COLE experiments at CLEF 2002 Spanish monolingual track

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

[1] Departamento de Computación, Universidade da Coruña
Campus de Elviña s/n, 15071 La Coruña, Spain
jvilares@mail2.udc.es alonso@udc.es
[2] Escuela Superior de Ingeniería Informática, Universidade de Vigo
Campus de As Lagoas, 32004 Orense, Spain
ribadas@ei.uvigo.es vilares@uvigo.es
http://www.grupocole.org/

Abstract. In this our first participation in CLEF, we have applied Natural Language Processing techniques for single word and multi-word term conflation. We have tested several approaches at different levels of text processing in our experiments: firstly, we have lemmatized the text to avoid inflectional variation; secondly, we have expanded the queries through synonyms according to a fixed threshold of similarity; thirdly, we have employed morphological families to deal with derivational variation; and fourthly, we have tested a mixed approach based on the employment of such families and syntactic dependencies to deal with the syntactic content of the document.

1 Introduction

In Text Retrieval, since the information is encoded as text, the task of deciding whether a document is relevant or not to a given information need can be viewed as a Natural Language Processing (NLP) problem, in particular for languages with rich lexical, morphological and syntactical structures, such as Spanish. Moreover, during recent years the progress in the field of NLP has resulted in the development of a new generation of more efficient, robust and precise tools. These advances, together with the increasing power of new computers, allow us to apply such NLP systems in real IR environments.

Nevertheless, at this point, we must face one of the main problems of NLP in Spanish, the lack of available linguistic resources: large tagged corpora, treebanks and advanced lexicons are not available. Therefore, while waiting for the availability of such resources, the only solution is to look for simplicity, employing a minimum of linguistic resources.

In this paper we present a set of NLP tools designed for dealing with different levels of linguistic variation in Spanish: morphological, lexical and syntactical.

The effectiveness of our solutions has been tested during this our first participation in the CLEF Spanish monolingual track.

This article is outlined as follows. Section 2 describes the techniques used for single word term conflation. Expansion of queries by means of synonyms is introduced in Sect. 3. Multi-word term conflation through syntactic dependencies is described in Sect. 4. Section 5 describes our module for recovering of uppercase phrases. In Sect. 6, the results of our experiments using the CLEF Spanish corpus are shown. Finally, in Sect. 7 we explain our conclusions and future work.

2 Conflation of words using inflectional and derivational morphology

Our proposal for single word term conflation is based on exploiting the lexical level in two phases: firstly, by lemmatizing the text to solve inflectional variation, and secondly, by employing morphological families to deal with derivational morphology.

In this process, the first step consists of tagging the document. Document processing starts by applying our linguistically-motivated preprocessor module [9, 3], performing 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. It is interesting to remark that classical techniques do not deal with many of these phenomena, resulting in wrong simplifications during conflation process.

The output of the preprocessor is taken as input by the tagger-lemmatizer. Although any kind of tagger could be applied, in our system we have used a second order Markov model for part-of-speech tagging. The elements of the model and the procedures to estimate its parameters are based on Brant's work [4], incorporating information from external dictionaries [10] which are implemented by means of numbered minimal acyclic finite-state automata [8].

Once text has been tagged, the lemmas of the content words (nouns, verbs, adjectives) are extracted to be indexed. In this way we are solving the problems derived from inflection in Spanish and, as a result, recall is increased. 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.

Now inflectional variation has been solved, the next logical step is to solve the problems caused by derivational morphology. Spanish has a great productivity and flexibility in its word formation mechanisms by using a rich and complex productive morphology, preferring derivation to other mechanisms of word formation. We have considered the derivational morphemes, the allomorphic variants of such morphemes and the phonological conditions they must satisfy, to automatically generate the set of morphological families from a large lexicon of

Spanish words [15]. The resulting morphological families can be used as a kind of advanced and linguistically motivated stemmer for Spanish, where every lemma is substituted by a fixed representative of its morphological family. Since the set of morphological families is generated statically, there is no increment in the running cost.

3 Using synonymy to expand queries

The use of synonymy relations in the task of automatic query expansion is not a new subject, but the approaches presented until now do not assign a weight to the degree of synonymy that exists between the original terms present in the query and those produced by the process of expansion [11]. Nevertheless, our system does have access to this information, so a threshold of synonymy can be set in order to control the degree of query expansion.

The most frequent definition of synonymy conceives it as a relation between two expressions with identical or similar meaning. The controversy of understanding synonymy as a precise question or as an approximate question, i.e. as a question of identity or as a question of similarity, has existed from the beginning of the study of this semantic relation. In our system, synonymy is understood as a gradual relation between words. In order to calculate the degree of synonymy, we use the *Jaccard's coefficient* as measure of similarity applied on the sets of synonyms provided by a dictionary of synonyms for each of its entries [6]. Given two sets X and Y, their *similarity* is measured as:

$$sm(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Let us consider a word w with m_i possible meanings, and another word w' with m_j possible meanings, where $dc(w, m_i)$ represents the function that gives us the set of synonyms provided by the dictionary for every entry w in the concrete meaning m_i. The degree of synonymy of w and w' in the meaning m_i of w is calculated as $dg(w, m_i, w') = \max_j \; sm[dc(w, m_i), dc(w', m_j)]$. Furthermore, by calculating $k = \arg\max_j \; sm[dc(w, m_i), dc(w', m_j)]$ we obtain in m_k the meaning of w' closest to the meaning m_i of w.

4 Extracting dependencies between words by means of a shallow parser

Our system is not only able to process the content of the document at word level, it can also process its syntactic structure. For this purpose, a parser module obtains from the tagged document the *head-modifier* pairs corresponding to the most relevant syntactic dependencies: *noun-modifier*, relating the head of a noun phrase with the head of a modifier; *subject-verb*, relating the head of the subject with the main verb of the clause; and *verb-complement*, relating the main verb of the clause with the head of a complement.

The kernel of the grammar used by this shallow parser is inferred from the basic trees corresponding to noun phrases[1] and their syntactic and morpho-syntactic variants [12, 14]:

- *Syntactic variants* result from the inflection of individual words and from modifying the syntactic structure of the original noun phrase by means of:
  - *Synapsy:* it corresponds to a change of preposition or the addition or removal of a determiner, e.g. *una caída de ventas* (a drop in sales).
  - *Substitution:* it consists of employing modifiers to make a term more specific, e.g. *una caída inusual de ventas* (an unusual drop in sales).
  - *Permutation:* this refers to the permutation of words around a pivot element, e.g. *una inusual caída de ventas* (an unusual drop in sales).
  - *Coordination:* this consists of employing coordinating constructions (copulative or disjunctive) with the modifier or with the modified term, e.g. *una inusual caída de ventas y de beneficios* (an unusual drop in sales and profits).
- *Morpho-syntactic variants* differ from syntactic variants in that at least one of the content words of the original noun phrase is transformed into another word derived from the same morphological stem, e.g. *las ventas han caído* (sales have dropped).

We must remark that syntactic variants involve inflectional morphology but not derivational morphology, whereas morpho-syntactic variants involve both inflectional and derivational morphology. In addition, syntactic variants have a very restricted scope (the noun phrase) whereas morpho-syntactic variants can span a whole sentence, including a verb and its complements.

Once the basic trees of noun phrases and their variants have been established, they are compiled into a set of regular expressions, which are matched against the tagged document in order to extract its dependencies in the form of pairs which are used as index terms after conflating their components through morphological families, as is described in [14]. In this way, we are identifying dependency pairs through simple pattern matching over the output of the tagger-lemmatizer, solving the problem by means of finite-state techniques, leading to a considerable reduction of the running cost.

5 The uppercase-to-lowercase module

An important characteristic of IR test collections that may have a considerable impact on the performance of linguistically motivated indexing techniques is the large number of typographical errors present in documents, as has been reported, in the case of the Spanish CLEF corpus, by [7]. In particular, words in news titles and subsection headings are generally written in capital letters without accents, and they can not be correctly managed by the preprocessor and tagger modules,

[1] At this point we will take as example the noun phrase *una caída de las ventas* (a drop in the sales).

thus leading to incorrect conflations. We must take into account that these titles are usually very indicative of the topic of the document.

Trying to solve this problem, we have incorporated an *uppercase-to-lowercase* module to our system to process uppercase sentences, converting them to lowercase and restoring the existent diacritics when necessary. Other approaches, such as [17], deal with documents where absolutely all diacritics have been eliminated. Nevertheless, our situation is different, because the main of the document is written lowercase and preserves their diacritics, only some sentences are written in capital letters; moreover, for our purposes we only need the grammatical category and lemma of the word, not the form.

So, we can employ the lexical context of an uppercase sentence, either forms and lemmas, to recover this lost information. The first step of this process is to identify the uppercase phrases. We consider that a sequence of words form an *uppercase phrase*, when it consists of three or more words written in capital letters and at least three of them have more than three characters. For each of these uppercase phrases we do the following:

1. We obtain its surrounding context.
2. For each of the words in the phrase:
   (a) We examine the context looking for entries with the same flattened form [2]. Each of these words become candidates.
   (b) If candidates are found, the most numerous is chosen, and in case of existing a draw, the closest to the phrase is chosen.
   (c) If no candidates are found, the lexicon is examined:
      i. We obtain from the lexicon all entries with the same flattened form, grouping them according to their category and lemma (we are not interested in the form, just in the category and the lemma of the word).
      ii. If no entries are found, we keep the actual tag and lemma.
      iii. If only one entry is found, we choose that one.
      iv. If more than one entry is found, we choose the most numerous in the context (according to the category and the lemma). Again, in case of existing a draw, we choose the closest to the sentence.

Sometimes, some words of the uppercase phrase preserve some of their diacritics, for example the ˜ of the Ñ. In this situations the candidates from the context or the lexicon must observe this restriction.

6 Experiments with CLEF Spanish corpus

The Spanish CLEF corpus used for these experiments is formed by 215,738 documents corresponding to the news provided by EFE, a Spanish news agency, in

[2] That is, after both words been converted to lowercase, and after eliminating all diacritics from them

1994. Documents are formatted in SGML, with a total size of 509 Megabytes. After deleting SGML tags, the size of the text corpus is reduced to 438 Megabytes. Each query consists of three fields: a brief title statement, a one-sentence description, and a more complex narrative specifying the relevance assessment criteria.

The techniques proposed in this article are independent of the indexing engine we choose to use. This is because we first conflate each document to obtain its index terms; then, the engine receives the conflated version of the document as input. So, any standard text indexing engine may be employed, which is a great advantage. Nevertheless, each engine will behave according to its own characteristics, that is, its indexing model, ranking algorithm, etc. [16]. In our case, we have worked with the vector-based engine SMART.

We have compared the results obtained by five different indexing methods:

- Stemming text after eliminating stopwords (*stm*). In order to apply this technique, we have tested several stemmers for Spanish. Finally, the best results we obtained were for the stemmer used by the open source search engine Muscat[3], based on Porter's algorithm [2]. Additionally, this process eliminates accents from text before converting it to lowercase.
- Conflation of content words via lemmatization (*lem*), i.e. each form of a content word is replaced by its lemma. This kind of conflation takes only into account inflectional morphology.
- Conflation of content words via lemmatization and expansion of queries by means of synonymy (*syn*). We have considered that two words are synonyms if their similarity measure is greater or equal to 0.80. Previous experiments have shown that the expansion of narrative field introduces too much noise in the system; for this reason we only allow title and description fields to be expanded.
- Conflation of content words by means of morphological families (*fam*), i.e. each form of a content word is replaced by the representative of its morphological family. This kind of conflation takes into account both inflectional and derivational morphology.
- Text conflated by means of the combined use of morphological families and syntactic dependency pairs (*f-sdp*).

The methods *lem*, *syn*, *fam*, and *f-sdp* are linguistically motivated. Therefore, they are able to deal with some complex linguistic phenomena such as clitic pronouns, contractions, idioms, and proper name recognition. In contrast, the method *stm* works simply by removing a given set of suffixes, without taking into account such linguistic phenomena, yielding incorrect conflations that introduce noise in the system. For example, clitic pronouns are simply considered a set of suffixes to be removed. Moreover, the employment of finite-state techniques in the implementation of our methods let us to reduce their computational cost, making possible their application in practical environments.

[3] Currently, Muscat is not an open source project, and the web site `http://open.muscat.com` used to download the stemmer is not operating. Information about a similar stemmer for Spanish (and other European languages) can be found at `http://snowball.sourceforge.net/spanish/stemmer.html`.

Table 1. CLEF 2002 (submitted): performance measures

	TD*lem*	TDN*lem*	TDN*syn*	TDN*f-sdp*
Documents retrieved	50,000	50,000	50,000	50,000
Relevant docs retrieved (2854 expected)	2,495	2,634	2,632	2,624
R-precision	0.3697	0.4466	0.4438	0.3983
Average precision per query	0.3608	0.4448	0.4423	0.4043
Average precision per relevant docs	0.3971	0.4665	0.4613	0.4472
11-points average precision	0.3820	0.4630	0.4608	0.4205

6.1 CLEF 2002 original experiments

Original results submitted to CLEF 2002 included four different runnings:

- TD*lem*: Conflation of title + description content words via lemmatization (*lem*).
- TDN*lem*: The same as before, but using title + description + narrative.
- TDN*syn*: Conflation of title + description + narrative via lemmatization and expansion by means of synonymy (*syn*). It must be noticed that only title and description fields were expanded.
- TDN*f-sdp*: Text conflated by means of the combined use of morphological families and syntactic dependency pairs (*f-sdp*), and using title + description + narrative for constructing the queries.

For this set of experiments, the following conditions were applied:

1. Employment of the `lnc-ltc` weighting scheme [5].
2. Stopword list obtained from the content word lemmas of the Spanish stopword list provided by SMART [4].
3. Employment of the uppercase-to-lowercase module for recovering uppercase sentences.
4. Except for TD*lem*, the terms extracted from the title section were given the double of importance with respect to description and narrative.

According to Table 1, all NLP-based methods showed a better behavior than standard stemming, but lemmatization method (TDN*lem*) seemed to be the best option, even when only dealing with inflectional variation. The expansion through synonymy (TDN*syn*) did not improve such results because the expansion was *total*, that is, all synonyms of all terms of the query were employed, and no word sense disambiguation procedures were available; this way, too much noise was introduced in the system. In the case of the employment of syntactic dependency pairs (TDN*f-dsp*), the results did not show any improvement with respect to the other NLP-based techniques considered, except in the case of average precision at N seen documents, where it obtained a better behavior for the 10 first retrieved documents.

[4] `ftp://ftp.cs.cornell.edu/pub/smart/`

6.2 New experiments: tuning the system with CLEF 2001 queries

After our attendance to CLEF 2002 Workshop, we decided to improve our system by applying some extra processing and by using a better weighting scheme, the `atn-ntc` [13]. Nevertheless, before testing our conflation techniques with the new changes, we tuned our system using CLEF 2001 queries. During this training phase we have worked only with *lem* conflation technique because, as it was shown in the original CLEF runnings and other previous experiments [16], it demonstrated to be a good starting point for the NLP techniques we are considering. For these training experiments, all information available from the queries has been employed, using the three fields of each topic: title + description + narrative. Moreover, the same conclusions were obtained for parallel experiments using title + description, as it is required by CLEF for competition purposes.

Table 2 shows the performance measures obtained during this tuning phase with CLEF 2001 topics. The monolingual Spanish task in 2001 considered a set of 50 queries, but for one query any relevant document existed in the corpus, and so the performance measures were computed over 49 queries.

Our initial approach consisted of not applying the uppercase-to-lowercase module, and using a very restricted stopword list formed by the lemmas of the most common verbs in Spanish[5]. The results obtained for this base case are shown in the column *step 1* of Table 2.

Our first improvement consisted of enlarging the stopword list using the list employed in the submitted results, that is, the lemmas of the content words of the Spanish stopword list provided with SMART engine. The results obtained, see column *step 2* of Table 2, are very similar to the previous ones, though there exist a slight improvement when using this longer list and an extra reduction of 6% in the size of the inverted file of the index. Therefore, we decided to continue using the SMART lemmas list.

The next step consisted in introducing our uppercase-to-lowercase module. The results, shown in the column *step 3* of Table 2, seem to demonstrate that the behavior of the system improves when the lemmas of uppercase sentences are recovered. Notice that, at this point, all the conditions considered were also applied to the original CLEF 2002 results.

Nevertheless, there still existed many typographical errors in the body of the documents, many of them consisting in unaccented vowels; part of this problem can be solved by eliminating the accents from the conflated text. The rationale of this solution is that once the lemma of a word has been identified there is no reason for keeping the accents. It can be argued that we will lose the *diacritical accents*[6], but if we are working with content word lemmas such problem disappears. However, we will keep *'ñ'* characters in the texts, i.e. not converting them to *'n'*, because it may introduce more noise in the system by conflating words, e.g. *cana* (grey hair) and *caña* (cane), into the same term. Moreover, in Spanish, it is relatively frequent to forget an accent when writing, but a confusion

[5] i.e. *ser*, *estar*, *haber*, *tener*, *ir* and *hacer*

[6] Accents for distinguishing between words with the same graphical form but different meaning, e.g. *mí* (me) - *mi* (my).

Table 2. CLEF 2001: training process using conflation through lemmatization (*lem*)

	step 1	***step 2***	***step 3***	***step 4***	***step 5***	***step 6***
Documents retrieved	49,000	49,000	49,000	49,000	49,000	49,000
Rel. docs retrieved (2694 exp.)	2,602	2,602	2,607	2,609	2,621	2,623
R-precision	0.5067	0.5115	0.5094	0.5156	0.5250	0.5269
Avg. non-interpolated precision	0.5231	0.5240	0.5312	0.5403	0.5512	0.5535
Avg. document precision	0.6279	0.6272	0.6339	0.6385	0.6477	0.6483
11-points avg. precision	0.5289	0.5301	0.5380	0.5467	0.5571	0.5600
3-points avg. precision	0.5422	0.5444	0.5513	0.5613	0.5727	0.5735

between a *'ñ'* and a *'n'* it is extremely rare. In the column *step 4* of Table 2 we see the improvements reached with this solution.

In a similar way, an additional experiment was made by also converting to lowercase the resulting text as in the case of stemming, and the results obtained showed an extra improvement, as we can see in column *step 5* of Table 2.

Our final case of study consisted, as for original submitted results, in giving double importance to the title statement of the topic with respect to description and narrative, as we suppose it concentrates the main information of the query. The improvement attained with this measure can be seen in column *step 6* of Table 2.

The conditions employed in this last running will be the retained for further experiments:

1. Employment of the `atn-ntc` weighting scheme.
2. Stopword list obtained from the content word lemmas of SMART stopword list.
3. Employment of the uppercase-to-lowercase module for recovering uppercase sentences.
4. Elimination of accents after conflation for minimizing typographical errors.
5. Conversion to lowercase after conflation.
6. Double importance of the title statement.

6.3 New experiments with CLEF 2001 and CLEF 2002 topics

In Table 3 and Table 4 we show the results obtained for 2001 topics by our NLP-based conflation techniques (*lem*, *syn*, *fam*, *f-sdp*) compared with respect to stemming (*stm*) when applying the new conditions.

In contrast with the results obtained in [1] for the same topics using `lnc-ltc` scheme, only *lem* conflation method beats *stm* now. This is due to a modification in the behavior of the system with the new weighting scheme. This new scheme improves the results obtained for all the conflation methods considered with respect to the previous scheme, but much more in the case of stemming and

Table 3. CLEF 2001: performance measures

	stm	***lem***	***syn***	***fam***	***f-sdp***
Documents retrieved	49,000	49,000	49,000	49,000	49,000
Relevant docs retrieved (2694 expected)	2,628	2,623	2,620	2,611	2,575
R-precision	0.5221	0.5269	0.5170	0.5139	0.4839
Average non-interpolated precision	0.5490	0.5535	0.5420	0.5360	0.5046
Average document precision	0.6277	0.6483	0.6326	0.6128	0.5370
11-points average precision	0.5574	0.5600	0.5486	0.5431	0.5187
3-points average precision	0.5691	0.5735	0.5660	0.5552	0.5306

Table 4. CLEF 2001: average precision at 11 standard recall levels

Recall	Precision				
	stm	***lem***	***syn***	***fam***	***f-sdp***
0.00	0.8895	0.8975	0.8693	0.8616	0.8648
0.10	0.7946	0.7951	0.7802	0.7672	0.7603
0.20	0.7393	0.7532	0.7426	0.7212	0.6975
0.30	0.6779	0.6994	0.6779	0.6684	0.6217
0.40	0.6394	0.6526	0.6367	0.6137	0.5712
0.50	0.5867	0.5878	0.5781	0.5559	0.5359
0.60	0.5299	0.5228	0.5145	0.4988	0.4707
0.70	0.4411	0.4412	0.4357	0.4355	0.4029
0.80	0.3814	0.3794	0.3772	0.3886	0.3585
0.90	0.2952	0.2831	0.2766	0.2956	0.2663
1.00	0.1561	0.1477	0.1459	0.1678	0.1563

lemmatization than in the case of the employment of synonymy and morphological families. The reason for that may be due to a higher sensitiveness to the noise introduced by bad constructed families in the case of *fam*, and therefore also in *f-sdp*, and to the noise introduced by our approach for expansion through synonymy in the case of *syn*.

Nevertheless, as we can see in Table 3 and Table 4, *lem* continues beating *stm*, even when being the simpler approach.

The behavior of the system with CLEF 2002 topics, see Table 5 and Table 6, is very similar to 2001, but with a lower recall in stemming (*stm*) with respect to NLP-based techniques. This difference shows more clearly in the case of morphological families (*fam*), which also deals with derivational morphology. Nevertheless, only lemmatization continues beating stemming. The column TD*lem* contains the results we would submit to CLEF competition at this moment, that is, the results the obtained with *lem* technique with the new conditions and employing only title + description topic fields.

Table 5. CLEF 2002: performance measures

	stm	***lem***	***syn***	***fam***	***f-sdp***	TD***lem***
Documents retrieved	50,000	50,000	50,000	50,000	50,000	50,000
Rel. docs retrieved (2854 exp.)	2,570	2,593	2,582	2,624	2,577	2504
R-precision	0.4892	0.4924	0.4721	0.4772	0.4317	0.4443
Aveg. non-interpolated precision	0.5097	0.5186	0.5057	0.4971	0.4546	0.4592
Avg. document precision	0.5255	0.5385	0.5264	0.5170	0.4560	0.4910
11-points avg. precision	0.5239	0.5338	0.5192	0.5155	0.4733	0.4764
3-points avg. precision	0.5193	0.5378	0.5249	0.5109	0.4605	0.4764

Table 6. CLEF 2002: average precision at 11 standard recall levels

Recall	Precision					
	stm	***lem***	***syn***	***fam***	***f-sdp***	TD***lem***
0.00	0.8887	0.8859	0.8492	0.8783	0.8758	0.8446
0.10	0.7727	0.7888	0.7753	0.7637	0.7664	0.7210
0.20	0.6883	0.7096	0.6965	0.6721	0.6704	0.6420
0.30	0.6327	0.6417	0.6246	0.6108	0.5936	0.5740
0.40	0.5909	0.6025	0.5848	0.5724	0.5265	0.5506
0.50	0.5465	0.5628	0.5447	0.5310	0.4458	0.4945
0.60	0.5041	0.4918	0.4720	0.4708	0.3861	0.4226
0.70	0.4278	0.4214	0.4109	0.4144	0.3309	0.3608
0.80	0.3231	0.3410	0.3336	0.3296	0.2654	0.2928
0.90	0.2456	0.2595	0.2547	0.2647	0.2131	0.2103
1.00	0.1422	0.1666	0.1653	0.1628	0.1322	0.1276

7 Conclusion

According to the results obtained for CLEF 2001 and CLEF 2002 topics, content word lemmatization (*lem*) seems to be the best conflation option, even when it only deals with inflectional variation. Nevertheless, it has a better behavior than standard stemming (*stm*), which also deals with derivational variation.

Our approach for solving lexical variation by means of query expansion through synonymy (*syn*) does not improve the results obtained, due to the noise introduced. A different approach, similar to relevance feedback, based on the expansion of the most relevant terms of the most relevant documents, may be more appropriate. Traditional automatic relevance feedback, followed by a phase of filtering and re-weighting of synonyms in the terms generated during expansion is another possibility.

In the case of derivational variation, the use of morphological families seems to introduce too much noise in the system due to badly constructed families, dealing to a worse performance than expected for single word term conflation (*fam*). Tuning of morphological families, or similar approaches to those proposed for synonymy may solve it.

The same problem is inherited by our proposal for dealing with syntactical variation through the employment of syntactic dependency pairs and morphological families (*f-sdp*).

These results, together with the previous ones obtained in other experiments with different weighting schemes and retrieval models [1, 14, 16], suggest that mere lemmatization is a good starting point. It should be investigated whether this initial search using lemmatization should be followed by a relevance feedback process based on the expansion through synonymy and/or morphological families. Another alternative to study for post-processing consists on the re-ranking of the results by means of syntactic information obtained in form of syntactic dependency pairs.

Acknowledgements

The research reported in this article has been supported in part by Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (grant TIC2000-0370-C02-01), Ministerio de Ciencia y Tecnología (grant HP2001-0044), FPU grants of Secretaría de Estado de Educación y Universidades, Xunta de Galicia (grants PGIDT01PXI10506PN, PGIDIT02PXIB30501PR and PGIDIT02SIN01E) and Universidade da Coruña.

References

1. Miguel A. Alonso Pardo, Vilares Ferro, and Víctor M. Darriba. On the usefulness of extracting syntactic dependencies for text indexing. In Michael O'Neill, Richard F. E. Sutcliffe, Conor Ryan, Malachy Eaton, and Niall J. L. Griffith, editors, *Artificial Intelligence and Cognitive Science*, volume 2464 of *Lecture Notes in Artificial Intelligence*, pages 3–11. Springer-Verlag, Berlin-Heidelberg-New York, 2002.
2. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval.* Addison-Wesley, Harlow, England, 1999.
3. Fco. Mario Barcala Rodríguez, Jesús Vilares Ferro, Miguel A. Alonso Pardo, Jorge Graña Gil, and Manuel Vilares Ferro. 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, September 2002. IEEE Computer Society Press.
4. Thorsten Brants. TnT - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP'2000)*, Seattle, 2000.
5. Chris Buckley, James Allan, and Gerard Salton. Automatic routing and ad-hoc retrieval using SMART: TREC 2. In D. K. Harman, editor, *NIST Special Publication 500-215: The Second Text REtrieval Conference (TREC-2)*, pages 45–56, Gaithersburg, MD, USA, 1993.
6. Santiago Fernández Lanza, Jorge Graña Gil, and Alejandro Sobrino Cerdeiriña. A Spanish e-dictionary of synonyms as a fuzzy tool for infor mation retrieval. In

Actas de las I Jornadas de Tratamiento y Recuperación de Información (JOTRI 2002), León, Spain, September 2002.
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, *Working notes for the CLEF 2001 workshop*, Darmstadt, Germany, September 2001.
8. Jorge Graña Gil, Fco. Mario Barcala Rodríguez, and Miguel A. Alonso Pardo. 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.
9. Jorge Graña Gil, Fco. Mario Barcala Rodríguez, and Jesús Vilares Ferro. 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.
10. Jorge Graña Gil, Jean-Cédric Chappelier, and Manuel Vilares Ferro. 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.
11. Jane Greenberg. Automatic query expansion via lexical-semantic relationships. *Journal of the American Society for Information Science and Technology*, 52(5):402–415, 2001.
12. 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.
13. J. Savoy, A. Le Calve, and D. Vrajitoru. Report on the trec-5 experiment: Data fusion and collection fusion, 1988.
14. Jesús Vilares Ferro, Fco. Mario Barcala Rodríguez, and Miguel A. Alonso Pardo. 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.
15. Jesús Vilares Ferro, David Cabrero Souto, and Miguel A. Alonso Pardo. 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.
16. Jesús Vilares Ferro, Manuel Vilares Ferro, and Miguel A. Alonso Pardo. Towards the development of heuristics for automatic query expansion. In Heinrich C. Mayr, Jiri Lazansky, Gerald Quirchmayr, and Pavel Vogel, editors, *Database and Expert Systems Applications*, volume 2113 of *Lecture Notes in Computer Science*, pages 887–896. Springer-Verlag, Berlin-Heidelberg-New York, 2001.
17. David Yarowsky. A comparison of corpus-based techniques for restoring accents in Spanish and French text. In *Natural Language Processing Using Very Large Corpora*, pages 99–120. Kluwer Academic Publishers, 1999.