Using WordNet to Improve Information Retrieval Accuracy

Yenory Rojas; Antonio Ferrández; Jose Navarro

Dept. Languages and Information Systems
University of Alicante
Carretera San Vicente S/N - Alicante - Spain - 03080
+34-96-590-3400
yrojas@dlsi.ua.es antonio@dlsi.ua.es jnavarro@dlsi.ua.es

Abstract. We have implemented an entity relation monolingual Information Retrieval model, for both English and Spanish. This model reports an important precision improvement with respect to traditional IR systems. Then, we extended it with WordNet. This paper explores the utility of use WordNet synsets instead of word representations at index level. Also we have used this tool in query expansion. The training set used in the experiments was Spanish and English corpora and queries from Cross Language Evaluation Forum (CLEF). Using WordNet we have obtained an improvement near of 40% respect to cosine baseline.

1 Introduction

An Information Retrieval (IR) application takes as input a user's query and it has to return a set of documents sorted by their relevance to the query. Nowadays, this kind of application is very important because of the high increase of information available to the users, mainly through Internet.

In literature, the Natural Language Processing (NLP) techniques have not reported significant improvement in retrieval performance. Although it seems that they may overcome the inadequacies of purely quantitative methods of text IR as statistical full-text retrieval or bag of words representations. The works from Strzalkowski or Baeza-Yates are examples of the attempts to overcome these inadequacies[19,4]. As stated there, one possible explanation is that the syntactic analysis is just not going far enough. Alternatively, and perhaps more appropriately, the semantic uniformity predictions made on the basis of syntactic structures are less reliable than we have hoped for. Other problems can be the relatively low quality of parsing and lack of effective weighting techniques for compound terms, which Voorhees claims is an important factor that affects NLP compared to current IR techniques[23].

In this paper, we show an IR model that incorporates NLP techniques and tools such as POS-tagging, partial parsing and WordNet to improve the traditional bag of words representations. This model indexes the entities and the relations between them, these relations are based on the clause splitting of the document, and the resolu-

© A. Gelbukh, C. Yáñez Márquez, O. Camacho Nieto (Eds.)
Advances in Artificial Intelligence and Computer Science
Research on Computing Science 14, 2005, pp. 179-190

tion of anaphora phenomenon between these same entities. Our proposal improves other approaches that use NLP for IR, because it merges more knowledge than other proposals (morphological, syntactical, WordNet and anaphora resolution), and is successfully exploited (increases up to 40% are obtained).

The following section presents the antecedents of the incorporation of NLP for IR tasks. Later, section 3 presents the model proposed. Next, the model evaluation on the English and Spanish CLEF corpora is shown in section 4. Finally, in section 5 contains paper conclusions and future works.

2 Antecedents

The traditional statistical IR systems search for the words of the user's query in documents, so that they consider relevant the documents that have these words. They sort the relevant documents by using different measures of similarity (e.g. the cosine, pivoted cosine or Okapi measures).

In order to improve the effectiveness of the IR systems, several research lines have arisen, for example the Passage Retrieval models, and the application of NLP techniques. However, so far the NLP techniques have not obtained significant improvement in regarding to the computational effort that supposes the utilization of these techniques.

The IR systems that use NLP can be classified according to the kind of NLP knowledge they use. For example, some of them take morphological values to use the lemma instead of the word stems, as well as several morphologic derivations, e.g. Vilares et al [21].

Other systems use query expansion techniques by means of adding new terms obtained from synonyms gathered from WordNet, e.g. Gonzalo et al. [9] or Arampatzis et al. [3], where they usually improve the recall, but they make the precision worse.

Finally, the third kind of knowledge that has been extensively used for IR is the syntactic; its basic idea is to index groups of words that are in relation instead of separated words, as occurs in the traditional IR systems. The main problem arisen by these systems is that the same concept can be expressed in terms of different syntactic trees, therefore a sort of similarity measures between different trees has to be used. Another problem is the quality, depth and robustness of the syntactic parsing. Many systems have tried to avoid these problems by means of indexing just contiguous words as pairs, ternary expressions (e.g. works of Zhai et al. [25], Mitra et al. [13] or Strzalkowski et al. [20]) or phrases (e.g. work of Arampatzis et al. [3]). In regarding to the pair and ternary expressions, these systems usually index the head of the constituents (mainly noun and verbal phrases) jointly with their modifiers. For example, Byung-Kwan et al. index just Korean compound nouns with only an increase of 0.84% in the average precision, with respect to the phrases, they have to devise complex measures of similarity between syntactic trees [6].

Some systems try to mix several kinds of knowledge, even jointly with the vector model, as occurs in Cornelis' work [7]. Another example is the use of head-modifier pairs to create a new indicator presented by Strzalkowski et al. [20]. Along with stems of the words, and other streams of data (pairs), they are able to improve the vector model close to 7% for the average precision in short questions (with few words) and 20% for long questions (more descriptive). Another similar work is reported by Alonso et al. [1], in which the authors combine stems, lemmas and derivation, with head-modifier pairs obtaining only 1.59% improvement.

Our approach utilizes a POS tagger to obtain morphological information using the lemma of each word as well as their lexical category (proper or common nouns, verbs, etc.). For English we have used TreeTagger[1], and Maco[2] for Spanish as POS tagger tools. For the syntactical parsing we have used a *Slot Unification Parser for Anaphora Resolution (SUPAR)*[8]. SUPAR is a partial parser that performs a deep parsing of the constituents that we consider important for extracting the concepts (noun phrases, coordination, juxtaposition, relative clauses, appositions, prepositional phrases, etc.) and relations of these concepts (clause segmentation). In addition our system uses WordNet for the query expansion and also to complement the index.

The expected success of our proposal is getting the better representation of the presented information merge.

3 Model Description

In this approach, queries and documents have been represented as vectors in an n-dimensional space, where the index and query terms are "entities", instead of simple words. In order to measure the similarity between those vectors, any term weighting model should be used.

The entities are represented by means of noun phrases (NP), and the relations between them are represented by means of clauses, in which the verb is the head and its modifiers are the NP and prepositional phrases (PP). These relations are obtained using syntactic analysis. Then, that have been completed by means of resolving anaphora. In Example 1 we can observe this, firs part of the model in which we can find two entities: *Lance Armstrong* and *Lance Raave*, and from the syntactic knowledge, we obtain additional information about the second one (*the accountant of UCI*). Because the anaphoric reference between *him* and *Lance Armstrong* is resolved, more information about this entity is obtained (*Lance Armstrong* is *the president of LS Foundation*).

Example 1 Lance Armstrong arrived first, so Lance Raave who is the accountant of UCI prized him, the president of LSF, with 10000€.

3.1 Extracting Entities

Each phrase is translated to three tables: NPT, PPT and CCT. The NPT stores information about the entities syntactically represented as noun phrases (NP). The PPT table stores the preposition as well as the head of the NP, for example, in the query *ar-*

---

[1] For TreeTagger information see http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[2] For MACO information see http://nipadio.lsi.upc.es/cgi-bin/demo/demo.pl

chitecture in San Diego, the preposition *in* means that *San Diego* may be a place entity, instead of a person entity. Therefore, a document in which appears a PP *in San Diego* is valued higher than other documents in which *San Diego* does not appear with that preposition. The CCT table stores the verb and all the content words in the clause. With this model structure extra information is stored, e.g. the prepositions and pronouns which belong to the list of stop-words in other models, as well as the information of each entity and the relations between them. As example in the second clause of Example 1, the CCT stores the verb *prize* jointly with the NP *Lance Armstrong* and *Lance Raave* and the *10000€ prize.*

The Example 2 has been extracted from the CLEF corpus Los Angeles Times 1994 (file la010194, news 166), in which the real operation of our system is presented. Here, the NP are marked between [ ], the prepositions are marked in *italics* and the verbs are underlined. In addition, one case of resolution of the anaphora is marked (crossed). The change of clause is marked by "¬".symbol. The case $_{(1)}$ marks an apposition and the case $_{(2)}$ a relative clause. The number of the sentence is indicated in bold face.

Example 2: Parsing, Clause Splitting and Anaphora Resolution:

5 [HUNTINGTON BANK ROBBERY] NETS [$ 780]
6 [A man] walked *into* [a bank Friday] ¬ , warned [a teller] ¬ that [he ~~(a man)~~] had [a gun] ¬ and made off *with* [$ 780] ¬ , [police] said.
7 [Huntington Beach Police Sgt. Larry Miller] said ¬ [the teller] *at* [[the World Savings] and [Loan Assn., $_{(1)}$[6902 Warner Ave.],]] did not see [a weapon] *during* [the robbery], $_{(2)}$which occurred *at* [4:35 p.m.]
8 [The robber] escaped out [the west door *of* [the building]].
9 [Police] have [no suspects *in* [the case]].

Head	Modifiers	POS	Sent.	Frequency
NPT table				
435	[pm]	CD	7	1
6902	[warner, av, loan, assn]	CD	7	1
Assn	[warner, av, loan, 6902]	PN	7	1
Av	[warner, loan, assn, 6902]	PN	7	1
Bank	[huntintgon, robberi], [fridai]	PN, NN	5, 6	2
Beach	[huntington, polic, sgt, larri, miller]	PN	7	1
Build	[]	NN	8	1
...	...	...	...	...

Preposition	NP heads	Sent.	Frequency
PPT table			
At	[world, save, warner, av, loan, assn], [pm]	7	2
Dure	[robberi]	7	1
In	[case]	9	1
Into	[bank, fridai]	6	1
Of	[build]	8	1
With	[780]	6	1

Verb	Content words	Sent.	Frequency
CCT table			
Escap	[robber, west, door, build]	8	1
Have	[gun], [polic, suspect, case]	6, 9	2
Make	[780, man]	6	1
Net	[huntington, bank, robberi, 780]	5	1
Occur	[435, pm]	7	1
Sai	[huntington, beach, polic, sgt, larri, miller]	6, 7	2
...	...	...	...

Table 1. NPT, PPT and CCT tables obtained from Example 2

As can be seen from Table 1, each structure stores the head of the constituent (used to search for the constituent) and a list of entity modifiers (used to make it more precise). For each entry in the tables, the standard vector frequency information is also stored: the frequency of the entity in the document and the frequency in the document collection. Complex NP´s are represented as structures composed of phrase heads and modifiers. In the NPT table the POS of the head is also store.

3.2 Optional Modules

Two optional modules have been incorporated to our system, one for query expansion, and other to complement the index.

The query expansion module uses two techniques. The first expansion technique consists of extracting the 20 most frequent constituents in the 2 most relevant documents. These constituents are added to the original query with frequency 1 and with no modifiers. In the case of long queries, we obtained a better result using Wordnet to filter the terms to be added. Only the terms that have one WordNet sense or those that are synonymous to those of the original query are added. The second expansion technique consists of a variation of the *windowing* technique proposed by Amo et al. [2], in which we use the first 5 relevant documents to modify the similarity value of the remaining documents, by means of benefiting those documents that share modifiers with the first 20 most frequent terms of the first 5 relevant documents.

The second module attempts to enrich the existing index, by grouping semantically related terms. The new index entries are the WordNet synsets of the terms, instead of its word representations. With this implementation we try to avoid matching terms with multiple semantically values, as example the term *cream* can be used like a dairy products or an emollient sense, also we should match terms with same senses, for example an user looking for exercise can be retrieve jogging documents

3.3 Searching Strategy

In order to measure the query and the documents similarity, we have modified a measure of similarity of the cosine presented by Kaszkiel et al. [11], to use the NLP knowledge of our model.

The first modification is that the weights are multiplied by two parameters: *NLPfactor* (knowledge obtained from the NLP techniques) and *proximity* (measures the proximity between the query entities in the document), as is presented in equation(1). The second modification is that the first two query syntactic constituents (i.e. NP or PP) are used to generate a list of documents, whereas the remaining constituents are only used to add weights to those documents. This technique looks for limiting the number of documents given back by the query, with the purpose of reducing the process time, with no significant variations of the accuracy [14, 15, 25].

$$sim(Q,D) = \left(\frac{\sum_i q_i \bullet d_i \bullet NLPfactor_i}{|D|} \right) \bullet proximity(Q,D) \tag{1}$$

Our modification:

$$NLPfactor = \log_e(1 + depth) \bullet MOD^{ka} \tag{2}$$

The parameter *proximity* in (1), it is used to catch the proximity between the entities and to penalize the documents that have the same entities but more dispersed. It is presented in (3), in which the average distance between constituents (in number of sentences), and the first and the last sentence in which a query entity appears in the document are used.

$$proximity = \left(1 - slope \bullet \frac{avgDistance}{lastSentence - firstSentence + 1} \right), slope = 0.3 \tag{3}$$

The parameter *NLPfactor* in (2) uses the knowledge obtained from the NLP techniques. It uses several parameters such as *depth* that is obtained from the level in the syntactic tree of the query. For example, the query *architecture in Berlin*, the *depth* of the whole NP (the head *architecture*) is 1, whereas the *depth* of the nested NP (*Berlin*) is 2. This is used because the NP with a larger depth restricts the searching more than the NP with a lower depth. That is to say, documents about *architecture* in general are less relevant than those about the *architecture* developed in *Berlin*. In our model, this means that the entry *Berlin [ ]* is valued higher than *architecture [Berlin]*. The *depth* value is normalized by means of the logarithm, although it is rarely higher than 3.

The parameter *MOD* in equation (2) corresponds with the comparison between the lists of modifiers of the query and the documents that share the same head.

We have found five possible cases to value MOD:

1. First when both, the query and the documents, do not have modifiers, e.g. when the user asks for *architecture*, and in the document a NP appears with the head *architecture* and with no modifiers. The second entry of case 1 corresponds to a document that contains a NP with the head *architecture* with more modifiers: *architectural transformation (...) architecture of Alex Schult*. This document is valued higher than the first one because the second document is presenting additional information about the general query concept, whereas the first document does not go more deeply into the *architecture* topic.
2. This case corresponds to the query *Berlin architecture* and a document with single apparitions of the NP *architecture*. In this case, the document is the least valued, which is correct because the user is specifying by means of the modifier *Berlin*, whereas the document is about general architecture, and not about the specified architecture.
3. This case corresponds to the query *Berlin architecture* but the document's modifiers add more information to the specified query entity (*a new architectural vocabulary for Berlin (...) monumental architecture (...) official architecture of Berlin*), which is the optimal situation.
4. This case valuates when some query modifiers (*new architecture of west Berlin*) do not appear in the document entity (*author's architecture (...) architecture of west Berlin*).
5. Finally, the case 5 is similar to cases 3 and 4, but shows that we are not sure that the document contains the searched entity, due to the presence of different modifiers in the query (*new architecture of west Berlin*) and document (*new architecture for east Berlin (...) monumental architecture*).

We generate two additional lists for the NPT table query: one with the modifiers in appositions, relative clauses and prepositional phrases, and another one with the remaining modifiers. This is because we intend to distinguish between the semantic knowledge of each kind of modifier.

After obtaining three final values of document relevance (one for each table: NPT, PPT and CCT), it is necessary to merge them into a unique value that represents the relevance that our system assigns to a document. To each table an importance factor is associated, that is experimentally calculated in the training phase. We have also tested some methods, including the ones exposed by in Bartell et al. [5] and by Voorhees et al. [22], without managing to improve the results with respect to a simple weighted sum.

4 Evaluation

4.1 Base Model Evaluation

The formula in (1) has been used for both languages (English/Spanish), and for the long(title + description + narrative) and short(title + description) version of the queries, in order to prove the applicability of the model to different kinds of questions and different languages. In both cases, parameters (*NLPfactor* and *proximity*) and their coefficients have been experimentally obtained in a training phase, and after that they have been used in a test phase.

For English language, the CLEF 2000 and 2002 English questions (from 1 to 40 and from 91 to 140) were used for the training. The CLEF 2001 questions were used (from 41 to 90) for the test[3]. The corpus used is the collection of the 113,005 news of the newspaper Los Angeles Times of the year 1994.

For Spanish language, the CLEF 2002 (from 41 to 140) questions were used for the training and the 2003 questions (from 141 to 200) were used for the test. The corpus used is the collection of the 454,045 news of the EFE of the years 1994 and 1995.

The test results are shown in Table 3, they report an improvement for English(CLEF 2000/2002 queries 41 to 90) of 35.11% in the average precision for short queries; and 12.96% for long queries with reference to the cosine baseline. For Spanish(CLEF 2003 queries 141 to 200) is 27.42% in the average precision (short queries); and 37.18% (long queries) with reference to the cosine baseline. The corpus used for english languge evaluation consist on Los Angeles Times news of the 1994 and EFE news of 1994 and 1995 for Spanish.

Language		English				Spanish			
Experiment	Q	AvgP 11-p	Precision at 5 docs	R-Precision	Recall	AvgP 11-p	Precision at 5 docs	R-Precision	Recall
Cosine	S	0.3506	0.4120	0.3301	0,9498	0.2852	0.3600	0.2963	0,8184
	L	0.4597	0.4640	0.4613	0,9556	0.3045	0.4100	0.2992	0,7965
Our Proposal	S	+35.11%	+16.72%	+35.81%	+0.91%	+27.42%	+36.11%	+22.65%	+0.90%
	L	+12.96%	+13.73%	+7.13%	+0.91%	+37.18%	+29.27%	+31.08%	+0.90%

Table 2. Results obtained in the English/Spanish test phase

Once the viability and efficacy of our base model has been demonstrated, we can complement it using the optional modules.

[3] The distribution between test and training queries has been done because we had referential results in CLEF 2001 from Llopis and Vicedo (2002).

4.2 Evaluation of Optional Modules

4.2.1 Query Expansion We have used two techniques to query expansion as was mentioned in section 3.2. The first technique uses WordNet. The second expansion technique is a variation of the windowing technique proposed by Amo et al., in which we use the first 5 relevant documents to modify the similarity value of the remaining documents[2].

WordNet is commonly used in IR because it permits matching relevant documents that could not contain any query terms[18].

Language		English			Spanish		
Experiment	Q	AvgP 11-p	Precision at 5 docs	R-Precision	AvgP 11-p	Precision at 5 docs	R-Precision
Cosine	S	0.3506	0.4120	0.3301	0.2852	0.3600	0.2963
	L	0.4597	0.4640	0.4613	0.3045	0.4100	0.2992
Cosine with query expansion and windowing	S	+20.84%	+1.94%	+24.50%	+4.63%	+0.92%	-4.39%
	L	+4.56%	-1.72%	+0.99%	+8.97%	+2.44%	+9.96%
Our Base Proposal	S	+35.11%	+16.72%	+35.81%	+27.42	+36.11%	+22.65%
	L	+12.96%	+13.73%	+7.13%	+37.18	+29.27%	+31.08%
Our Proposal with query expansion and windowing	S	+40.59%	+17.48%	+47.74%	+44.95%	+42.58%	+32.33%
	L	+13.88%	+10.34%	+8.39%	+39.17%	+26.82%	+33.75%

Table 3. Results obtained in the English/Spanish test phase using query expansion

As shown in Table 3, when query expansion was applied we have obtained better average precision rates than the baseline ones. We have got values of 40.59% in short queries and 13.88% in long queries for English. For Spanish improves were 44.95% and 39.17%, for short and long queries respectively. Also, the application of query expansion have got an increase in short and long queries of 5.48% and 0.92% for English and 17.53% and 1.99% for Spanish, against the base model results.

4.2.2 Complementing Index with WordNet. We propose a WNT table as index complement to identify semantically related terms.

The table implementation can be yield as NPT table copy, changing the word representation by its synset representation. Two implementations have been done for this table:

1. Using synsets for terms in modifiers list and word representation for entity head.
2. Using synsets for entity head and word representation for terms in modifiers list.

For this phase we have not apply disambiguation, because its use in word sense produce only minor effects on retrieval accuracy, as Sanderson shown, apparently confirming that query/document matching strategies already perform an implicit one. In addition if explicit word sense disambiguation is performed with less than 90% of accuracy, the results are worse than non disambiguating at all[16].

For this approach, we have taken the EuroWordNet 1.0 synsets. The first experiment replaced the only one sense terms; the second one was for the terms with three senses maximum (without disambiguation.). The modifiers list had been expanded like a cartesian product of the respective senses of each term, processing one by one, creating all the possible combinations.

The results of first experiment for Spanish are shown in Table 4. The application of WNT table in modifiers list improves the precision over our baseline, and also improves our basic model results.

Experiment	Q	AvgP 11-p	Precision at 5 docs	R-Precision
Cosine	S	0.2852	0.3600	0.2963
Our base approach	S	+27.42%	+37.40%	+22,65%
Only one sense	S	+44.81%	+53,03%	+32.87%
Three or less senses	S	+43.51%	+43,53%	+26.12%

Table 4. Using WNT with modifiers list in Spanish test phase over short queries.

The performance of second representation is worse than using the modifiers synsets. The precision drops drastically(Table 5). It may be the result of ambiguity in entity head. The more senses the worse system performance.

Experiment	Q	AvgP 11-p	Precision at 5 docs	R-Precision
Cosine	S	0.2852	0.3600	0.2963
Our base approach	S	+27.42%	+37.40%	+22,65%
One sense	S	+17.99%	+27.67%%	+12.82%
Two senses	S	+17.99%	+27.67%	+12.82%
Three senses	S	+7.99%	+20.86%	+3.81%
Four senses	S	+0.70%	+8.19%	-2.16%
Five senses	S	-2.45%	+4.28%	-4.12%

Table 5. Using WNT with heads table in Spanish short queries test phase.

5 Conclusions and Future Work

Our model improves precision over traditional bag of words approaches (the assumption that the terms occur independently from the others, that is not true) by merging NLP knowledge into the same model: morphological, syntactical, anaphora resolution and WordNet.

This model improves other proposals by means of not indexing just independent head-modifier pairs, but phrases, in order to consider relations between different modifiers.

This proposal has been used in the vector space model. Three modifications have been done. The first one stores entities instead of terms. The second one introduces two additional parameters in the measure of similarity between the query and document vectors: *NLPfactor* and *proximity*. And the third one stores WordNet synsets or terms instead of their word representation. As NLP tools, we have used a POS tagger, a partial parser and an anaphora resolver and WordNet as a NLP resource.

We have evaluated the applicability of our model on two languages (Spanish and English), on two different versions of a query (long and short).

The success of the model has been proven by means of high increases in the average precision with regard to the vector space model. In a first phase we test the base model, and then we test the optional modules. The results for base model English short queries have an increase of 35.11% in the average precision, and the long queries 12.96%. The Spanish short queries have a 27.42% increase and the long queries 37.18%. Similar percentages have been obtained on the usage of optional modules,. the query expansion module reports for English 40.59% and 13.88%; for Spanish 44.95% and 39.17%. Using WordNet index module we have obtained for Spanish 44.81% and 27.42%. These increases are much higher than those obtained in other previous works.

Our future research will be focused on improving obtained results with incorporation of WordNet by other experiments, and the usage of disambiguation. We hope that the use of word sense disambiguation for index construction could improve precision, like experiments reports by Gonzalo et al [10]. In addtion, we will implement other similarity measures, like Pivoted Cosine or Okapi.

Acknowledgments

The corresponding author wants to thank the Ministerio de Ciencia y Tecnología-CONICIT for the partial support giving to this project.

References

1. Alonso, M. A., Vilares, J., Darriba, V. M. (2002) On the Usefulness of Extracting Syntactic Dependencies for Text Indexing. *Artificial Intelligence and Cognitive Science*. Volume 2464 of Lecture Notes in Artificial Intelligence, pp. 3-11.
2. Amo, P., Ferreras, F.L., Cruz, F., Rosa, M. (2000). Smoothing Functions for Automatic Relevance Feedback in Information Retrieval. In the Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA'00), pp. 115-119.
3. Arampatzis, A. T., van der Weide, Th. P., Koster, C. H. A., and van Bommel, P. (2000). Linguistically motivated Information Retrieval. Encyclopedia of Library and Information Science. Marcel Dekker, Inc., New York, Basel.
4. Baeza-Yates, R. (2004) Challenges in the Interaction of Information Retrieval and Natural Language Processing. *Computational Linguistics and Intelligent Text Processing*. Volume 2945 of Lecture Notes in Computer Science, pp. 445-456.
5. Bartell, B., Cottrell, G., Belew, R. (1994). Automatic combination of multiple ranked retrieval systems. In the Proceedings of the 17th International Conference on Research and Development in Information Retrieval (SIGIR'94), pp. 173-181.
6. Byung-Kwan, K., Jee-Hyub, K., Geunbae, L., Jung Yun, S.(2000). Corpus-Based Learning of Compound Noun Indexing. In the Proceedings of the ACL 2000 Workshop on Recent Advances in NLP and IR, pp. 57-66.
7. Cornelis H.A. Koster. (2004) Head/Modifier Frames for Information Retrieval. *Computational Linguistics and Intelligent Text Processing*. Volume 2945 of Lecture Notes in Computer Science, pp. 420-433.
8. Ferrández, A., Palomar, M., Moreno, L. (1999). An empirical approach to Spanish anaphora resolution. *Machine Translation*, 14(3/4), pp. 191-216.

9. Gonzalo, J.,Verdejo, F., Chugur, I., Cigarrán, J. (1998) Indexing with WordNet synsets can improve text retrieval. In the Proceedings of the ACL/COLING Workshop on Usage of WordNet for Natural Language Processing, pp. 38-44.
10. Gonzalo J., Verdejo F., and Chugur I. (1999). Using EuroWordNet in a concept-basedapproach to cross-language text retrieval. *Applied Artificial Intelligence*, 13(7):647--678.
11. Kaszkiel, M., Zobel, J., Sacks-Davis, R. (1999). Efficient passage ranking for document databases. *ACM Transactions of Information Systems*, 17(4), pp. 406-439.
12. Llopis, F., Vicedo, J.L. (2002). IR-n: A Passage Retrieval System at CLEF-2001. *Evaluation of Cross-Language Information Retrieval Systems.* Volume 2406 of Lecture Notes in Computer Science, pp. 244-252.
13. Mitra M., Buckley C., Singhal A., Cardie C. (1997). An analysis of statistical and syntactic phrases. In the Proceedings of the 5th International Conference "Recherche d'Information Assistee par Ordinateur" (RIAO'97), pp. 200-214.
14. Moffat, A., Zobel, J. (1996). Self-indexing inverted files for fast text retrieval. *ACM Transactions on Information Systems*, 14(4), pp. 349-379.
15. Persing, M., Zobel, J. (1996). Filtered document retrieval with frequency-sort indexes. *Journal of the American Society of Information Science*, 47(10), pp. 749-764.
16. Sanderson, M.(1994). Word Sense Disambiguation and Information Retrieval. In the Proceedings of 17th International Conference on Research and Development in Information Retrieval.
17. Singhal, A., Buckley, C., Mitra, M. (1996). Pivoted Document Length Normalization. In the Proceedings of the 19th International Conference on Research and Development in Information Retrieval (SIGIR'96), pp. 21-29.
18. Smeaton, A., Kelledy, F., O'Donnell. (1995). TREC-4 experiments at Dublin City University: Thresholding posting lists, Query Expansion with WordNet and POS Tagging of Spanish. In Proceedings of TREC-4.
19. Strzalkowski, T. (1999). Natural Language Information Retrieval. Kluwer Academic Publishers.
20. Strzalkowski, T. , Fang Lin, Jin Wang, Jose Perez-Carballo (1999). Evaluating Natural Language Processing Techniques in Information Retrieval. In (Strzalkowski, 1999a), pp. 113-146.
21. Vilares, J., Alonso, M.A., Ribadas, F.J. (2003). COLE Experiments at CLEF 2003 Spanish Monolingual Track. In the Proceedings of the Working Notes for the CLEF 2003 Workshop, pp. 197-206.
22. Voorhees, E., Gupta, N., Johnson-Laird, B. (1995). The collection Fusion Problem. In the Proceedings of the Third Text Retrieval Conference (TREC-3), pp. 95-104.
23. Voorhees (1999). Natural Language Processing and Information Retrieval. *Information Extraction: towards scalable, adaptable systems.* Volume 1714 of Lecture Notes in Artificial Intelligence, pp. 32-48.
24. Zhai, Ch., Tong X., Milic-Frayling N., A. Evans, D. (1997). Evaluation of Syntactic Phrase Indexing - CLARIT NLP Track Report. In the Proceedings of the Fifth Text REtrieval Conference (TREC-5).
25. Zobel J., Moffat A (1998). Exploring the similarity space. In the Proceedings of the 21st International Conference on Research and Development in Information Retrieval (SIGIR'98), pp. 18-34.