COLE Experiments at CLEF 2003 Spanish Monolingual Track

Jesús Vilares[1], Miguel A. Alonso[1], and Francisco J. Ribadas[2]

[1] Departamento de Computación, Universidade da Coruña
Campus de Elviña s/n, 15071 La Coruña, Spain
{jvilares,alonso}@udc.es
[2] Escuela Superior de Ingeniería Informática, Universidade de Vigo
Campus de As Lagoas, 32004 Orense, Spain
ribadas@ei.uvigo.es
http://www.grupocole.org/

Abstract. In this our second participation in the CLEF Spanish monolingual track, we have continued applying Natural Language Processing techniques for single word and multi-word term conflation. Two different conflation approaches have been tested. The first approach is based on the lemmatization of the text in order to avoid inflectional variation. Our second approach consists of the employment of syntactic dependencies as complex index terms, in an attempt to solve the problems derived from syntactic variation and, in this way, to obtain more precise terms. Such dependencies are obtained through a shallow parser based on cascades of finite-state transducers.

1 Introduction

In Information Retrieval (IR) systems, the correct representation of a document through an accurate set of index terms is the basis for obtaining a good performance. If we are not able to both extract and weight appropriately the terms which capture the semantics of the text, this shortcoming will have an effect on all the subsequent processing.

In this context, one of the major limitations we have to deal with is the linguistic variation of natural languages [5], particularly when processing documents written in languages with more complex morphologic and syntactic structures than those present in English, as in the case of Spanish. When managing this type of phenomena, the employment of Natural Language Processing (NLP) techniques becomes feasible. This has been our working hypothesis since our research group, COLE Group, started its work on Spanish Information Retrieval.

As in our first participation in CLEF [23], our main premise is the search for simplicity, motivated by the lack of available linguistic resources for Spanish such as large tagged corpora, treebanks or advanced lexicons. This work is a continuation and refinement of the previous work presented in CLEF 2002, but

centered this time on the employment of lemmatization for solving the *inflectional variation* and the employment of syntactic dependencies for solving the *syntactic variation*.

This article is outlined as follows. Section 2 describes the techniques used for single word term conflation. Section 3 introduces our approach for dealing with syntactic variation through shallow parsing. Official runs are presented and discussed in Section 4. Next, Section 5 describes the set of experiments performed after our attendance at the workshop, in an attempt to eliminate some of the drawbacks detected. Finally, our conclusions and future developments are presented in Section 6.

2 Single Word Term Conflation

Our proposal for single word term conflation continues to be based on exploiting the lexical level in two phases: firstly, by solving the *inflectional variation* through lemmatization, and secondly, by solving the *derivational morphology* through the employment of morphological families.

The process followed for single word term conflation starts by tagging the document. The first step consists of applying our linguistically-motivated preprocessor module [12, 6] 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. Classical approaches, such as stemming, rarely manage these phenomena, resulting in erroneous simplifications during the conflation process.

The output generated by our preprocessor is then taken as input by our tagger-lemmatizer, `MrTagoo` [9], although any high-performance part-of-speech tagger 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 [7], 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 [11]—, management of unknown words, the possibility of integrating external dictionaries in the probabilistic frame defined by the HMM [13], and the possibility of managing ambiguous segmentations [10]

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, 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.

Once text has been tagged, the lemmas of the content words (nouns, verbs and adjectives) are extracted to be indexed. In this way we are solving the problems derived from inflection in Spanish. With regard to computational cost,

the running cost of a lemmatizer-disambiguator is linear in relation to the length of the word, and cubic in relation to the size of the tagset, which is a constant. As we only need to know the grammatical category of the word, the tagset is small and therefore the increase in cost with respect to classical approaches (stemmers) becomes negligible.

Our previous experiments in CLEF 2002 showed that lemmatization performs better than stemming, even when using stemmers which also deal with derivational morphology.

Once inflectional variation has been solved, the next logical step consists of solving the problems caused by derivational morphology. For this purpose, we have grouped the words derivable from each other by means of mechanisms of derivational morphology; each one of these groups is a *morphological family*. Each one of the lemmas belonging to the same morphological family is conflated into the same term, a *representative* of the family. The set of morphological families are automatically generated from a large lexicon of Spanish words by means of a tool which implements the most common derivational mechanisms of Spanish [25]. Since the set of morphological families is generated statically, there is no increment in the running cost.

Nevertheless, our previous experiments in CLEF 2002 showed that the employment of morphological families for single word term conflation introduced too much noise in the system. Thus, we have chosen lemmatization as the conflation technique to be used with single word terms, while morphological families will only be used as a complement in multi-word term conflation, as shown in Section 3.

3 Managing the Syntactic Variation through Shallow Parsing

Following the same scheme of our previous experiments, once we have established the way to process the content of the document at word level, the next step consists of deciding how to process, at phrase level, its syntactic content in order to manage the *syntactic variation* of the document. For this purpose, we will extract the pairs of words related through syntactic dependencies in order to use them as complex index terms. This process is performed in two steps: firstly, the text is parsed by means of a *shallow parser* and, secondly, the syntactic dependencies are extracted and conflated into index terms.

3.1 The Shallow Parser

When dealing with syntactic variation, we have to face the problems derived from the high computational cost of parsing. In order to maintain a linear complexity with respect to the length of the text to be analyzed, we have discarded the employment of full parsing techniques [16], opting for applying *shallow parsing* techniques, also looking for greater robustness.

The theoretical basis for the design of our parser comes from formal language theory, which tells us that, 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. It can be argued that the parsing capability of the system is, in this way, limited by the height of the parseable trees. Nevertheless, this kind of shallow parsing [4] has shown itself to be useful in several NLP application fields, particularly in Information Extraction. Its application in IR, which has not been deeply studied, has been tested by Xerox for English [14], showing its superiority with respect to classical approaches based on contiguous words.

This way, we have implemented a shallow parser based on a five layer architecture whose input is the output of our tagger-lemmatizer. Next, we will describe the function of each layer:

Layer 0: improving the preprocessing. Its function is the management of certain linguistic constructions in order to minimize the noise generated during the subsequent parsing. Such constructions include:

- *Numerals in non-numerical format.*
- *Quantity expressions.* Expressions of the type *algo más de dos millones* (a little more than two million) or *unas dos docenas* (about two dozens), which denote a number but with a certain vagueness about its concrete value, are identified as numeral phrases ($NumP$).
- *Expressions with a verbal function.* Some verbal expressions such as *tener en cuenta* (to take into account), must be considered as a unit, in this case synonym of the verb *considerar* (to consider), to avoid errors in the upper layers such as identifying *en cuenta* as a complement of the verb.

Layer 1: adverbial phrases and first level verbal groups. In this layer the system identifies, on the one hand, the *adverbial phrases* ($AdvP$) of the text, either those with an adverbial head —e.g., *rápidamente* (quickly)—, or those expressions which are not properly adverbial but having an equivalent function —e.g., *de forma rápida* (in a quick way)—. On the other hand, non-periphrastic verbal groups, which we name *first level verbal groups*, are processed, both their simple and compound forms, and both their active and passive forms.

Layer 2: adjectival phrases and second level verbal groups. Adjectival phrases ($AdjP$) such as *azul* (blue) or *muy alto* (very high) are managed here, together with periphrastic verbal groups, such as *tengo que ir* (I have to go), which we name *second level verbal groups*. *Verbal periphrases* are unions of two or more verbal forms working as a unit, giving attributing shades of meaning, such as obligation, degree of development of the action, etc., to the semantics

of the main verb. Moreover, these shades can not be expressed by means of the simple and compound forms of the verb.

Layer 3: noun phrases. In the case of noun phrases (NP), together with simple structures such as the attachment of determiners and adjectives to the noun, we have considered more complex phenomena, such as the existence of *partitive complements* (PC) —e.g., *alguno de* (some of), *ninguno de* (none of)—, in order to cover more complex nominal structures —e.g., *cualquiera de aquellos coches nuevos* (any of those new cars)—.

Layer 4: prepositional phrases. Formed by a noun phrase (NP) preceded by a preposition (P), we have considered three different types according to this preposition, in order to make the extraction of dependencies easier: those preceded by the preposition *por* (by) or $PPby$, those preceded by *de* (of) or $PPof$, and the rest of prepositional phrases or PP.

Each of the rules involved in the different stages of the parsing process has been implemented through a finite-state transducer, compounding, in this way, a parser based on a cascade of finite-state transducers. Therefore, our approach maintains a linear complexity.

3.2 Extraction and Conflation of Dependencies

Once the text has been parsed, the system identifies the syntactic roles of the phrases recognized and extracts the *dependency 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. From a semantical point of view, copulative verbs are mere links, so the dependency is directly established between the subject and the attribute.
- An active verb and the head of its direct 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 such dependencies have been identified, they are conflated through the following conflation scheme:

1. The simple terms compounding the pair are conflated employing morphological families —see Section 2— in order to improve the management of the syntactic variation by covering the appearance of morphosyntactic variants of the original term [24, 15]. In this way, terms such as *cambio en el clima* (change of the climate) and *cambio climático* (climatic change), which express the same concept in different words —but semantically and derivatively related—, can be matched.

2. Conversion to lowercase and elimination of diacritical marks, as in the case of stemmers. Previous experiments show that this process eliminates much of the noise introduced by spelling errors [23].

The process of shallow parsing and extraction of dependencies is explained in detail in [21].

4 CLEF 2003 Official Runs

In this new edition of CLEF, the document corpus for the Spanish Monolingual Track has been enlarged with respect to previous editions. The new corpus is formed by the 215,738 news items (509 MB) from 1994 plus 238,307 more news items (577 MB) from 1995; that is, 454,045 documents (1086 MB). The set of topics consists of 60 queries (141 to 200).

Our group submitted four runs to the CLEF 2003 Spanish monolingual track:

- `coleTDlemZP03` (`TDlemZP` for short): Conflation of content words via lemmatization, that is, each form of a content word is replaced by its lemma. This kind of conflation takes only into account inflectional morphology. The resulting conflated document was indexed using the probabilistic engine ZPrise [3], employing the Okapi BM25 weight scheme [17] with the constants defined in [19] for Spanish ($b = 0.5$, $k_1 = 2$). The query is formed by the set of meaning lemmas present in the *title* and *description* fields —i.e., short topics.
- `coleTDNlemZP03` (`TDNlemZP` for short): The same as before, but the query also includes the set of meaning lemmas obtained from the *narrative* field —i.e., long topics.
- `coleTDNlemSM03` (`TDNlemSM` for short): As in the case of `TDNlemZP`, the three fields of the query are conflated through lemmatization. Nevertheless, this time the indexing engine is the vector-based SMART [8], with an `atn-ntc` weighting scheme [20]. This run was submitted in order to use it as baseline for the rest of runs employing long topics.
- `coleTDNpdsSM03` (`TDNpdsSM` for short): Text conflated via the combination of simple terms, obtained through the lemmatization of content words, and complex terms, obtained through the conflation of syntactic dependencies, as was described in Section 3. According to the results of a previous tuning phase described in [22], the balance factor between the weights of simple and complex terms was fixed at 4 to 1 —i.e., the weights of simple terms are quadrupled— with the aim of increasing the precision of the top ranked documents.

There are no experiments indexing syntactic dependencies with the Okapi BM25 weight scheme, since we are still studying the best way to integrate them into a probabilistic model. With respect to the conditions employed in the official runs, they were:

1. The stopword list was obtained by lemmatizing the content words of the Spanish stopword list provided with SMART [1].

Table 1. CLEF 2003: official results

	TDlemZP	*TDNlemZP*	*TDNlemSM*	*TDNpdsSM*
Documents	57k	57k	57k	57k
Relevant (2368 expected)	2237	2253	2221	2249
R-precision	.4503	.4935	.4453	.4684
Non-interpolated precision	.4662	.5225	.4684	.4698
Document precision	.5497	.5829	.5438	.5408
Precision at 0.00 Re.	.8014	.8614	.7790	.7897
Precision at 0.10 Re.	.7063	.7905	.6982	.7165
Precision at 0.20 Re.	.6553	.7301	.6331	.6570
Precision at 0.30 Re.	.5969	.6449	.5738	.6044
Precision at 0.40 Re.	.5485	.5911	.5388	.5562
Precision at 0.50 Re.	.4969	.5616	.5003	.5092
Precision at 0.60 Re.	.4544	.4871	.4457	.4391
Precision at 0.70 Re.	.3781	.4195	.3987	.3780
Precision at 0.80 Re.	.3083	.3609	.3352	.3191
Precision at 0.90 Re.	.2093	.2594	.2292	.2248
Precision at 1.00 Re.	.1111	.1512	.1472	.1525
Precision at 5 docs.	.5930	.6421	.5930	.5684
Precision at 10 docs.	.5070	.5596	.5018	.4965
Precision at 15 docs.	.4713	.4971	.4515	.4573
Precision at 20 docs.	.4307	.4614	.4281	.4202
Precision at 30 docs.	.3719	.4012	.3784	.3678
Precision at 100 docs.	.2316	.2393	.2316	.2305
Precision at 200 docs.	.1461	.1505	.1455	.1458
Precision at 500 docs.	.0726	.0731	.0718	.0719
Precision at 1000 docs.	.0392	.0395	.0390	.0395

2. Employment of the uppercase-to-lowercase module to process uppercase sentences during tagging.
3. Elimination of spelling signs and conversion to lowercase after conflation in order to reduce typographical errors.
4. Except for the first run, `TDlemZP`, the terms extracted from the *title* field of the query are given double relevance with respect to *description* and *narrative*, since it summarizes the basic semantics of the query.

According to Table 1, the probabilistic-based approach through a BM25 weighting scheme —`TDlemZP` and `TDNlemZP`— proves to be clearly superior to the vector-based `atn-ntc` weighting scheme —`TDNlemSM` and `TDNpdsSM`—, even when only lemmatizing the text. As we can see, `TDlemZP` obtains similar or better results than `TDNlemSM` even when the latter also employs the extra information provided by the *narrative*.

With respect to the main contribution of this work, the use of syntactic dependencies as complex index terms, the results differ slightly from those obtained

Table 2. Distribution of terms in CLEF 2003 collection

	[1..1]	[2..2]	[3..4]	[5..8]	[9..16]	[17..32]	[33..64]	[65..128]	[129..∞)
Lemmas	51.68	13.54	10.16	7.29	5.05	3.50	2.56	1.86	4.36
Dependencies	57.48	14.89	10.61	7.02	4.40	2.65	1.50	0.80	1.04

during the tuning phase [22], where syntactic dependencies clearly showed an improvement in the precision of the top ranked documents. With respect to global performance measures, the `TDNpdsSM` run obtains better results than `TDNlemSM`, except for average document precision. However, the behavior of the system with respect to ranking is not good, since the results obtained for precision at N documents retrieved when employing complex terms —`TDNpdsSM`— are worse than those obtained using only simple lemmatized terms —`TDNlemSM`—. On the other hand, the results for precision vs. recall continue to be better.

5 New Experiments with CLEF 2003 Topics

5.1 Re-tuning the Weight Balance Factor

Taking into account the possibility that the weight balance factor between lemmas and dependencies could be more collection-dependent than supposed, we decided to try different values in a range of 1 to 12. Preliminary experiments [22] showed that a balance factor of 10 could be more appropriate for this larger collection.

Nevertheless, in order to minimize the noise introduced by rare or misspelled terms, and also to reduce the size of the index, we decided to eliminate the most infrequent terms according to their *document frequency* (df) in the collection. Table 2 shows the percentage of different terms (lemmas or dependencies) which appear in only 1 document, in 2 documents, between 3 and 4 documents, between 5 and 8 documents, and so on. As we can observe, for example, 52% of lemmas and 57% of dependencies only appear in one document of the collection. Taking into account these statistics, we decided to discard those terms which appear in less than five documents. This pruning of the index allowed us to eliminate 75% of the lemmas and 82% of the dependency pairs with minimal incidence in the performance of the system.

Table 3 shows the new results obtained. The first column, *lem*, shows the results for our baseline, lemmatization —as in the official run `TDNlemSM`—, whereas the next columns, *sdx*, contain the results obtained by merging lemmatized simple terms and complex terms based on syntactic dependencies (*sd*) when the weight balance factor between simple and complex terms, x to 1, changes —i.e., when the weight of simple terms is multiplied by x. As already stated in Section 4, *sd4* shows the results obtained with $x = 4$, the balance factor used in the official run `TDNpdsSM`. The column *opt* shows the best results obtained for *sdx*,

Table 3. CLEF 2003: Re-tuning the system a posteriori

	lem	*sd1*	*sd2*	*sd4*	*sd6*	*sd8*	*sd10*	*sd12*	*opt*	*Δ*
Documents	57k	57k	57k	57k	57k	57k	57k	57k	- -	- -
Relevant (2368 expected)	2221	2218	2241	**2248**	2243	2244	2242	2239	2248	27
Non-interpolated precision	.4681	.4014	.4413	.4613	.4656	.4696	**.4710**	.4705	.4710	.0029
Document precision	.5431	.4647	.5149	.5394	.5444	.5470	**.5475**	.5472	.5475	.0044
R-precision	.4471	.3961	.4415	**.4542**	.4480	.4463	.4454	.4450	.4542	.0071
Precision at 0.00 Re.	.7805	.7562	**.7940**	.7721	.7827	.7811	.7776	.7856	.7940	.0135
Precision at 0.10 Re.	.6994	.6428	.6817	.7036	.7032	**.7125**	.7109	.7104	.7125	.0131
Precision at 0.20 Re.	.6343	.5640	.6097	.6392	.6501	**.6526**	.6464	.6421	.6526	.0183
Precision at 0.30 Re.	.5736	.5144	.5614	**.5925**	.5913	.5922	.5912	.5867	.5925	.0189
Precision at 0.40 Re.	.5332	.4706	.5108	.5348	.5347	.5307	.5336	**.5357**	.5357	.0025
Precision at 0.50 Re.	.4987	.4222	.4647	.4931	.4936	.4975	**.5040**	.5032	.5040	.0053
Precision at 0.60 Re.	.4462	.3643	.4183	.4380	.4382	.4444	.4467	**.4468**	.4462	.0006
Precision at 0.70 Re.	.3969	.3122	.3466	.3758	.3884	.3941	**.3969**	.3958	.3969	.0000
Precision at 0.80 Re.	.3343	.2578	.2964	.3186	.3274	.3291	.3296	**.3299**	.3299	-.0044
Precision at 0.90 Re.	.2294	.1951	.2156	.2245	.2306	**.2333**	.2313	.2316	.2333	.0039
Precision at 1.00 Re.	.1470	.1316	.1499	**.1514**	.1493	.1501	.1488	.1489	.1514	.0044
Precision at 5 docs.	.5965	.4842	.5228	.5684	.5825	.6000	**.6070**	.5930	.6070	.0105
Precision at 10 docs.	.5000	.4333	.4825	.4947	.5018	.5035	**.5053**	**.5053**	.5053	.0053
Precision at 15 docs.	.4515	.3860	.4409	**.4561**	.4503	.4503	.4503	.4526	.4561	.0046
Precision at 20 docs.	.4281	.3632	.4009	.4193	.4184	.4211	**.4237**	**.4237**	.4237	-.0044
Precision at 30 docs.	.3813	.3205	.3497	.3673	.3760	.3772	**.3789**	**.3789**	.3789	-.0024
Precision at 100 docs.	.2314	.2053	.2221	.2309	.2332	**.2340**	.2337	.2333	.2340	.0026
Precision at 200 docs.	.1455	.1368	.1429	.1454	.1464	**.1465**	.1461	.1461	.1465	.0010
Precision at 500 docs.	.0718	.0692	.0711	.0718	.0719	.0719	**.0720**	.0719	.0720	.0002
Precision at 1000 docs.	.0390	.0389	.0393	.0394	**.0394**	.0394	.0393	.0393	.0394	.0004

written in boldface, whereas the column *Δ* shows the improvement of *opt* with respect to *lem*.

These new results corroborate those obtained in previous experiments [22], since *sd10* continues to be the best choice for our purpose, which is to increase the precision of the top ranked documents. It obtains the best results for precision at N documents, and non-interpolated and document precision, being slightly better than those obtained through lemmatization (*lem*). In the case of precision vs. recall, *sd4* is the best compromise in the range 0.00–0.40.

5.2 Incorporating Pseudo-relevance Feedback

A second set of experiments consisted of the application of pseudo-relevance feedback (blind-query expansion) adopting Rocchio's approach [18] in the case of lemmas indexed with SMART:

$$Q_1 = \alpha Q_0 + \beta \sum_{k=1}^{n_1} \frac{R_k}{n_1} - \gamma \sum_{k=1}^{n_2} \frac{S_k}{n_2}$$

where Q_1 is the new query vector, Q_0 is the vector for the initial query, R_k is the vector for relevant document k, S_k is the vector for non-relevant document k, n_1 is the number of relevant documents, n_2 is the number of non-relevant documents, and α, β and γ are, respectively, the parameters that control the relative

Table 4. Tuning the parameters for blind-query expansion ($\beta = 0.10$ fixed)

	Non-interpolated precision									
α	1.00		1.20		1.40		1.60		1.80	
no. of docs.	5	10	5	10	5	10	5	10	5	10
5 terms	.5175	.4986	.5176	.4988	.5181	.4987	.5180	.4986	.5172	.4984
10 terms	.5201	.5027	.5210	.5035	.5211	.5039	.5211	.5041	.5204	.5038
15 terms	.5205	.5055	.5218	.5064	.5225	.5070	.5226	.5071	.5227	.5069
20 terms	.5223	.5074	.5234	.5087	.5243	.5091	.5244	.5093	.5252	.5093

contributions of the original query, relevant documents, and non-relevant documents. In the case of our system, it only takes into account relevant documents ($\gamma = 0$).

Firstly, we explored the space of solutions searching for, on the one hand, the best relation α/β, and on the other hand, for the most accurate number of documents and terms to be used in the expansion. Table 4 shows the non-interpolated precision when different values of α and different numbers of terms and documents are used. Preliminary experiments, not presented here so as not to tire the reader, had shown that the behaviour of our system improved when increasing the value of α; these experiments were made using a fixed value of $\beta = 0.10$ while varying α. The best results were obtained with a value of α in the range 1.40–1.80, finally opting for expanding the query with the best 10 terms of the 5 top ranked documents using $\alpha = 1.40$ and $\beta = 0.10$.

Table 5 contains the final set of results obtained for CLEF 2003 topics. The runs are the same as those submitted to the official track, except for the following changes:

1. Those terms which appear in less than five documents have been discarded.
2. The balance factor for the run `TDNpdsSM` has been increased to 10 —i.e., the weight of lemmas is multiplied by 10.
3. A new run has been considered, `TDNlemSM-f`. This run is the same as `TDNlemSM`, but applying Rocchio's approach for pseudo-relevance feedback. The initial query is expanded with the best 10 terms of the 5 top ranked documents using $\alpha = 1.40$ and $\beta = 0.10$.

The results and their interpretation are similar to those obtained in the official runs. Nevertheless, the use of a bigger balance factor in `TDNpdsSM` now leads to a slight improvement with respect to the baseline, `TDNlemSM`. On the other hand, as was expected, the employment of relevance feedback in `TDNlemSM-f` produces major improvement. We expect that the application of pseudo-relevance feedback to `TDNpdsSM` will produce a similar increase in performance. Currently, we are are investigating how to adapt Rocchio's approach to this case.

Table 5. CLEF 2003: final results

	TDlemZP	*TDNlemZP*	*TDNlemSM*	*TDNpdsSM*	*TDNlemSM-f*
Documents	57k	57k	57k	57k	57k
Relevant (2368 expected)	2235	2253	2221	2242	2260
Non-interpolated precision	.4619	.5163	.4681	.4710	.5211
Document precision	.5478	.5818	.5431	.5475	.6086
R-precision	.4480	.4928	.4471	.4454	.4796
Precision at 0.00 Re.	.7894	.8429	.7805	.7776	.7760
Precision at 0.10 Re.	.7027	.7717	.6994	.7109	.7134
Precision at 0.20 Re.	.6447	.7161	.6343	.6464	.6636
Precision at 0.30 Re.	.5932	.6377	.5736	.5912	.6204
Precision at 0.40 Re.	.5401	.5895	.5332	.5336	.5925
Precision at 0.50 Re.	.4905	.5544	.4987	.5040	.5467
Precision at 0.60 Re.	.4544	.4844	.4462	.4467	.4932
Precision at 0.70 Re.	.3758	.4189	.3969	.3969	.4655
Precision at 0.80 Re.	.3042	.3570	.3343	.3296	.4095
Precision at 0.90 Re.	.2076	.2586	.2294	.2313	.3364
Precision at 1.00 Re.	.1145	.1539	.1470	.1488	.2306
Precision at 5 docs.	.5860	.6281	.5965	.6070	.6000
Precision at 10 docs.	.5053	.5561	.5000	.5053	.5421
Precision at 15 docs.	.4632	.4971	.4515	.4503	.4982
Precision at 20 docs.	.4272	.4605	.4281	.4237	.4640
Precision at 30 docs.	.3737	.4035	.3813	.3789	.4105
Precision at 100 docs.	.2321	.2404	.2314	.2337	.2461
Precision at 200 docs.	.1463	.1504	.1455	.1461	.1527
Precision at 500 docs.	.0726	.0731	.0718	.0720	.0742
Precision at 1000 docs.	.0392	.0395	.0390	.0393	.0396

6 Conclusions and Future Work

Throughout this article we have studied the employment of Natural Language Processing techniques for managing linguistic variation in Spanish Information Retrieval. At word-level, inflectional variation has been solved through lemmatization whereas, at phrase-level, syntactic variation has been managed through the employment of syntactic dependencies as complex index terms. Such dependencies were obtained through a shallow parser based on cascades of finite-state transducers, and then conflated by means of derivational morphology.

The improvement obtained using syntactic information is not as great as expected, which suggests that our actual way of integrating such information must be improved. Our future work focuses on this goal in three different ways: firstly, its integration in a probabilistic retrieval model (e.g., using the Okapi BM25 weight scheme); secondly, testing its behaviour during feedback; thirdly, the possibility of storing simple and complex terms in separate indexes, combining them afterwards by means of *data fusion* techniques [26].

Acknowledgements

The research described in this paper has been supported in part by Ministerio de Ciencia y Tecnología (TIC2000-0370-C02-01, HP2001-0044 and HF2002-81), FPU grants of Secretaría de Estado de Educación y Universidades, Xunta de Galicia (PGIDIT03SIN30501PR, PGIDT01PXI10506PN, PGIDIT02PXIB30501PR and PGIDIT02SIN01E), Universidade de Vigo, and Universidade da Coruña. The authors also would like to thank Darrin Dimmick, from NIST, for giving us the opportunity to use the ZPrise system, and Fernando Martínez, from Universidad de Jaén, for helping us to make it operative in our system.

References

1. `ftp://ftp.cs.cornell.edu/pub/smart` (site visited October 2003).
2. `http://www.clef-campaign.org` (site visited October 2003).
3. `http://www.itl.nist.gov` (site visited October 2003).
4. S. Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1997.
5. A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. Linguistically motivated information retrieval. In *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., New York and Basel, 2000.
6. Fco. Mario Barcala, Jesús Vilares, Miguel A. Alonso, Jorge Graña, and Manuel Vilares. Tokenization and proper noun recognition for information retrieval. In A Min Tjoa and Roland R. Wagner, editors, *Thirteen International Workshop on Database and Expert Systems Applications. 2-6 September 2002. Aix-en-Provence, France*, pages 246–250, Los Alamitos, California, USA, September 2002. IEEE Computer Society Press.
7. Thorsten Brants. TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP'2000)*, Seattle, 2000.
8. C. Buckley. Implementation of the SMART information retrieval system. Technical report, Department of Computer Science, Cornell University, 1985. Source code available at [1].
9. Jorge Graña. *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*. PhD thesis, University of La Coruña, La Coruña, Spain, 2000.
10. Jorge Graña, Miguel A. Alonso, and Manuel Vilares. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. In Petr Sojka, Ivan Kopecek, and Karel Pala, editors, *Text, Speech and Dialogue*, volume 2448 of *Lecture Notes in Computer Science*, pages 3–10. Springer-Verlag, Berlin-Heidelberg-New York, 2002.
11. 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, *Implementation and Application of Automata*, volume 2494 of *Lecture Notes in Computer Science*, pages 135–148. Springer-Verlag, Berlin-Heidelberg-New York, 2002.
12. 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.

13. Jorge Graña, Jean-Cédric Chappelier, and Manuel Vilares. Integrating external dictionaries into stochastic part-of-speech taggers. In *Proceedings of the Euroconference Recent Advances in Natural Language Processing (RANLP 2001)*, pages 122–128, Tzigov Chark, Bulgaria, 2001.
14. 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 *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, pages 167–180, 1997.
15. 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.
16. Jose Perez-Carballo and Tomek Strzalkowski. Natural language information retrieval: progress report. *Information Processing and Management*, 36(1):155–178, 2000.
17. S.E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In E. Voorhees and D. K. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, NIST Special Publication 500-264, pages 151–161, 2000.
18. J.J. Rocchio. Relevance Feedback in Informatio Retrieval. In G. Salton, editor, *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.
19. J. Savoy. Report on CLEF-2002 Experiments: Combining Multiple Sources of Evidence. In C. Peters, editor, *Results of the CLEF 2002 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2002 Workshop*, pages 31–46, Rome, Italy, Sept. 2002. Available at [2].
20. J. Savoy, A. Le Calve, and D. Vrajitoru. Report on the TREC-5 experiment: Data fusion and collection fusion. Proceedings of TREC'5, NIST publication #500-238, pages 489–502, Gaithersburg, MD, 1997.
21. Jesús Vilares, and Miguel A. Alonso. A Grammatical Approach to the Extraction of Index Terms. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov and Ni. Nikolov, editors, *Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, pages 500-504, Borovets, Bulgaria, 2003.
22. Jesús Vilares, Miguel A. Alonso and Francisco J. Ribadas. COLE experiments at CLEF 2003 Spanish monolingual track. In C. Peters and F. Borri, editors, *Results of the CLEF 2003 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2003 Workshop*, pages 197–206, Trondheim, Norway, August 2003. Available at [2].
23. Jesús Vilares, Miguel A. Alonso, Francisco J. Ribadas, and Manuel Vilares. COLE experiments at CLEF 2002 Spanish monolingual track. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Advances in Cross-Language Information Retrieval: Results of the CLEF 2002 Evaluation Campaign*, volume 2785 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin-Heidelberg-New York, 2003.
24. Jesús Vilares, Fco. Mario Barcala, and Miguel A. Alonso. Using syntactic dependency-pairs conflation to improve retrieval performance in Spanish. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 381–390. Springer-Verlag, Berlin-Heidelberg-New York, 2002.
25. Jesús Vilares, David Cabrero, and Miguel A. Alonso. Applying productive derivational morphology to term indexing of Spanish texts. In Alexander Gelbukh, editor,

Computational Linguistics and Intelligent Text Processing, volume 2004 of *Lecture Notes in Computer Science*, pages 336–348. Springer-Verlag, Berlin-Heidelberg-New York, 2001.
26. C. C. Vogt and G. W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.