Managing Syntactic Variation in Text Retrieval

Jesús Vilares
Departamento de Computación
Universidade da Coruña
Campus de Elviña s/n
15071 - A Coruña (Spain)
jvilares@udc.es

Carlos Gómez-Rodríguez
Departamento de Computación
Universidade da Coruña
Campus de Elviña s/n
15071 - A Coruña (Spain)

Miguel A. Alonso
Departamento de Computación
Universidade da Coruña
Campus de Elviña s/n
15071 - A Coruña (Spain)
alonso@udc.es

ABSTRACT

Information Retrieval systems are limited by the linguistic variation of language. The use of Natural Language Processing techniques to manage this problem has been studied for a long time, but mainly focusing on English. In this paper we deal with European languages, taking Spanish as a case in point. Two different sources of syntactic information, queries and documents, are studied in order to increase the performance of Information Retrieval systems.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods, Linguistic processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*

General Terms

Algorithms

Keywords

Information retrieval, natural language processing, shallow parsing

1. INTRODUCTION

Natural Language Processing (NLP) has attracted the attention of the Information Retrieval (IR) community [10]. This is because one of the major limitations IR systems have to deal with is *linguistic variation* [3], that is, the different ways in which the same concept can be expressed, 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 and other similar Romance languages. When managing this phenomena, the use of Natural Language Processing techniques becomes feasible.

DocEng'05, November 2–4, 2005, Bristol, United Kingdom.

Word-level NLP techniques have been applied in order to reduce the *morphological variation* due to inflection and derivation [12, 13]. Moreover, the employment of such techniques does not imply an increase in computational cost with respect to classical techniques such as *stemming*, because they can be implemented through finite-state automata and transducers.

Once the viability of word-level NLP techniques has been established, the next step consists of applying phrase-level analysis techniques in order to increase the precision of index terms and to deal with *syntactic variation*. In this paper we study how the use of syntactic information can improve the performance of Information Retrieval systems, analyzing the performance of different approaches for managing the syntactic variation of texts.

2. THE SHALLOW PARSER

When processing the syntactic content of a text, the first step consists in obtaining its syntactic structure. Nevertheless, full parsing of the text is non-viable because of its high computational cost, which makes its application on a large scale impractical. Moreover, the lack of robustness of such approaches greatly reduces their coverage, 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 [1] allows us to reduce computational complexity and increase robustness. Shallow parsing has shown itself to be useful in several NLP application fields, particularly in Information Extraction [7], although its application in IR has not yet been studied in depth.

The shallow parser employed by our system consists of five layers, the input of the parser being the output of a tagger-lemmatizer [5]. 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 finite-state cascade-based parser which maintains a linear complexity. The first layer, *layer 0*, preprocesses quantity and verbal expressions. Next, *layer 1* identifies adverbial phrases and non-periphrastic verbal groups, *layer 2* whilst deals with adjectival phrases and periphrastic verbal groups. *Layer 3* manages noun phrases. Finally, *layer 4* processes prepositional phrases. These layers and the grammar rules employed by the parser are explained in detail in [11].

Our goal is to obtain those pairs of words related through the most significative syntactic dependencies, those of the type *noun-modifier*, *subject-verb*, *verb-object*, etc. Once the

dependencies have been extracted, they are conflated into *complex index terms* [8] in order to complement simple terms, since their degree of specificity is greater than those for their individual component terms. In our case, we have used a conflation technique based on the use of morphological relations in order to improve the management of syntactic variation [2]. Our intention is to cover the appearance of both the syntactic and morphosyntactic variants of a term [8] —e.g., *una caída de las ventas* (a drop in the sales) vs. *una caída de ventas* (a drop in sales) vs. *las ventas han caído* (sales have dropped).

3. EXPERIMENTAL RESULTS

Our approaches have been integrated in the well-known vector-based engine SMART [4], using an `atn-ntc` weighting scheme. The evaluation was made using the Spanish corpus employed in CLEF 2001/02 editions[1], formed by 215,738 news reports (509 MB), and the 46 odd-numbered topics of those editions (41 to 140) with more than 5 relevant documents. These topics are formed by three fields: a brief *title* statement, a one-sentence *description*, and a more complex *narrative* specifying the relevance assessment criteria. Only *title* and *description* statements have been employed.

The results obtained for our different conflation approaches are shown in Table 1, each row containing one of the parameters employed to measure the performance: number of documents retrieved, number of relevant documents retrieved, 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. Stemming (*stm*) through the Snowball Spanish stemmer[2] —based on Porter's algorithm— is taken as baseline. As a second reference, we have included the results obtained through the indexing of the content word lemmas —nouns, adjectives and verbs— of the text (*lem*) [2, 12]. For each conflation approach we also show the degree of improvement attained with respect to stemming (*%Δ*) —those results for which we have obtained positive improvement appearing in boldface.

3.1 Dependencies extracted from queries

In this first set of experiments, both simple and complex terms are combined and indexed. Simple terms consist of the lemmatized content words of the text [13], whereas complex terms consist of the conflated syntactic dependency pairs of the text. 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. This way, we decided to discard those terms contained in less than five documents of the collection (i.e., $df<5$). During querying process, the conflating process is the same, extracting and combining the lemmas and dependencies of the query. The results obtained are shown in column *qdC* of Table 1, showing a positive and consistent improvement both in global results and in precision in the top documents retrieved.

In order to compare our proposal with other classical approaches based on indexing only the information extracted from noun phrases [6], we have included a new set of results obtained using only those dependencies corresponding to noun phrases, that is, those dependencies between a noun and each of its modifying adjectives, and between a noun and the head of its prepositional complements. The results obtained —see column *qnC*— are very similar, with only slightly lesser improvements than those obtained employing all type of dependencies. However, the number of unique dependencies to be indexed is 48% less, with the consequential reduction in the size of the index. At this point, we should also remark that the computational cost of the dependency generation process remains the same, because noun and prepositional phrases are identified in the last layers of the parser.

	qdC	***qdC ∩ ddC***
noun–adjective	37.50%	57.89%
noun–complement	45.83%	34.21%
subject–active verb	1.38%	–
verb–object	6.25%	2.63%
subject–passive verb	0.69%	–
verb–complement	9.02%	5.26%

Table 2: Distribution of types of dependencies

3.2 Dependencies extracted from documents

A second set of experiments was made employing not the syntactic information obtained from the queries, but that obtained from the documents. In this way, the indexing process is the same as before, combining and indexing both content word lemmas and dependency pairs, but the querying process is now performed in different stages. Firstly, the lemmatized query is submitted to the system. Next, the most informative dependencies of the top documents retrieved with this initial query are used to expand it, but with no re-weighting. Such dependencies are selected automatically using Rocchio's approach to feedback [9], taking the dependencies contained between the t' best terms (both lemmas and dependencies) of the n'_1 top documents retrieved. These parameters are estimated in a previous tuning phase. Finally, the expanded query is then submitted to the system in order to obtain the final set of documents retrieved.

The results obtained with this new approach are presented in column *ddC* (n'_1=10, t'=50), showing a clear and consistent improvement at all levels with respect to the employment of the dependencies extracted from the queries.

Due to the positive results obtained with this new approach, we decided to investigate the possibility of there being some kind of relation with respect to the index terms introduced by each type of syntactic relation. Thus, as is shown in Table 2, we studied the distribution of the different types of syntactic dependencies corresponding to those pairs obtained from the queries in the previous approach (*qdC*), and to those pairs common to those extracted from the documents (*qdC ∩ ddC*). Our aim was to find any bias or preference. As can be seen, dependencies corresponding to noun phrases —those between a noun and its modifying adjectives, and between a noun and the head of its prepositional complement— seem to be preferred.

[1] `http://www.clef-campaign.org`
[2] `http://snowball.tartarus.org`

	stm	*lem*	%Δ	*qdC*	%Δ	*qnC*	%Δ	*ddC*	%Δ	*dnC*	%Δ
Documents retrieved	46k	46k	–	46k	–	46k	–	46k	–	46k	–
Relevant (3007 expected)	2719	2700	-0.70	**2728**	**0.33**	**2726**	**0.26**	**2758**	**1.43**	**2756**	**1.36**
Non-interpolated precision	.4720	**.4829**	**2.31**	**.4965**	**5.19**	**.4940**	**4.66**	**.5286**	**11.99**	**.5190**	**9.96**
Document precision	.5155	**.5327**	**3.34**	**.5491**	**6.52**	**.5457**	**5.86**	**.5768**	**11.89**	**.5761**	**11.76**
R-precision	.4599	**.4848**	**5.41**	**.4895**	**6.44**	**.4858**	**5.63**	**.5119**	**11.31**	**.5001**	**8.74**
Precision at 5 docs.	.6391	**.6609**	**3.41**	**.6913**	**8.17**	**.6870**	**7.49**	**.7000**	**9.53**	**.7043**	**10.20**
Precision at 10 docs.	.5935	**.6283**	**5.86**	**.6500**	**9.52**	**.6522**	**9.89**	**.6717**	**13.18**	**.6717**	**13.18**
Precision at 15 docs.	.5551	**.5928**	**6.79**	**.6029**	**8.61**	**.5971**	**7.57**	**.6203**	**11.75**	**.6203**	**11.75**
Precision at 20 docs.	.5174	**.5446**	**5.26**	**.5620**	**8.62**	**.5587**	**7.98**	**.5935**	**14.71**	**.5837**	**12.81**
Precision at 30 docs.	.4710	**.4928**	**4.63**	**.5036**	**6.92**	**.5036**	**6.92**	**.5348**	**13.55**	**.5326**	**13.08**
Precision at 100 docs.	.3157	**.3300**	**4.53**	**.3348**	**6.05**	**.3361**	**6.46**	**.3474**	**10.04**	**.3470**	**9.91**
Precision at 200 docs.	.2186	**.2234**	**2.20**	**.2263**	**3.52**	**.2257**	**3.25**	**.2336**	**6.86**	**.2315**	**5.90**
Precision at 500 docs.	.1097	.1090	-0.64	**.1103**	**0.55**	**.1101**	**0.36**	**.1117**	**1.82**	**.1123**	**2.37**
Precision at 1000 docs.	.0591	.0587	-0.68	**.0593**	**0.34**	**.0593**	**0.34**	**.0600**	**1.52**	**.0599**	**1.35**

Table 1: Experiments using the CLEF corpus

We were therefore able, as before, to obtain a new set of results using only those dependencies corresponding to noun phrases. Such results are presented in column *dnC* of Table 1 (n'_1=10, t'=40), showing, also as before, a slight decrease in the improvements obtained, particularly in the case of global measures.

4. CONCLUSIONS

Throughout this article we have studied the use of syntactic dependencies as complex index terms in an attempt to improve the performance of Information Retrieval systems by, on the one hand, increasing the precision of index terms, and on the other, by dealing with *syntactic variation*. To extract such dependencies, both documents and queries are processed by means of a finite-state shallow parser, it being fast and robust enough to face the processing of extensive text collections. The results we have shown here are encouraging, particularly when employing the syntactic information extracted from documents. Moreover, storage resources can be saved by restricting the dependencies employed to those corresponding to noun phrases, with a slight reduction of the improvement obtained. It should be noted that, although these experiments were made for Spanish, our approach can be applied to any Romance language.

5. ACKNOWLEDGMENTS

Supported in part by Ministerio de Educación y Ciencia and FEDER (TIN2004-07246-C03-02), and Xunta de Galicia (PGIDIT02PXIB30501PR, PGIDIT02SIN01E and PGIDIT03SIN30501PR).

6. REFERENCES

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

[2] M. A. Alonso, J. Vilares, and V. M. Darriba. On the usefulness of extracting syntactic dependencies for text indexing. In volume 2464 of *Lecture Notes in Artificial Intelligence*, pages 3–11. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

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

[4] C. Buckley. Implementation of the SMART information retrieval system. Technical report, Cornell University, 1985. Source code available at `ftp://ftp.cs.cornell.edu/pub/smart`.

[5] J. Graña. *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*. PhD thesis, University of A Coruña, Spain, 2000.

[6] M. Hearst, J. Pedersen, P. Pirolli, H. Schutze, G. Grefenstette, and D. Hull. Xerox site report: Four TREC-4 tracks. In *The Fourth Text REtrieval Conference (TREC-4)*, pages 97–119, 1996.

[7] J. R. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In *Finite-State Language Processing*. MIT Press, 1997.

[8] C. Jacquemin and E. Tzoukermann. NLP for term variant extraction: synergy between morphology, lexicon and syntax. In Strzalkowski [10], pages 25–74.

[9] J. Rocchio. *The SMART Retrieval System - Experiments in Automatic Document Processing*, chapter Relevance feedback in information retrieval, pages 313–323. Prentice-Hall, NJ, 1971.

[10] T. Strzalkowski, editor. *Natural Language Information Retrieval*. Kluwer Academic Publishers, 1999.

[11] J. Vilares and M. A. Alonso. A grammatical approach to the extraction of index terms. In *International Conference on Recent Advances in Natural Language Processing, Proceedings*, pages 500–504, Borovets, Bulgaria, 2003.

[12] J. Vilares, M. A. Alonso, F. J. Ribadas, and M. Vilares. COLE experiments at CLEF 2002 Spanish monolingual track. In volume 2785 of *Lecture Notes in Computer Science*, pages 265–278. Springer-Verlag, Berlin-Heidelberg-New York, 2003.

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