particularly in the case of the attachment of prepositional phrases. We are also considering the possibility of storing simple and complex terms in separate indexes, combining them afterwards by means of *data fusion* techniques.

References

(Abney 97) S. Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1997.

(Buckley 85) C. Buckley. Implementation of the SMART information retrieval system. Technical report, CS Dept., Cornell University, 1985. Source code: `ftp://ftp.cs.cornell.edu/pub/smart`.

(Gamallo *et al.* 01) P. Gamallo, A. Agustini, and G. P. Lopes. Selections restrictions acquisition from corpora. In volume 2258 of *Lecture Notes in Artificial Intelligence*, pages 30–43, Berlin-Heidelberg-New York, 2001. Springer-Verlag.

(Hull *et al.* 97) D. A. Hull, G. Grefenstette, B. M. Schulze, E. Gaussier, H. Schutze, and J. O. Pedersen. Xerox TREC-5 site report: routing, filtering, NLP, and Spanish tracks. In *Proc. of the Fifth Text REtrieval Conference (TREC-5)*, pages 167–180, 1997.

(Jacquemin & Tzoukermann 99) C. Jacquemin and E. Tzoukermann NLP for term variant extraction: synergy between morphology, lexicon and syntax. In volume 7 of *Text, Speech and Language Technology*, pages 25–74. Kluwer Academic Publishers, Dordrecht-Boston-Londres, 1999.

(Kraaij & Pohlmann 98) W. Kraaij and R. Pohlmann. Comparing the effect of syntactic vs. statistical phrase indexing strategies for Dutch. In volume 1513 of *Lecture Notes in Computer Science*, pages 605–614. Springer-Verlag, Berlin-Heidelberg-New York, 1998.

(Peters 02) C. Peters, editor. *Working Notes for the CLEF 2002 Workshop*, Rome, Italy, 2002.

(Vilares *et al.* 01) J. Vilares, D. Cabrero, and M. A. Alonso. Applying productive derivational morphology to term indexing of Spanish texts. In volume 2004 of *Lecture Notes in Computer Science*, pages 336–348. Springer-Verlag, Berlin-Heidelberg-New York, 2001.

(Vilares *et al.* 02) J. Vilares, M. A. Alonso, F. J. Ribadas, and Manuel Vilares. COLE experiments at CLEF 2002 Spanish monolingual track. In (Peters 02), pages 153–160.

(Vilares *et al.* 02b) J. Vilares, F. M. Barcala, and M. A. Alonso. Using syntactic dependency-pairs conflation to improve retrieval performance in Spanish. In volume 2276 of *Lecture Notes in Computer Science*, pages 381–390. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

all relevant documents and average document precision for all relevant documents.