Morphological and syntactic processing for Text Retrieval*

Jesús Vilares[1], Miguel A. Alonso[1], and Manuel Vilares[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. vilaress@uvigo.es

http://www.grupocole.org/

Abstract. This article describes the application of lemmatization and shallow parsing as a linguistically-based alternative to stemming in Text Retrieval, with the aim of managing linguistic variation at both word level and phrase level. Several alternatives for selecting the index terms among the syntactic dependencies detected by the parser are evaluated. Though this article focuses on Spanish, this approach is extensible to other languages by simply adapting the grammar used by the parser.

1 Introduction

Natural Language Processing (NLP) has frequently attracted the attention of the Information Retrieval (IR) community. This is because the task of deciding about the relevance of a given document with respect to a query basically consists of deciding whether the text of the document satisfies the information need expressed by the text of the query. To perform this task, IR systems have to deal with *linguistic variation*, that is, the different ways in which the same concept can be expressed. This way, textual information retrieval could be considered as a Natural Language Processing problem.

The research in this field has been mainly focused on English and its employment in other languages has not been studied enough, even when the possibilities of success in Spanish and other similar romance languages seem to be greater than those for English, since their syntax and morphology are more complex.

1.1 Dealing with Linguistic Variation

The lowest level of linguistic variation in natural language is *inflection*, those predictable changes a word undergoes as a result of gender, number, person,

* Supported in part by Ministerio de Ciencia y Tecnología (HF2002-81), FPU grants of Secretaría de Estado de Educación y Universidades (AP2001-2545), Xunta de Galicia (PGIDIT02PXIB30501PR, PGIDIT02SIN01E and PGIDIT03SIN30501PR) and Universidade da Coruña.

mood, time, tense, etc. This first level of variation is generally solved by means of *stemmers*, which reduce the word to its supposed grammatical root, or *stem*, through suffix stripping based on a list of frequent suffixes. In English, the results obtained are satisfactory enough since its inflectional morphology being very simple. Nevertheless, in the case of Spanish, its inflectional morphology is much more complex, with modifications at multiple levels and with many irregularities. The case of verbs also includes the possibility of having attached one, two or three clitic pronouns at the end, which confuses the stemmer. In this context, stemming does not seem to be an accurate solution due to many of these phenomena cannot be managed by such a simple tool. Instead, lemmatization seems to be a better solution.

For lemmatizing the documents, we employ `MrTagoo` [4], a high-performance part-of-speech tagger with lemmatization capability whose input is provided by a linguistically-motivated preprocessor module [2]. `MrTagoo` is based on a second order Hidden Markov Model whose core structure for storage and search has been implemented by means of finite-state automata [6]. Other advanced capabilities of our tagger are the management of unknown words, the possibility of integrating external dictionaries, and the possibility of managing ambiguous segmentations [5]. All these capabilities have been implemented using finite-state techniques in order to maintain linear complexity.

Once the viability of NLP techniques for managing morphological variation at word level has been established, the next step consists of applying phrase-level analysis techniques to reduce the *syntactic variation* present in both documents and queries. In order to do it, it is necessary to identify the syntactic structure of the text by means of parsing techniques. Nevertheless, full parsing of the text is non-viable because of its high computational cost, which makes non-practical its application on a large scale. Moreover, the lack of robustness of such approaches reduces their coverage to a grant extent, particularly in the case of Spanish, due to the lack of freely available resources such as grammars, treebanks, etc. In this context, the employment of *shallow parsing* techniques allows, on the one hand, to reduce the computational complexity and, on the other hand, to increase the robustness.

2 The Shallow Parser

We propose a shallow parser based on a cascade of finite-state transducers consisting of five layers, whose input is the output of the tagger-lemmatizer. Next, we will describe briefly the function of each of these layers.

Layer 0: 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* ($NumP$), and *expressions with a verbal function.*

Layer 1: adverbial phrases and first level verbal groups. This layer 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 call *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 ($AdjP$) —e.g., *muy alto* (very high)— are managed here, together with periphrastic verbal groups —e.g., *tengo que ir* (I have to go)—, which we call *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.

Layer 3: noun phrases. We have considered some complex phenomena in noun phrases (NP), such as the existence of *partitive complements* (PC) —e.g., *ninguno de* (none of)—, in order to cover 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: those preceded by the preposition *por* (by) or $PPby$, those preceded by *de* (of) or $PPof$, and the rest of prepositional phrases or PP.

These layers and the rules of the grammar employed by the parser are explained in detail in [17]. Each of the rules involved in the different stages of the parsing process has been implemented through a finite-state transducer. Unlike other tasks, such as Information Extraction or the extraction of lexical patterns [7], our goal is not to get as output a bracketed version of the input text, with the brackets delimiting its phrases, but to obtain, as pairs, a list of the syntactic dependencies of the text. The formation of such pairs only involves the heads of the phrases, so we only need to retain the lemma of the head, together with its corresponding morphosyntactic features. As a result, our parser behaves as a finite-state filter rather than as a finite-state marker.

2.1 Identification of Syntactic Roles and Syntactic Dependencies

Taking as our working hypothesis that for each verbal head there must exist an associated sentence, we will consider as an end of sentence the appearance of any of these elements: *punctuation marks*, *relatives*, *conjunctions*, and *verbal groups in personal form* when there is no other sentence boundary between such a verbal group and the previous one.

This limitation in the scope of the dependency extractor is not a severe problem in the context it has been designed for, index term extraction for IR, because we are not looking for exhaustivity but for reliability in the dependencies obtained, trying to minimize the noise introduced in the system. What we seek is to identify sentences with an underlying structure of the type:

– Active subject + active predicative verbal group + direct object.

– Active subject + copulative verbal group + attribute.
– Passive subject + passive predicative verbal group + agent.

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 will only take into account the prepositional $PPof$ phrases due to their high reliability. This way, when the system finds a $PPof$ immediately after a noun or prepositional phrase, it is identified as a noun complement.

Subject. The closest noun phrase (NP) preceding a verbal group ($VG2$) in personal form will be considered its subject.

Attribute. For 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. It is the closest NP after an active predicative $VG2$.

Agent. It is the closest $PPby$ following a passive predicative $VG2$.

Prepositional verb complement. Due to the problem of prepositional phrase attachment, we have opted for a strict criterion when searching these complements in order to minimize the noise introduced by erroneous identifications. We will only consider 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 syntactic dependencies existing between them are extracted in the form of pairs that involve:

– 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 its 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 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.

Once the dependencies have been extracted, they are conflated into *complex 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 variation [18]. Our intention is to cover the appearance of both the syntactic and morphosyntactic variants of a term [8].

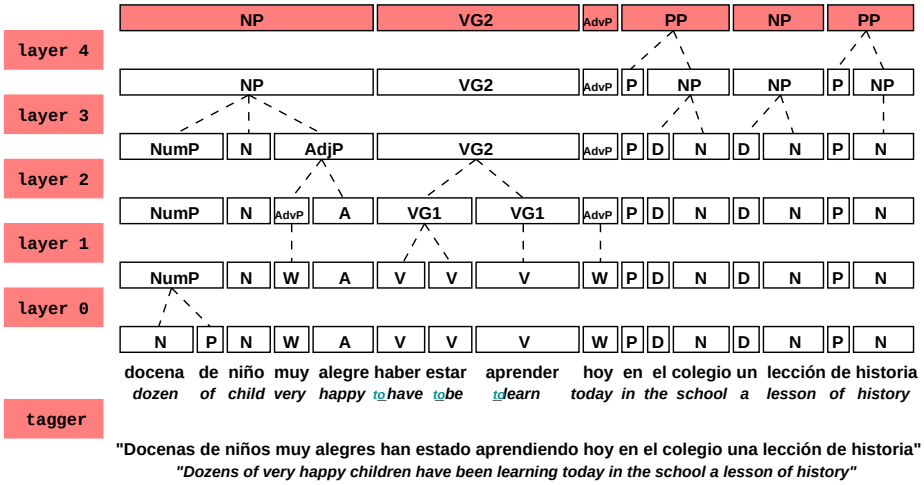

Fig. 1. Overview of the parsing process for the running example

2.2 A Running Example

Let us take the sentence *Docenas de niños muy alegres han estado aprendiendo hoy en el colegio una lección de historia* (Dozens of very happy children have been learning today in the school a lesson of history). The output of the tagger, formed by terns *form* `tag` *lemma*, is converted into the input format required by the parser: *lemma* `tag non-terminal`[1]. In this initial stage, the non-terminal is the grammatical category of the term:

[*docena (dozen)* NCFP N] [*de (of)* P P] [*niño (child)* NCMP N]
[*muy (very)* WQ W][*alegre (happy)* AQFP A]
[*haber (to have)* V3PRI V] [*estar (to be)* VPMS V]
[*aprender (to learn)* VRG V] [*hoy (today)* WI W]
[*en (in)* P P] [*el (the)* DAMS DA] [*colegio (school)* NCMS N]
[*un (a)* DAFS DA] [*lección (lesson)* NCFS N]
[*de (of)* P P] [*historia (history)* NCFS N]

This initial input is then processed by the shallow parser, as is shown, in a summarized way, in Fig. 1. At the output of the cascade we obtain the sequence of heads corresponding to the phrases identified in the parsing process:

[*niño (child)* NCMP NP] [*aprender (to learn)* V3PRI $VG2$]
[*hoy (today)* WI $AdvP$] [*colegio (school)* NCMS PP]
[*lección (lesson)* NCFS N] [*historia (history)* NCFS $PPof$]

[1] For a better understanding of the example, we will complete this notation by adding the translated lemma and by separating the terms by means of square brackets.

Once this first stage has finished, the dependencies contained in the parsed text are extracted in order to be conflated into complex index terms. Firstly, we identify the syntactic roles of the phrases obtained during the parsing, and then the syntactic dependencies existing between them are extracted, except for the case of the dependencies between a noun and its adjectives, which are extracted during the processing of noun phrases in layer 3.

According to the criteria established in Sect. 2.1, we show the syntactic roles of the phrases identified in our running example: an active subject ($SUBJact$), an active predicative verbal group ($Vact$), its prepositional verb complement (PVC), its direct object (DO), and the prepositional noun complement of the latter (PNC):

$$
\begin{array}{llrl}
[\ \textit{niño (child)} & \texttt{NCMP} & NP\ ] & -\ \langle\ SUBJact\ \rangle \\
[\ \textit{aprender (to learn)} & \texttt{V3PRI} & VG2\ ] & -\ \langle\ Vact\ \rangle \\
[\ \textit{hoy (today)} & \texttt{WI} & AdvP\ ] & -\ \langle\ \rangle \\
[\ \textit{colegio (school)} & \texttt{NCMS} & PP\ ] & -\ \langle\ PVC\ \rangle \\
[\ \textit{lección (lesson)} & \texttt{NCFS} & NP\ ] & -\ \langle\ DO\ \rangle \\
[\ \textit{historia (history)} & \texttt{NCFS} & PPof\ ] & -\ \langle\ PNC\ \rangle
\end{array}
$$

Once the syntactic roles of each phrase have been identified, their associated dependencies are extracted according to Sect. 2.1:

$$
\begin{array}{lllll}
ADJ & (\textit{niño} & \texttt{NCMP}, & \textit{alegre} & \texttt{AQFP}) \\
PNC & (\textit{lección} & \texttt{NCFS}, & \textit{historia} & \texttt{NCFS}) \\
SUBJact & (\textit{aprender} & \texttt{V3PRI}, & \textit{niño} & \texttt{NCMP}) \\
DO & (\textit{aprender} & \texttt{V3PRI}, & \textit{lección} & \texttt{NCFS}) \\
PVC & (\textit{aprender} & \texttt{V3PRI}, & \textit{colegio} & \texttt{NCMS})
\end{array}
$$

3 Evaluation

Our conflation approaches have been integrated in the well-known vector-based engine SMART [3], using an `atn-ntc` weighting scheme[2]. The corpus employed for their evaluation is the Spanish monolingual corpus corresponding to CLEF 2003 edition [15]. This corpus is formed by 454,045 news reports (1.06 GB) provided by EFE, a Spanish news agency, corresponding to the years 1994 and 1995. The set of topics consists of 60 queries, from 141 to 200, formed by 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 employed to build the running queries, but giving double relevance to the *title* statement, because it summarizes the basic semantics of the query.

We have compared the behavior of four different conflation approaches:

Stemming (*stm*). This classical approach will be taken as baseline. The tool employed was Snowball Spanish stemmer [1], based on Porter's algorithm, and one of the most popular stemmers employed in research. The stopword list used was that one provided by SMART for Spanish.

[2] Our aim is to investigate whether NLP techniques can be used to improve the performance of (non NLP-based) IR systems. Thus, we have chosen as working environment a classic configuration which can be considered, to a certain extent, standard.

Lemmatization (*lem*). The lemmas of the content words of the text —nouns, adjectives and verbs, the grammatical categories which concentrate the semantics of a text— are used as index terms. The corresponding stopword list was obtained by lemmatizing the content words of the SMART stopword list.

Syntactic dependency pairs obtained from the topic (*tsd*). Based on the combined indexing of lemmatized simple terms —as in the case of *lem*— and complex terms derived from the syntactic dependencies existent in the documents. The dependencies containing stopwords are discarded. The final query submitted to the system is formed by the index terms obtained from the topic through the same process of lemmatization and shallow parsing.

Syntactic dependency pairs obtained from top documents (*dsd*). The indexing process is the same of *tsd*, but the querying process is performed in two stages:

1. The lemmatized query is submitted to the system.
2. The n top documents retrieved by this initial query are employed to select the most informative dependencies, which are used to expand the lemmatized query, but with no re-weighting. These dependencies are selected automatically using Rocchio's approach [16] to feedback. They are selected from the 10 best terms (both lemmas and dependencies) of the top 5 documents. In [9], relevance feedback is also used to select relevant noun phrases. However, our approach is more complete in the sense that we deal with all kind of syntactic dependencies and that our evaluation is performed on a large set of standard queries, which have not been specifically created for the experiments.

The expanded query is then submitted to the system in order to obtain the final set of documents retrieved.

3.1 Initial Experiments

Table 1 shows the results of the experiments performed. Each column contains one of the parameters employed to measure the performance: number of documents retrieved, number of relevant documents retrieved (2368 expected), average precision (non-interpolated) for all relevant documents (averaged over queries), average document precision for all relevant documents (averaged over relevant documents), R-precision, and precision at N documents retrieved.

The first row, *stm*, contains the results for the baseline, stemming. The second row, *lem*, shows the results of lemmatization, whereas the next row, Δlem, shows its improvement with respect to *stm*. As it can be seen, the improvement is clear. The next two rows, *tsd*1 and *tsd*10, contain the results obtained using the syntactic dependency pairs obtained from the topic when the weight of simple terms is multiplied by 1 and 10 with respect to complex terms. As indicated in [13], when simple and complex terms are used together as index terms, the assumption of term independence is violated because words forming a dependency-pair also occur in the documents from which the dependency has

Table 1. Experimental results on the CLEF 2003 collection

	Docs.	Rlv.	Precision			Precision at N docs.								
			N.I.	Doc.	R	5	10	15	20	30	100	200	500	1000
stm	57k	2216	.4577	.5145	.4372	.5754	.5070	.4585	.4175	.3567	.2247	.1422	.0702	.0389
lem	57k	2221	.4681	.5431	.4471	.5965	.5000	.4515	.4281	.3813	.2314	.1455	.0718	.0390
Δlem	57k	5	.0104	.0286	.0099	.0211	-.0070	-.0070	.0106	.0246	.0067	.0033	.0016	.0001
tsd1	57k	2218	.4014	.4647	.3961	.4842	.4333	.3860	.3632	.3205	.2053	.1368	.0692	.0389
tsd10	57k	2242	.4710	.5475	.4454	.6070	.5053	.4503	.4237	.3789	.2337	.1461	.0720	.0393
Δtsd10	57k	26	.0133	.0330	.0082	.0316	-.0017	-.0082	.0062	.0222	.0090	.0039	.0018	.0004
dsd1	57k	2256	.4741	.5538	.4425	.5825	.5070	.4561	.4211	.3731	.2328	.1486	.0733	.0396
dsd3	57k	2255	.4747	.5562	.4454	.5930	.5070	.4550	.4246	.3825	.2353	.1481	.0729	.0396
Δdsd3	57k	39	.0170	.0417	.0082	.0176	.0000	-.0035	.0071	.0258	.0106	.0059	.0027	.0007
Experiments with pseudo-relevance feedback														
stm	57k	2253	.5048	.5908	.4749	.5754	.5281	.4772	.4456	.4000	.2426	.1519	.0733	.0395
lem	57k	2260	.5211	.6086	.4796	.6000	.5421	.4982	.4640	.4105	.2461	.1527	.0742	.0396
Δlem	57k	7	.0163	.0178	.0047	.0246	.0140	.0210	.0184	.0105	.0035	.0008	.0009	.0001
tsd10	57k	2266	.5125	.6011	.4681	.6070	.5439	.4901	.4500	.4012	.2437	.1542	.0742	.0398
Δtsd10	57k	13	.0077	.0103	-.0068	.0316	.0158	.0129	.0044	.0012	.0011	.0023	.0009	.0003
dsd3	57k	2258	.5151	.6047	.4752	.5965	.5421	.4901	.4535	.4058	.2418	.1530	.0741	.0396
Δdsd3	57k	5	.0103	.0139	.0003	.0211	.0140	.0129	.0079	.0058	.0008	.0011	.0008	.0001

been extracted. To address this problem, we must increase the weight of simple terms relative to the weight of complex terms. Training experiments were performed on the smaller CLEF 2001/2002 collection, consisting of 215,738 documents and 100 queries. We found that balance factors between *7 to 1* and *10 to 1* obtained the best results. Among them, the balance factor *10 to 1* obtained better precision at the top ranked documents. As we can observe, with the CLEF 2003 collection, *tsd10* obtains the best results for precision at N documents, and non-interpolated and document precision, being better than those obtained through stemming (*stm*) and lemmatization (*lem*). Its improvement with respect to *stm* is shown in *Δtsd10*.

The next two rows of table 1 correspond to syntactic dependency pairs obtained from top documents (*dsd*). As in the case of *tsd*, we have introduced a balance factor between the weight of simple and complex terms, The interferences introduced through this new approach due to the violation of term independence are much lesser than when using the dependencies from the topics, as we can immediately observe when we compare *tsd1* and *dsd1*. These new results are similar, and even better in some cases, to those reached with *tsd10*; the only exception is the precision at 5 top documents. The best behavior of *dsd* was obtained with a balance factor of *3 to 1*, whose corresponding results are shown in the row *dsd3*, together with its improvement with respect to *stm*, shown in *Δdsd3*.

From these experiments, we can conclude that the pairs chosen automatically by the system are much more accurate than those obtained directly from the

topic, as it seems to be demonstrated when we compare *tsd1* and *dsd1*. Comparing the pairs obtained from topics and those obtained automatically from top documents, we have found that only a small set of pairs —28 to be precise, 5.92%— of the topic pairs were chosen by the system from top documents. This points at that the rest of them do not represent accurately the semantics of the topic. Nevertheless, these common terms represent more than a quarter —28.87%— of the terms selected automatically by the system, which indicates us that the contribution of the syntactic information of the topic continues to be important, but that it has to be adequately filtered and selected.

With respect to lemmatization, the approaches based on the employment of syntactic dependencies show a behavior not so good as expected, obtaining only slight improvements in recall, non-interpolated precision, and document precision.

3.2 Experiments Using Pseudo-Relevance Feedback

A second group of experiments has been performed in order to compare the behavior of our NLP-based conflation approaches with respect to stemming during pseudo-relevance feedback (blind-query expansion). For these tests we have adopted Rocchio's approach [16], expanding the original topic with the best 10 terms of the 5 top ranked documents. As a result of a tuning process, the parameter α, which stands for the contribution of the original query vector, has been set at 1.40, whereas β, which stands for the contribution of the vectors of the relevant documents, has been set at 0.10. We have not considered any contribution from non-relevant documents, and so γ has been set to 0.

The results obtained for these new experiments are shown in the bottom part of Table 1. As we can see, NLP-based conflation techniques clearly outperform stemming (*stm*). The best behavior corresponds to lemmatization (*lem*) whereas the employment of syntactic dependencies in *tsd* and *dsd*, which outperform stemming, obtain quite similar results to those of lemmatization.

4 Conclusions

Throughout this article we have studied the employment of Natural Language Processing techniques in Text Retrieval as an alternative to stemming for managing linguistic variation. Their implementation by means of finite-state techniques result in a minimal overhead with respect to classical techniques, a key issue for their employment in practical environments.

Three different approaches have been tested. The first one employs lemmatization to solve linguistic variation derived from inflection. The other two approaches employ shallow parsing to manage syntactic variation by using syntactic dependencies as complex index terms.

The results obtained show that lemmatization seems to be, at this moment, the best option for Spanish conflation, since the improvement obtained using syntactic information is not so good as expected. Nevertheless, our experiments

seem to indicate that the employment of syntactic information must not to be discarded, but the way it is employed should be reconsidered. Our experiences point at its employment for refining the results obtained through lemmatization, due to the noise introduced by syntactic dependencies when they are not accurately selected. As an alternative, syntactic dependencies could be used as a base to construct conceptual graphs representing the semantic of sentences [11, 12, 14, 10], paying the price of a higher computational cost.

References

1. `http://snowball.tartarus.org` (site visited October 2003).
2. F. M. Barcala, J. Vilares, M. A. Alonso, J. Graña, and M. Vilares. Tokenization and proper noun recognition for information retrieval. In *DEXA Workshop 2002*, pages 246–250. IEEE Computer Society Press, 2002.
3. C. Buckley. Implementation of the SMART information retrieval system. Technical report, Department of Computer Science, Cornell University, 1985.
4. J. Graña. *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*. PhD thesis, University of La Coruña, La Coruña, Spain, 2000.
5. J. Graña, M. A. Alonso, and M. Vilares. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. *Lecture Notes in Computer Science (LNCS)*, 2448:3–10, 2002.
6. J. Graña, F. M. Barcala, and M. A. Alonso. Compilation methods of minimal acyclic automata for large dictionaries. *LNCS*, 2494:135–148, 2002.
7. G. Grefenstette, A. Schiller, and S. Aït-Mokhtar. Recognizing lexical patterns in text. In F. Van Eynde and D. Gibbon, editors, *Lexicon Development for Speech and Language Processing*, pages 141–168. Kluwer Academic, Dordrecht, 2000.
8. C. Jacquemin and E. Tzoukermann. NLP for term variant extraction: synergy between morphology, lexicon and syntax. In T. Strzalkowski, editor, *Natural Language Information Retrieval*, pages 25–74. Kluwer Academic, Dordrecht, 1999.
9. M. S. Khan and S. Khor. Enhanced web document retrieval using automatic query expansion. *JASIST*, 55(1):29–40, 2004.
10. C. S.-G. Khoo. The use of relation matching in Information Retrieval. LIBRES: Library and Information Science Research, 7(2), 1997.
11. M. Montes-y-Gómez, A. Gelbukh, A. López-López and R. Baeza-Yates. Flexible Comparison of Conceptual Structures. *LNCS*, 2113:102–111, 2001.
12. M. Montes-y-Gómez, A. López-López and A. Gelbukh. Information Retrieval with conceptual graph matching. *LNCS*, 1873:312–321, 2000.
13. M. Narita and Y. Ogawa. The use of phrases from query texts in information retrieval. In *Proc. of ACM SIGIR 2000*, pages 318–320, Athens, Greece, 2000.
14. S. Nicolas, B. Moulin and G. W. Mineau. Sesei: A CG-based filter for Internet search engines. *Lecture Notes in Artificial Intelligence (LNAI)*, 2746:362–377, 2003.
15. C. Peters and F. Borri, editors. *Results of the CLEF 2003 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway, August 2003.
16. J. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.

17. J. Vilares, and M. A. Alonso. A Grammatical Approach to the Extraction of Index Terms. *Proceedings of International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, pages 500-504, Borovets, Bulgaria, 2003.
18. J. Vilares, F. M. Barcala, and M. A. Alonso. Using syntactic dependency-pairs conflation to improve retrieval performance in Spanish. *LNCS*, 2276:381–390, 2002.