The Successful Application of Natural Language Processing for Information Retrieval

Antonio Ferrández; Yenory Rojas; Jesús Peral
Dept. Languages and Information Systems
University of Alicante
Carretera San Vicente S/N - Alicante - Spain - 03080
+34-96-590-3400
antonio@dlsi.ua.es yrojas@dlsi.ua.es jperal@dlsi.ua.es

ABSTRACT

In this paper, a novel model for monolingual Information Retrieval in English and Spanish language is proposed. This model uses Natural Language Processing techniques (a POS-tagger, a Partial Parser, and an Anaphora Resolver) in order to improve the precision of traditional IR systems, by means of indexing the "entities" and the "relations" between these entities in the documents. This model is evaluated on both the Spanish and English CLEF corpora. For the English queries, there is a maximum increase of 35.11% in the average precision. For the Spanish queries, the maximum increase is 37.18%.

Keywords: Information Retrieval, Natural Language Processing, Entity, CLEF, anaphora resolution.

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 been reported to show no significant improvement in retrieval performance, although it seems that they may overcome the inadequacies of purely quantitative methods of text IR, i.e. statistical full-text retrieval or bag of words representations. As examples of the attempts to overcome these inadequacies, the works from Strzalkowski (1999a) or Baeza-Yates (2004) can be read. 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. Of course, the relatively low quality of parsing may be a major problem, although there is little evidence to support that. Voorhees (1999) claims that the lack of good weighting techniques for compound terms is an important factor that affects NLP compared to current IR techniques.

In this paper, we propose a novel IR model that incorporates NLP techniques such as POS-tagging and partial parsing to improve the traditional bag of words representations. This model indexes entities and the relations between these entities. These relations are based on the clause splitting of the document, and the resolution of anaphora phenomenon between these entities. Our proposal improves other approaches that use NLP knowledge for IR, because it merges more knowledge than other proposals (morphological, syntactical and anaphora resolution), and this knowledge is successfully exploited (increases up to 37.18% are obtained) through the vector space model indexing compound terms in an effective way. Moreover, this model runs in a very computational efficient way.

In the following section, the antecedents of the incorporation of NLP for IR tasks are presented. Later, the model proposed in this paper is presented in its intuitive view, and in its inclusion in the vector space model. It is finally evaluated on the English and Spanish CLEF[1] corpora and it is compared with several measures of similarity.

2. ANTECEDENTS OF NLP IN IR

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 vector model and the cosine measure). In the set of *Text REtrieval Conferences*[2] *(TREC)* different statistical approaches can be found.

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 with regard to the computational effort that supposes the utilization of this kind of knowledge.

The IR systems that use NLP can be classified according to the kind of NLP knowledge they use. For example, some of them use morphological knowledge to use the lemma instead of the stem of the words, as well as several morphologic derivations, e.g. Vilares et al. (2003).

Other systems use query expansion techniques by means of adding new terms obtained from synonyms gathered from WordNet, e.g. Gonzalo et al. (1998) or Arampatzis et al. (2000), 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. The 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 measure of similarity 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. Zhai et al. 1997, Mitra et al. 1997, Strzalkowski et al. 1999b) or phrases (e.g. Arampatzis et al. 2000). With regard to the pair and ternary expressions, these systems usually

[1] The Cross Language Evaluation Forum. http://www.clef-campaign.org/
[2] http://trec.nist.gov

index the head of the constituents (mainly noun and verbal phrases) jointly with their modifiers. For example, Byung-Kwan et al. (2000) index just Korean compound nouns with only a 0.84% of improvement in the average precision. With respect to the phrases, they have to devise complex measures of similarity between syntactic trees.

Some systems try to mix several kinds of knowledge, even jointly with the vector model, as occurs in Cornelis (2004). Another example is Strzalkowski et al. (1999b), where they use head-modifier pairs to create a new indicator. Along with stems of the words, and other streams of data, they are able to improve by 7% the average precision in short questions (with few words) and 20% in long questions (more descriptive), with respect to a vector system base only with stems. Nevertheless, the most important component of the system continues to be vector model with stems, where the streams of pairs are used in a secondary form. Another similar work is Alonso et al. (2002), in which the authors combine stems, lemmas and derivation, jointly with head-modifier pairs on Spanish CLEF corpora with only 1.59% improvement.

Our proposal improves these proposals because more NLP knowledge is merged in the same model: morphological, syntactical, and anaphora resolution. Morphological knowledge means using a POS tagger to obtain the lemma of each word as well as their lexical category (proper or common nouns, verbs, etc.). The syntactical one uses a partial parser that performs a deep parsing of the constituents that we consider important for extracting the concepts (noun phrases, relative clauses, appositions, prepositional phrases, etc.) and relations of these concepts (clause segmentation). The anaphora resolution is carried out both on definite descriptions and pronoun resolution. Most of this knowledge has not been used in previous proposals. Furthermore, our proposal obtains successful results with increases as high as 37.18% in the average precision, whereas the other proposals usually obtain around 3%. This is because we model all this knowledge in a different way, since we do not index just head-modifier pairs, but phrases (noun and prepositional phrases and clauses). In this way, we also consider to be very important the relation between different modifiers, and we capture more information by means of resolving pronominal anaphora and definite descriptions. In addition, the phrase indexation model allows a high normalization of different syntactic tree structures into the same structure. Finally, our model is run in a computationally efficient way that allows its implementation in real applications of IR.

3. THE PROPOSED MODEL

Our model is based on the intuitive idea that a document should be represented by means of its "entities" and the "relations between its entities". Since this model is mainly based on syntactic knowledge, the entities are represented by means of noun phrases (NP), whereas the relations between them are represented by means of the clauses, in which the verb is the head and its modifiers are the NP and prepositional phrases (PP). These relations are completed by means of resolving anaphora. These facts can be explained by means of the example (1) where there are two entities: *Mary Blake* and *Mary Spencer*, and from the syntactic knowledge, we obtain additional information about the second one (*the secretary of ARS*). Since the anaphoric reference between *her* and *Mary Blake* is resolved, more information about this entity is obtained (*Mary Blake* is *the president of ISS*).

(1) Mary Blake arrived late, so Mary Spencer who is the secretary of ARS fined her, the president of ISS, with 1000€.

In this way, we can successfully resolve a user's query demanding information about *Mary Blake, the president of ISS*, and it can be discarded for other queries, e.g. about *Mary Blake, the president of ARS*.

4. THE INCLUSION OF OUR MODEL IN THE VECTOR SPACE MODEL

In this section, the intuitive model is implemented in a traditional statistical or bag of words IR representation, in order to overcome the main problem of statistical methods, i.e. the assumption that the terms occur independently from the others, which is not true. This problem is overcome by transforming the terms into entities, and by introducing NLP knowledge.

Specifically, the statistical IR method to use is the vector space model, in which queries and documents are represented as vectors in an n-dimensional space, where *n* is the number of indexing terms, and then they are compared by applying a measure of similarity such as the cosine of the angle between the query and document vectors. Such quantitative measure allows the ranking of the retrieved documents.

In the first subsection, the NLP tools used for the implementation are briefly described. In the following subsection, the modifications to the vector model needed to transform the terms into entities are introduced. Finally, the modifications introduced in the measure of similarity are explained.

4.1. The NLP tools

As the selected tool to obtain the knowledge needed in the intuitive model, we have worked on the output of the computational system called *Slot Unification Parser for Anaphora Resolution (SUPAR)*. This system, which was presented in Ferrández et al. (1999), resolves anaphora in both English and Spanish texts, although it can be easily extended to other languages[3].

SUPAR works on the output of a POS tagger (e.g. for English, the TreeTagger[4] is used, and for Spanish the Maco[5] is used), and partial parses the text. SUPAR partial parses coordinated NP, coordinated PP, verbal phrases and conjunctions, where NP can include relative clauses, appositions, coordinated PP and coordinated adjectives. Conjunctions are used to split sentences into clauses. In this

[3] The SUPAR system can be tested in http://supar.dlsi.ua.es/supar/. It resolves English pronominal anaphora with 74% of success, and Spanish pronominal anaphora with 81%.

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[5] http://nipadio.lsi.upc.es/cgi-bin/demo/demo.pl

way, we select the NP as the entities of the document, and the clauses as the relations between these entities. An example of the parsing process and the detection of noun phrase entities in a sentence can be observed in (2), where 10 entities have been extracted.

(2) **[**[David R. Marples's**]$_1$** new book, his second on **[**the Chernobyl accident of **[**April 26, 1986**]$_2$]$_3$]$_4$**, is **[**a shining example of **[**the best type of **[**non-Soviet analysis into **[**topics**]$_5$]$_6$]$_7$]$_8$** that only recently were **[**absolutely taboo in **[**Moscow official circles**]$_9$]$_{10}$**.

4.2. The transformation of the vector terms into entities

In order to implement the entities and the relations between them in the vector model, they are represented in three tables: NPT, PPT and CCT. The NPT stores information about the entities syntactically represented as noun phrases (NP). The two remaining tables store additional information about these entities or relations between them, in the form of prepositional phrases (PPT) and clauses (CCT). The PPT represents some specific knowledge about the entities that one obtains from the query. For example, in the query *architecture in San Louis*, the preposition *in* means that *San Louis* may be a place entity instead of a person entity. Therefore, a document in which appears a PP *in San Louis* is valued higher than other documents in which *San Louis* does not appear with that preposition. Finally, the CCT stores the verb of a clause and all its entities. For example, in the second clause of (1), the CCT stores the verb *fine* jointly with the NP *Mary Spencer* and *Mary Blake* and the *1000€ fine*. In this way, some extra information is stored in the model that is not present in the traditional vector representation, e.g. the prepositions and pronouns (which belong to the list of stop-words[6]), as well as the information of each entity and the relations between them.

In order to store the entities in a computationally efficient way, each table stores the head of the constituent (which is used to search for the constituent) and a list of entity modifiers (which are used to fine-tune the searching). For example, in (1) the following entry is stored in NPT: *Mary [[Blake, president, ISS], [Spencer, secretary, ARS]]*, which means that there are two entities with the same head (*Mary*), with all the information obtained from each one (*Mary Blake the president of ISS* and *Mary Spencer the secretary of ARS*). In this way, complex NP are represented as structures composed of phrase heads and modifiers. The structures attempt to preserve the original logic of the phrase, unlike in earlier proposals where these were broken up into independent head-modifiers pairs.

For each entry of 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, as well as additional information that is explained in the following subsections.

The NPT table stores each NP in the text. For example, *a convertible car* is stored as *car [convertible]*, i.e. *car* as the head and *convertible* as its modifier. However, the rotation of this entry is also stored as *convertible [car]* in order to solve references to this entity such as *the convertible*. It also occurs with people, e.g. *John Fitzgerald Kennedy* is stored as *Kennedy [John, Fitzgerald], Fitzgerald [John, Kennedy]* and *Kennedy [John, Fitzgerald]*, which can catch references such as *John* or *Kennedy*. In cases where the semantic value depends on the order, e.g. *junior college* versus *college junior*, our system distinguishes between both entities by means of a set of penalties according to the lexical type of the head of the constituent. These penalties are obtained in the training phase of the system, and they are presented in full detail in section 4.3. For example, a proper noun is valued as 1.4, a common noun is valued as 1.0, whereas an adjective is valued as 0.9. It means that a query that asks for *junior college*, and a document contains *college junior* it is valued lower than another that contains *junior college*, because the rotation of the first one is penalized by 0.9 in its rotated entry $college_{adjective}$ *[junior]*. These rotations are not applied on the relative clauses or PP that can appear in a NP.

It should be mentioned that each head or modifier is stored as the stem of the lemma (e.g. for the word *escaped*, the lemma is *escape*, and its stem *escap*). This gave the best results in the evaluation phase in comparison with the results of using the lemma or the stem separately.

In this way, our model overcomes a traditional drawback of approaches that use NLP knowledge for IR: it can easily normalize different syntactic-tree-structures into the same entity or concept. For example, *Spain mountains reforestation, reforestation of Spain mountains, reforestation of mountains of Spain, reforestation of mountains that are from Spain, reforestation of mountains that are in Spain* and *reforestation of Spain mountains*, are conflated in the entity *reforestation [Spain, mountains]*.

With regard to the anaphora resolution process, we consider that we find a reference to a previously named entity when there is an inclusion relation between the lists of modifiers of both NP, otherwise we have found a new entity and a new list of modifiers is stored in the table. For example, in the first clause in (1), we store the entity *Mary [Blake]*. When the NP *Mary Spencer who is the secretary of ARS* appears in the text, both entities share the head, *Mary*, but they do not share any modifier. Therefore, two entities are stored in the table as *Mary [[Blake], [Spencer, secretary, ARS]]*. Finally, when the NP *her, the president of ISS* appears and the pronoun is resolved as *Mary Blake*, then the NP stays as *Mary [Blake, president, ISS]*, and since *Mary [Blake]* is included in it, then we determine that it is referring to the same entity, so the new modifiers are added to the list. Finally, the entry in the table stays as *Mary [[Blake, president, ISS], [Spencer, secretary, ARS]]*.

The PPT and CCT tables work in a similar way to the NPT table, but the PPT table stores the preposition as well as the head of the NP, whereas the CCT table stores the verb and all the content words in the clause.

4.3. Comparing the query and the document vector

In our model, the vector terms have been transformed into entities for both the query and the documents. After that, a well-known measure of similarity of the vector space model (the cosine of the angle between the query and document

[6] Words considered having no indexing value, which are removed from text.

vectors) is modified in order to use the NLP knowledge of our model.

The three tables (NPT, PPT and CCT) in the documents and in the query are separately compared with the cosine as it appears in Eq. [1] (Kaszkiel et al. 1999), with two differences. The first difference is that the weights are multiplied by two parameters: *NLPfactor* (knowledge obtained from the NLP techniques) and *proximity* (it measures the proximity between the query entities in the document), as it is presented in equation Eq. [2]. The second difference is that the first two query syntactic constituents (i.e. NP or PP) 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 (Moffat and Zobel 1996, Persin and Zobel 1996, Zobel and Moffat 1998).

[1] Kaszkiel cosine formula:

$$sim(Q,D) = \left(\frac{\sum_i q_i * d_i}{\|D\|} \right) \qquad q_i = \log_e(tf_{q,i} + 1) * \log_e(\frac{N}{df_i} + 1)$$

$$d_i = \log_e(tf_{d,i} + 1)$$

[2] Our modification:

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

[3] $$NLPfactor = \log_e(1 + depth) * MOD^{lex}$$

The parameter *NLPfactor* uses the knowledge obtained from the NLP techniques. It is presented in Eq. [3] and 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.

C	Description	MOD_{NPT}
1	|LModifQuery| = 0	1.7+ 0.4 * $\log_e$(1+|LModifDB|)
2	|LModifDB|= 0	0.6
3	$\exists i$ / LModifQuery $\subset$ $LModifDB_i$ |LModifQuery| ≠ 0 AND | LModifDB| ≠ 0	2.0 + 0.3 * R * $\log_e$(Common+1)
4	$\exists i$ / $LModifDB_i \subset$ LModifQuery |LModifQuery| ≠ 0 AND | LModifDB| ≠ 0	1.4 + 0.9 * R * $\log_e$(Common+1)
5	$\forall i$ / (LModifQuery $\not\subset$ $LModifDB_i$ AND $LModifDB_i \not\subset$ LModifQuery) |LModifQuery| ≠ 0 AND | LModifDB| ≠ 0	0.8 + 0.9 * R * $\log_e$(Common+1)

Table 1. Description of parameter *MOD* for the NPT table

With reference to the parameter *MOD* in equation [3], it corresponds with the comparison between the lists of modifiers of the query and the documents that share the same head. It is summarized in the five cases (C) in Table 1 for the NPT table (MOD_{NPT}), where the operation |…| corresponds to the cardinality of a list, *LModifQuery* is the list of modifiers of the query, *LModifDB* is the list of entities with the same head as the query, $LModifDB_i$ is the list of modifiers of the entity number *i* stored in the NPT table, *Common* is the maximum number of modifiers that are repeated in both lists and *R* is the number of lists that have the *Common* shared modifiers. The coefficients in the formulas vary according to the language (English/Spanish) and type of query (long/short), and the coefficients in Table 1 have been experimentally obtained in the training phase for the short English queries.

C	LModifQuer	LModifDB	Variables	MOD_{NPT}
1	[ ]	[ ]	|LModifDB|=0 Common=0 R=0	1.7
1	[ ]	[transform] [axel,schult]	|LModifDB|=2 Common=0 R=0	2.139
2	[berlin]	[ ]	|LModifDB|=0 Common=0 R=0	0.6
3	[berlin]	[new,vocabulari,berlin] [monument] [offici,berlin]	|LModifDB|=3 Common=1 R=2	2.415
4	[new, west, berlin]	[author] [berlin, west]	|LModifDB|=2 Common=2 R=1	2.388
5	[new, west, berlin]	[new,east,berlin] [monument]	|LModifDB|=2 Common=2 R=1	1.788

Table 2. Some examples of the calculation of MOD_{NPT}.

In Table 2, some explanatory examples are presented for each case (C) of Table 1. The first entry of Table 2 corresponds to the case 1 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 (MOD_{NPT} is 2.139 compared to 1.7) 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. The example of case 2 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 (MOD_{NP} takes 0.6), which is correct since the user is specifying by means of the modifier *Berlin*, whereas the document is about general architecture, and not about the specified architecture. The case 3 corresponds to the same query 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, and therefore it obtains the maximum MOD_{NPT} (2.415). The case 4 also takes a high MOD_{NPT}

(2.388), but it is lower than case 3 because there are some query modifiers (*new architecture of west Berlin*) that do not appear in the document entity (*author's architecture (...) architecture of west Berlin*), which is correct. 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*).

The five cases (C) in Table 1 have also been experimentally obtained in the training phase for the long English queries and (long/short) Spanish queries, with similar coefficients and behaviours.

We generate two additional lists for the NPT query: one with the modifiers in appositions, relative clauses and prepositional phrases (*list3*), and another one with the remaining modifiers (*list2*). This is because we intend to distinguish between the semantic knowledge of each kind of modifier. In this way, the parameter MOD_{NPT} takes the value in Eq. [4]. This division, whose percentages have been experimentally obtained, is mainly justified due to the high rate of errors produced in parsing by the attachment of appositions, relative clauses and prepositional phrases. In this way, the modifiers obtained from *list3* will be valued less than those obtained from *list2*.

[4] $MOD_{NPT} = 0.7 * MOD_{list2} + 0.3 * MOD_{list3}$

The parameter *lex* in Eq. [3] depends on the lexical type returned by the POS tagger for each head of constituent. As an example in the English version, if the head of the constituent has been tagged as a proper noun then *lex* takes the value of 1.4, if it is a common noun it takes 1.0, if it is an adjective it takes 0.9, if it is a verb it takes 0.7 and the remaining lexical tags take 0.3. The values taken by this parameter are quite natural, because the proper nouns are the most valued (the most prominent items in Information Retrieval). The same coefficients have also been experimentally obtained in the training phase for the Spanish queries, with a difference that should be mentioned. In Spanish queries the common nouns are valued less than adjectives. This fact is justified by the errors of the POS tagger, specifically in the tagging of nouns, adjectives and verbs. As previously mentioned, this parameter allows us to perform the entity rotations in an optimal way, in order to capture different references to the same entity.

The $NLPfactor_{PPT}$ is obtained in the same way to the $NLPfactor_{NPT}$ but it is different for the CCT table because it gave very bad results during the training phase. After analysing the results, we conclude that there were few coincidences between the query and document verbs, as well as in the CLEF queries there were a small number of verbs. Therefore, we decided to change the formula for the $NLPfactor_{CCT}$ as shown in Eq. [5], in order to return a higher value to the few coincidences between the query and the document. In this formula, *common* is the number of coincident modifiers, and *maxCommon* is the maximum length of any list of modifiers of the constituent.

[5] $NLPfactor_{CCT} = 10^{common^{\max Common}}$

With regard to the parameter *proximity* in Eq. [2], 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 Eq. [6], 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.

[6] $proximity = \left(1 - slope * \frac{averageDis\tan ce}{lastSentence - firstSentence + 1}\right), slope = 0.3$

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. We have tested diverse methods, including the ones exposed in Bartell et al. (1994) and Voorhess et al. (1995), without managing to improve the results with respect to a simple weighted sum. To each table an importance factor is associated, that is experimentally calculated in the training phase. The empirical results reveal a great importance of NPT in both languages, for English: 85% (NPT), 5% (PPT) and 10% (CCT), for Spanish: 75% (NPT), 12% (PPT) and 13% (CCT).

5. EVALUATION

We have performed several experiments in two different languages (English and Spanish) in order to test our proposal. The same formula in [2] has been used for both languages, and for the long and short 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[7]. 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.

All the questions are in two versions: short and long, e.g. the question in (3) has three fields: *title, desc, narr*, where its long version uses the three fields and its short version only uses the *title* and *desc* fields.

(3) **<title>** Area of Kaliningrad
<desc> Find documents discussing the political or economic future of the Kaliningrad Exclave.
<narr> Only political or economic information on Kaliningrad is of interest. Prospects for future relations

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

with Scandinavia, the Baltic countries and Russia. Historical or tourist information is not important.

In the following tables, the obtained results are presented. Table 3 presents the test results for English; whereas Table 4 presents the test results for Spanish. The second column means the type of queries (S: short, L: long), the remaining columns mean different measures obtained from the *trec_eval* package with the relevance judgment created by means of the pooling technique from the CLEF participants. The first row shows the results of our baseline: the cosine baseline with stemmed terms and using the formula in [1] (Kaszkiel et al. 1999). With regard to the training results, a final improvement of 14.60% in the average precision in English language to short queries and an improvement of 5.59% in long queries are obtained (in comparison with the cosine baseline). In Table 3 the test results are shown: an improvement of 35.11% in the average precision for short queries; and 12.96% for long queries with reference to the cosine baseline.

Experiment	Q	AvgP 11-p	Precision at 5 docs	R-Precision	Recall
Cosine	S	0.3506	0.4120	0.3301	0,9498
	L	0.4597	0.4640	0.4613	0,9556
Our Proposal	S	+35.11%	+16.72%	+35.81%	+0.91%
	L	+12.96%	+13.73%	+7.13%	+0.91%

Table 3. Results obtained in the English test phase.

Similar results are obtained for Spanish, where in the training phase, an improvement of 22.05% in the average precision in short queries and an improvement of 24.03% in long queries are obtained with reference to the cosine baseline. In Table 4 the test results are shown, in which the improvement is 27.42% in the average precision (short queries); and 37.18% (long queries) with reference to the cosine baseline.

Experiment	Q	AvgP 11-p	Precision at 5 docs	R-Precision	Recall
Cosine	S	0.2852	0.3600	0.2963	0,8184
	L	0.3045	0.4100	0.2992	0,7965
Our Proposal	S	+27.42%	+36.11%	+22.65%	+0.90%
	L	+37.18%	+29.27%	+31.08%	+0.90%

Table 4. Results obtained in the Spanish test phase.

We have also tested the applicability of our proposal to a different similarity formula: the pivoted cosine (Singhal et al. 1996), in which the same *NLPfactor* and *proximity* parameters are used. The same training and test process is carried out. The test results are summed up in Table 5, in which considerable improvements are also obtained with regard to pivoted cosine (for English 21.12% and 9.91%, for Spanish 19.76% and 36.67%). Furthermore, our proposal has been compared by implementing the Deviation From Randomness (DFR) measure of similarity proposed in Amati et al. (2003), which was the first IR system in monolingual Spanish CLEF 2003; an improvement of 5% in the average precision has been obtained when our proposal is used jointly with DFR measure.

Experiment	Q	AvgP 11-p English	AvgP 11-p Spanish
Cosine	C	0.3506	0.2852
	L	0.4597	0.3045
Pivoted Cosine	C	0.4120	0.3725
	L	0.4640	0.3469
Our Proposal with regard to Cosine	C	+35.11% (0.4737)	+27.42% (0.3634)
	L	+12.96% (0.5193)	+37.18% (0.4177)
Our Proposal with regard to the Pivoted Cosine	C	+21.12% (0.4990)	+19.76% (0.4461)
	L	+9.91% (0.5100)	+36.67% (0.4741)

Table 5. Results obtained with pivoted cosine.

Our results compared with earlier works report important improvements in average precision. For example, for English corpora, Strzalkowski et al. (1999b) use head-modifier pairs to create a new indicator, and they improve the average precision in short queries by 7% and in long queries by 20% (against our 35.11% and 12.96%), but the most important component of the system continues to be vector model with stems, where the streams of pairs are used in a secondary form. However, we just use our model and with no combination of the vector model itself at all. With regard to Spanish corpora, Alonso et al. (2002), in which the authors combine stems, lemmas and derivation, jointly with head-modifier, they only obtain a 1.59% improvement, whereas we obtain up to 37.18% improvement.

Regarding the fact that the improvement in English long queries is not as good as for short queries, this suggests that a special processing should be carried out, e.g. a higher understanding of the query in order to deal with negations as occurs in example (4): *"not about…"*. Moreover, prominent NP should be detected, e.g. in the narrative version of query (3) (page 5) the prominent NP is "*political or economic information on Kaliningrad"*, whereas the remaining NPs introduce less relevant information.

(4) Relevant documents will report about the new technology that has permitted the discovery of new galaxies and stars, **not about** satellites or celestial bodies of our own solar system.

To conclude, just a few comments about the computational implementation of our proposal. It has been implemented in PHP language in an efficient way, in order to avoid the computational drawback presented in most of the earlier proposals about the use of NLP in IR. In this way, 75 CLEF Spanish short queries are run on the Spanish corpora previously described (2.10 GB of text) in 7 minutes in a Pentium IV 2.8 GHz, whereas these queries in their long version are run in 16 minutes, which allows the costly training process.

6. CONCLUSION

In this paper, we have proposed a novel model of IR that exhaustively uses NLP. This model overcomes the problems of traditional bag of words approaches (the assumption that the terms occur independently from the others, which is not true), by means of indexing the "entities" and the "relations"

between these entities in the documents. It also overcomes the problems of the use of NLP knowledge for IR, because more NLP knowledge is merged into the same model: morphological, syntactical, and anaphora resolution. Most of this knowledge has not been used in other proposals. Moreover, our model improves other proposals by means of not indexing just independent head-modifier pairs, but phrases, in order to consider relations between different modifiers. Another contribution of this work is that the NLP knowledge is incorporated in the IR model in a computational efficient way, in order to avoid the computational drawback presented in most of the proposals about the use of NLP in IR.

Our proposal has been used in the vector space model, and two modifications have been done. The first one is to store entities instead of terms. The second one is to introduce two additional parameters in the measure of similarity between the query and document vectors: *NLPfactor* and *proximity*. As NLP tools, we have used a POS tagger, a partial parser and an anaphora resolver.

We have evaluated the applicability of our model on two languages (Spanish and English), on two different versions of a query (long and short), on different CLEF corpora, and on different measures of similarity (cosine and pivoted cosine). In this way, the portability and generalization of the model has been proven by means of high increases in the average precision with regard to the vector space model. The 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 pivoted cosine: for English 21.12% and 9.91%; for Spanish 19.76% and 36.67%. These increases are much higher than those obtained in other previous works.

As future projects, the authors continue developing new modules and optimizing the existing ones in order to improve the global system. Although we obtain good results, our system still does not take advantage of all the resources that it arranges. In specific, the tables PPT and CCT contribute to the order of a 2% improvement in both languages, this is the reason why they must be improved for an effective use. Equally, the system must be proven in other languages and other corpora, as well as in tasks of Question Answering.

7. REFERENCES

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.

Amati, G., Carpineto, C., Romano, G. (2003). Comparing weighting models for monolingual Information Retrieval. In the Proceedings of the Working Notes for the CLEF 2003 Workshop, pp. 169-178.

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.

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.

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.

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.

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.

Ferrández, A., Palomar, M., Moreno, L. (1999). An empirical approach to Spanish anaphora resolution. *Machine Translation*, 14(3/4), pp. 191-216.

Gonzalo, J., F. Verdejo, I. Chugur, J. Cigarrán (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.

Kaszkiel, M., Zobel, J., Sacks-Davis, R. (1999). Efficient passage ranking for document databases. *ACM Transactions of Information Systems*, 17(4), pp. 406-439.

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.

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.

Moffat, A., Zobel, J. (1996). Self-indexing inverted files for fast text retrieval. *ACM Transactions on Information Systems*, 14(4), pp. 349-379.

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.

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.

Strzalkowski, T. (1999a). Natural Language Information Retrieval. Kluwer Academic Publishers.

Strzalkowski, T. , Fang Lin, Jin Wang, Jose Perez-Carballo (1999b). Evaluating Natural Language Processing Techniques in Information Retrieval. In (Strzalkowski, 1999a), pp. 113-146.

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.

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.

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.

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).

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.