Aplicación del Procesamiento de Lenguaje Natural en la Recuperación de Información

Yenory Rojas; Antonio Ferrández; Jesús Peral
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
Carretera San Vicente S/N - Alicante - España - 03080
+34-96-590-3400
yrojas@dlsi.ua.es antonio@dlsi.ua.es jperal@dlsi.ua.es

Resumen: En este artículo se presenta un modelo innovador para la Recuperación de Información monolingüe en inglés y español. El modelo usa técnicas de Procesamiento de Lenguaje Natural (un etiquetador de categorías gramaticales –POS tagger–, un analizador sintáctico parcial y un módulo para la resolución de la anáfora) para mejorar la precisión de los sistemas tradicionales de Recuperación de Información; para ello, se realiza una indexación de las "entidades" y las "relaciones" entre estas entidades en los documentos. Para la evaluación del modelo se ha utilizado los corpus CLEF en español e inglés. Para las preguntas en inglés, se ha obtenido una mejora de 35,11% en la precisión media. Para las preguntas en español, el aumento máximo es de 37,18%.
Palabras clave: Procesamiento de Lenguaje Natural, Recuperación de Información

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 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: Natural Language Processing, Information Retrieval

1. Introducción

Una aplicación de Recuperación de Información (RI) recibe como entrada una pregunta de un usuario y tiene que devolver un conjunto de documentos ordenados por su relevancia a la pregunta. En la actualidad esta clase de aplicación es muy importante debido al elevado aumento de información disponible para los usuarios, principalmente a través de Internet.

En la literatura, las técnicas de Procesamiento de Lenguaje Natural (PLN) no han mostrado mejoras significativas en el rendimiento de la recuperación de información, aunque parece que éstas pueden superar los inconvenientes de los métodos puramente cuantitativos de RI, como los métodos estadísticos o las representaciones de sacos de palabras. Como intentos de superar estos inconvenientes están los trabajos de Strzalkowski (1999a) o Baeza-Yates (2004). Tal y como ellos mencionan, una posible explicación son las limitaciones del análisis sintáctico obtenido. Otra posible explicación, quizás más apropiada, sería que las predicciones de uniformidad semántica realizadas sobre la base de las estructuras sintácticas son menos fiables de lo esperado. Por supuesto, la relativa baja calidad del análisis sintáctico puede ser un problema mayor. Voorhees (1999) expone que la falta de buenas técnicas para calcular el peso de los términos compuestos es un factor importante que afecta al PLN comparado con las técnicas actuales de RI.

En este artículo proponemos un método de RI innovador que incorpora técnicas de PLN tales como el etiquetado de las categorías gramaticales de las palabras y el análisis sintáctico para mejorar las representaciones tradicionales de sacos de palabras. Este modelo indexa las entidades y las relaciones entre las mismas. Estas relaciones están basadas en la división de cláusulas del documento, y en la resolución del fenómeno de la anáfora entre estas entidades. Nuestra propuesta mejora otras aproximaciones que usan PLN para la RI, porque mezcla más conocimiento que otras propuestas (morfológico, sintáctico y resolución de la anáfora), y este conocimiento es explotado satisfactoriamente (se han obtenido incrementos de hasta un 37,18%) con el modelo vectorial indexando términos compuestos de una forma efectiva. Además, el modelo es computacionalmente muy eficiente.

En la siguiente sección se presentan los antecedentes de la incorporación del PLN para las tareas de RI. Posteriormente se presenta, de una manera intuitiva, el modelo propuesto en este artículo y su inclusión en el modelo vectorial. Finalmente, se evalúa en los corpus CLEF[1] en español e inglés y es comparado con diversas medidas de similitud.

2. *Antecedentes de PLN en RI*

Los sistemas estadísticos tradicionales de RI buscan las palabras de la pregunta del usuario en los documentos, por lo que consideran relevantes los documentos que tienen estas palabras. Ellos ordenan los documentos relevantes usando distintas medidas de similitud (por ejemplo el modelo vectorial y la medida del coseno). En el conjunto de las *Text REtrieval Conferences*[2] *(TREC)* aparecen distintas aproximaciones estadísticas.

Con el objetivo de mejorar la eficiencia de los sistemas de RI, han aparecido distintas líneas, como los modelos de Recuperación de Pasajes y la aplicación de técnicas de PLN. Sin embargo, hasta el momento las técnicas de PLN no han obtenido mejoras significativas con respecto al esfuerzo computacional que supone la utilización de esta clase de conocimiento.

Los sistemas de RI que usan PLN se pueden clasificar según el tipo de conocimiento que ellos usan. Por ejemplo, algunos de ellos usan información morfológica para usar el lema en vez del "stem" de las palabras, así como diversas derivaciones morfológicas, por ejemplo Vilares et al. (2003).

Otros sistemas usan técnicas de expansión de las preguntas sumando nuevos términos obtenidos de los sinónimos extraídos de WordNet, Gonzalo et al. (1998) o Arampatzis et al. (2000), obteniendo normalmente mejoras en la cobertura (recall) pero empeorando la precisión.

Finalmente, la tercera clase de conocimiento que se ha usado extensivamente para la RI es el sintáctico. La idea básica es indexar grupos de palabras relacionados, en vez de palabras separadas como ocurre en los sistemas tradicionales de RI. El problema principal que tienen estos sistemas es que un mismo concepto se puede expresar mediante distintos árboles sintácticos, por lo que es necesario utilizar una ordenación de la medida de similitud entre diferentes árboles. Otro problema es la calidad, profundidad y robustez del análisis sintáctico. Muchos sistemas han intentado evitar estos problemas indexando palabras contiguas como pares, expresiones ternarias (Zhai et al. 1997, Mitra et al. 1997, Strzalkowski et al. 1999b) o sintagmas (Arampatzis et al. 2000). Con respecto a los pares y expresiones ternarias, estos sistemas normalmente indexan el núcleo de los constituyentes (principalmente sintagmas nominales y verbales) junto con sus modificadores. Por ejemplo, Byung-Kwan et al. (2000) indexan sólo los nombres compuestos coreanos obteniendo únicamente un 0,84% de mejora en la precisión media. Con respecto a los sintagmas, ellos tienen que definir complejas medidas de similitud entre árboles sintácticos.

Algunos sistemas intentan mezclar diferentes clases de conocimiento, incluso junto con el modelo vectorial, como se presenta en Cornelis (2004). Otro ejemplo es Strzalkowski et al. (1999b), en el que usan pares núcleo-modificador para crear un nuevo indicador. Junto con los stems de las palabras y otras cadenas de datos, ellos son

[1] Cross Language Evaluation Forum. http://www.clef-campaign.org/

[2] http://trec.nist.gov

capaces de mejorar un 7% la precisión media en preguntas cortas (con pocas palabras) y un 20% en preguntas largas (más descriptivas), con respecto a un sistema vectorial base sólo con stems. Sin embargo, el componente más importante del sistema continúa siendo el modelo vectorial con stems, donde los pares se usan de forma secundaria. Otro trabajo similar es Alonso et al. (2002), en el que los autores combinan stems, lemas y derivación, junto con pares núcleo-modificador en los corpus CLEF en español, obteniendo únicamente una mejora de 1,59%.

Nuestra propuesta mejora a estas propuestas porque más conocimiento de PLN se mezcla en el mismo modelo: morfológico, sintáctico y resolución de la anáfora. Conocimiento morfológico significa usar un etiquetador de categorías gramaticales para obtener el lema de cada palabra, así como su categoría (nombre común o propio, verbo, etc.). El conocimiento sintáctico usa un analizador sintáctico parcial que lleva a cabo un análisis profundo de los constituyentes que consideramos importantes para extraer los conceptos (sintagmas nominales, oraciones de relativo, aposiciones, sintagmas preposicionales, etc.) y relaciones de estos conceptos (segmentación de cláusulas). La resolución de la anáfora se realiza sobre las descripciones definidas y los pronombres.

La mayor parte de este conocimiento no se ha usado en trabajos previos. Además, nuestra propuesta obtiene resultados satisfactorios con incrementos de hasta un 37,18% en la precisión media, mientras que las otras propuestas normalmente obtienen una mejora alrededor de un 3%. Esto es debido a que nosotros modelamos todo este conocimiento de una manera diferente, ya que no indexamos sólo pares núcleo-modificador, sino que indexamos sintagmas (sintagmas nominales y preposicionales y cláusulas). De este modo, también consideramos que es muy importante la relación entre diferentes modificadores y capturamos más información por medio de la resolución de la anáfora pronominal y las descripciones definidas. Además, el modelo de indexación de los sintagmas permite una mayor normalización de diferentes árboles sintácticos en la misma estructura. Finalmente, nuestro modelo es computacionalmente muy eficiente lo que permite su implementación en aplicaciones reales de RI.

3. El modelo propuesto

Nuestro modelo se basa en la idea intuitiva de que un documento se represente por medio de sus "entidades" y las "relaciones entre sus entidades". Ya que este modelo se basa principalmente en el conocimiento sintáctico, las entidades se representan por medio de sintagmas nominales (NP, *Noun Phrase*), mientras que las relaciones entre ellas se representan mediante cláusulas, en las que el verbo es el núcleo y sus modificadores son los NP y los sintagmas preposicionales (PP, *Prepositional Phrase*). Estas relaciones se completan con la resolución de la anáfora. Veamos el ejemplo (1) donde hay dos entidades: *Mary Blake* y *Mary Spencer*, y del conocimiento sintáctico, obtenemos información adicional acerca de la segunda (*the secretary of ARS*). Además, como se resuelve la referencia anafórica entre *her* y *Mary Blake*, obtenemos más información acerca de esta entidad (*Mary Blake* es *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€.

De este modo, podemos resolver con éxito una pregunta del usuario pidiendo información acerca de *Mary Blake, the president of ISS*, y puede ser descartada para otras preguntas, por ejemplo acerca de *Mary Blake, the president of ARS*.

4. La inclusión de nuestro modelo en el modelo vectorial

En esta sección, el modelo intuitivo se implementa en una representación estadística tradicional o saco de palabras, con el objetivo de superar el problema principal de los métodos estadísticos, es decir, la suposición de que los términos ocurren independientemente de los otros, que no es cierta. Este problema se supera transformando los términos en entidades e introduciendo conocimiento de PLN.

Específicamente, el método estadístico de RI para usar es el modelo vectorial, en el que las preguntas y los documentos se representan como vectores en un espacio n-dimensional, donde *n* es el número de términos indexados, y después son comparados aplicando una medida de similitud tal

pregunta y el documento. La mencionada medida cuantitativa permite ordenar los documentos recuperados.

En la primera subsección, se describen superficialmente las herramientas de PLN usadas para la implementación del modelo. En la subsección siguiente, se introducen las modificaciones del modelo vectorial necesarias para transformar los términos en entidades. Finalmente, se presentan las modificaciones introducidas en la medida de similitud.

4.1 Las herramientas de PLN

Para obtener el conocimiento mencionado en el modelo intuitivo, hemos trabajado sobre la salida del sistema computacional llamado *Slot Unification Parser for Anaphora Resolution (SUPAR)*. Este sistema, presentado en Ferrández et al. (1999), resuelve la anáfora en español e inglés, aunque se puede extender fácilmente a otros idiomas[3].

SUPAR trabaja sobre la salida de un etiquetador, POS tagger (se ha usado el TreeTagger[4] para inglés y para español el Maco[5]), y realiza un análisis sintáctico parcial del texto. SUPAR analiza NP coordinados, PP coordinados, sintagmas verbales y conjunciones, en los que los NP pueden incluir oraciones de relativo, aposiciones, PP coordinados y adjetivos coordinados. Las conjunciones se usan para dividir las oraciones en cláusulas. De este modo, seleccionamos los NP como las entidades del documento y las cláusulas como las relaciones entre las entidades. En (2) se puede observar un ejemplo del proceso de análisis y la detección de las entidades (NP) en una oración (en este caso 10 entidades).

(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 La transformación de los términos en entidades

Para implementar las entidades y las relaciones entre ellas en el modelo vectorial, éstas se representan en tres tablas: NPT, PPT y CCT. NPT almacena información acerca de las entidades sintácticamente representadas como NP. Las dos tablas restantes almacenan información adicional sobre estas entidades o relaciones entre ellas, en la forma de sintagmas preposicionales (PPT) y cláusulas (CCT). La tabla PPT representa algún conocimiento específico acerca de las entidades que se obtienen de la pregunta. Por ejemplo, en la pregunta *architecture in San Louis*, la preposición *in* significa que *San Louis* puede ser una entidad lugar en lugar de una entidad persona. Por lo tanto, un documento en el que aparezca un PP *in San Louis* es valorado más alto que otros documentos en los que *San Louis* no aparezca con esa preposición. Finalmente, la tabla CCT almacena el verbo de la cláusula y todas sus entidades. Por ejemplo, en la segunda cláusula de (1), la tabla CCT almacena el verbo *fine* junto con el NP *Mary Spencer* y *Mary Blake* y *1000€*. De este modo, en el modelo se almacena alguna información extra que no está presente en el modelo vectorial tradicional, por ejemplo las preposiciones y los pronombres (que pertenecen a la lista de palabras de parada[6]), así como la información de cada entidad y las relaciones entre ellas.

Para almacenar las entidades de una manera eficiente computacionalmente, cada tabla almacena el núcleo del constituyente (que será usado para buscar el constituyente) y una lista de modificadores de la entidad (que se usan para afinar la búsqueda). Por ejemplo, en (1) se almacena la siguiente entrada en NPT: *Mary [[Blake, president, ISS], [Spencer, secretary,*

[3] El sistema SUPAR se puede probar en *http://supar.dlsi.ua.es/supar/*. Resuelve la anáfora pronominal en inglés con un 74% de éxito y la anáfora pronominal en español con un 81%.

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

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

[6] Palabras consideradas que no tienen valor de indexación y que son eliminadas del texto.

ARS]], que significa que hay dos entidades con el mismo núcleo (*Mary*), con toda la información obtenida de cada una (*Mary Blake the president of ISS* y *Mary Spencer the secretary of ARS*). De este modo, los NP complejos se representan por estructuras compuestas de núcleos de sintagmas y modificadores. Las estructuras intentan preservar la lógica original del sintagma, a diferencia de otras propuestas anteriores en las que éstas se rompían en pares independientes núcleo-modificadores.

Para cada entrada de la las tablas, también se almacena la información de la frecuencia del modelo vectorial: la frecuencia de la entidad en el documento y la frecuencia en la colección de documentos, así como información adicional que se explicará en las siguientes subsecciones.

En la Tabla 1, se presentan las tablas NPT, PPT y CCT obtenidas para el ejemplo (1), en contraste con la tabla que se obtiene con el modelo vectorial tradicional. En la sección 4.3, explicaremos el modo en que esta información es usada para devolver satisfactoriamente el documento donde se requiere la información acerca de *Mary Blake, the president of ISS*, y no donde se requiere la información acerca de *Mary Blake, the president of ARS*. Las dos subsecciones siguientes dan más detalles sobre estas tablas.

Tabla vectorial tradicional	
Término	Frecuencia
Mary	2
Blake	1
arrived	1
late	1
Spencer	1
secretary	1
ARS	1
fined	1
President	1
ISS	1
1000	1
€	1

Tabla NPT				
Núcleo	Modificadores	Cat	Orac.	Frecuencia
Mary	[Blake, president, ISS], [Spencer, secretary, ARS]	PN	1	2
1000	[€]	CD	1	1

Tabla PPT			
Preposición	Núcleos de NP	Orac.	Frecuencia
of	[ARS], [ISS]	1	2
with	[1000, €]	1	1

Tabla CCT			
Verbo	Palabras con contenido	Orac.	Frecuencia
arrived	[Mary,Blake]	1	1
fined	[Mary,Spencer,Mary,Blake,1000,€]	1	1

Tabla 1. Vectorial tradicional, tablas NPT, PPT y CCT obtenidas para el ejemplo (1).

4.2.1 La tabla NPT

La tabla NPT almacena cada NP del texto. Por ejemplo, *a convertible car* se almacena como *car [convertible]*, es decir, *car* como núcleo y *convertible* como su modificador. Sin embargo, también se almacena la rotación de esta entrada como *convertible [car]* con el objetivo de resolver referencias a esta entidad como *the convertible*. El mismo proceso se sigue para las personas, por ejemplo *John Fitzgerald Kennedy* se almacena como *Kennedy [John, Fitzgerald]*, *Fitzgerald [John, Kennedy]* y *Kennedy [John, Fitzgerald]*, que permite capturar referencias como *John* o *Kennedy*. En los casos en que el valor semántico depende del orden, por ejemplo *junior college* y *college junior*, nuestro sistema distingue entre ambas entidades mediante un conjunto de penalizaciones según el tipo léxico del núcleo del constituyente. Estas penalizaciones se obtienen en la fase de entrenamiento del sistema y serán presentadas en detalle en la sección 4.3. Por ejemplo, un nombre propio se valora como 1,4; un nombre común con 1,0 y un adjetivo con 0,9. Esto significa que si en una pregunta se pide información acerca de *junior college*, y un documento contiene *college junior*, éste se valora más bajo que otro que contenga *junior college*, ya que la rotación del primero se penaliza con 0,9 en su entrada rotada $college_{adjective}$ *[junior]*. Estas rotaciones no se aplican en las oraciones de relativo o PP que puedan aparecer en un NP.

Hay que mencionar que cada núcleo o modificador se almacena como el stem del lema (por ejemplo, para la palabra *escaped*, el lema es *escape*, y su stem *escap*). De este modo se han obtenido los mejores resultados en la fase de evaluación, en comparación con los resultados obtenidos usando el lema o el stem por separado.

Usando esta estrategia, nuestro sistema supera el inconveniente tradicional de las aproximaciones que usan PLN para RI: podemos normalizar fácilmente diferentes estructuras de árboles sintácticos en la misma entidad o concepto. Por ejemplo, *China communist invasion, invasion of*

communist China, invasion of communists of China, invasion of communists that are from China, invasion of China communists, e *invasion of China that is communist*, son almacenados en la entidad *invasion [China, communist]*.

Con respecto al proceso de resolución de la anáfora, consideramos que podemos encontrar una referencia a una entidad mencionada previamente donde existe una relación de inclusión entre las listas de modificadores de ambos NP; por otra parte, podemos encontrar una nueva entidad por lo que en la tabla se almacenará una nueva lista de modificadores. Por ejemplo, en la primera cláusula de (1), almacenamos la entidad *Mary [Blake]*. Cuando el NP *Mary Spencer who is the secretary of ARS* aparece en el texto, ambas entidades comparten el núcleo, *Mary*, pero no comparten ningún modificador. Por lo tanto, las dos entidades se almacenan en la tabla como *Mary [[Blake], [Spencer, secretary, ARS]]*. Finalmente, cuando aparece el NP *her, the president of ISS* y se resuelve el pronombre como *Mary Blake*, el NP se queda como *Mary [Blake, president, ISS]*, ya que *Mary [Blake]* está incluido en él, entonces determinamos que se está refiriendo a la misma entidad, por lo que los nuevos modificadores se añaden a la lista. Finalmente, la entrada en la tabla queda como *Mary [[Blake, president, ISS], [Spencer, secretary, ARS]]*.

4.2.2 Las tablas PPT y CCT

Las tablas PPT y CCT trabajan de manera similar a la tabla NPT; la tabla PPT almacena la preposición y el núcleo del NP, mientras que la tabla CCT almacena el verbo y todas las palabras con contenido de la cláusula.

En (3), se presenta un ejemplo complejo extraído del corpus CLEF Los Angeles Times 1994 (fichero la010194, noticia 166), en el que aparece el funcionamiento real de nuestro sistema. Los NP aparecen marcados entre [ ], las preposiciones están escritas en *itálicas* y los verbos están subrayados. Además, está marcado (tachado) un caso de resolución de la anáfora. El cambio de cláusula se marca con el símbolo "¬". El subíndice $_{(1)}$ marca una aposición y $_{(2)}$ una oración de relativo. El número de la oración está en negrita.

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

Análisis sintáctico, división de cláusulas y resolución de la anáfora:

to Huntington". Estas rotaciones son útiles para capturar referencias a este "sergeant" como *Miller [Larry, Sgt]* y *Larry [Miller, Sgt]*.

Tabla NPT				
Núcleo	Modificadores	Cat.	Orac.	Frec.
435	[pm]	CD	7	1
6902	[warner, av, loan, assn]	CD	7	1
780	[]	CD	5, 6	2
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
case	[]	NN	9	1
door	[west, build]	NN	8	1
fridai	[bank]	PN	6	1
gun	[]	NN	6	1
huntington	[bank, robberi], [beach, polic, sgt, larri, miller]	PN	5, 7	2
larri	[huntington, beach, polic, sgt, miller]	PN	7	1
loan	[warner, av, assn, 6902]	PN	7	1
man	[]	NN	6	2
miller	[huntington, beach, polic, sgt, larri]	PN	7	1
pm	[435]	NN	7	1
polic	[huntington, beach, sgt, larri, miller]	NN, PN	6,7, 9	3
robber	[]	NN	8	1
robberi	[huntintgon, bank]	NN	5, 7	2
save	[world]	PN	7	1
Sgt	[huntington, beach, polic, larri, miller]	PN	7	1
suspect	[case]	NN	9	1
teller	[]	NN	6, 7	2
warner	[av, loan, assn, 6902]	PN	7	1
west	[door, build]	JJ	8	1

$$q_i = \log_e(tf_{q,i} + 1) * \log_e(\frac{N}{df_i} + 1)$$

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

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

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

El parámetro NLPfactor usa el conocimiento obtenido de las técnicas de PLN. Se presenta en [3] y usa distintos parámetros como *depth* (profundidad) que se obtiene del nivel de la pregunta en el árbol sintáctico. Por ejemplo, en la pregunta *architecture in Berlin*, el valor de *depth* de todo el NP (con núcleo *architecture*) is 1, mientras que el valor de *depth* del NP anidado (*Berlin*) es 2. Esto se usa porque el NP con una profundidad más grande restringe más la búsqueda que el NP con una profundidad más baja. Es decir, documentos acerca de "architecture" en general son menos relevantes que aquellos documentos acerca de la "architecture developed in Berlin". En nuestro modelo, esto significa que la entrada *Berlin [ ]* se valorará más que la entrada *architecture [Berlin]*. El valor de *depth* se normaliza mediante el logaritmo, aunque raramente es más alto que 3.

C	Descripción	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)

Tabla 3. Descripción del parámetro *MOD* para la tabla NPT.

Con referencia al parámetro MOD de la ecuación [3], se corresponde con la comparación entre las listas de modificadores de la pregunta y los documentos que comparten el mismo núcleo. Se resume en los cinco casos (C) de la Tabla 3 para la tabla NPT (MOD_{NPT}), donde la operación |…| corresponde a la cardinalidad de una lista, *LModifQuery* es la lista de modificadores de la pregunta, *LModifDB* es la lista de entidades con el mismo núcleo que la pregunta, $LModifDB_i$ es la lista de modificadores de la entidad número *i* almacenada en la tabla NPT, *Common* es el número máximo de modificadores que están repetidos en ambas listas y *R* es el número de listas que tienen *Common* modificadores compartidos. Los coeficientes en las fórmulas varían de acuerdo al idioma (inglés/español) y al tipo de pregunta (larga/corta); estos coeficientes presentados en la Tabla 3 se han obtenido experimentalmente en la fase de entrenamiento para las preguntas cortas en inglés.

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

Tabla 4. Algunos ejemplos del cálculo de MOD_{NPT}.

En la Tabla 4, se presentan algunos ejemplos aclaratorios para cada caso (C) de la Tabla 3. La primera entrada de la Tabla 4 corresponde al caso 1 donde ni la pregunta ni los documentos tienen modificadores, por ejemplo cuando el usuario pregunta por *architecture* sin modificadores. La segunda entrada del caso 1 corresponde a un documento que contiene un NP con núcleo *architecture* con más modificadores: *architectural transformation (...) architecture of Alex Schult*. Este documento se valorará más alto que el primero (MOD_{NPT} es 2,139 comparado con 1,7) porque el segundo documento presenta información adicional acerca del concepto general de la pregunta, mientras que el primer documento no profundiza en el tópico *architecture*. El ejemplo del caso 2 corresponde a la pregunta *Berlin architecture* y un documento con apariciones simples del NP *architecture*. En este caso, el documento es el menos valorado (MOD_{NP} vale 0,6), que es correcto porque el usuario está especificando por medio del modificador *berlin* un tipo de "architecture", mientras que el

documento trata sobre "architecture" en general y no trata de la "architecture" especificada. El caso 3 corresponde a la misma pregunta pero los modificadores del documento añaden más información a la entidad especificada en la pregunta (*a new architectural vocabulary for Berlin (...) monumental architecture (...) official architecture of Berlin*), que es la situación óptima y, consecuentemente, obtiene el máximo MOD_{NPT} (2,415). El caso 4 también tiene un valor alto de MOD_{NPT} (2,388), pero es más bajo que el caso 3 porque hay algunos modificadores de la pregunta (*new architecture of west Berlin*) que no aparecen en la entidad del documento (*author's architecture (...) architecture of west Berlin*), que es correcto. Finalmente, el caso 5 es similar a los casos 3 y 4, pero muestra que no estamos seguros que el documento contenga la entidad buscada, debido a la presencia de modificadores diferentes en la pregunta (*new architecture of west Berlin*) y el documento (*new architecture for east Berlin (...) monumental architecture*).

Los cinco casos (C) de la Tabla 3 también se han obtenido experimentalmente en la fase de entrenamiento para las preguntas largas en inglés y para las preguntas en español (largas/cortas), con similares coeficientes y comportamientos.

Hemos generado dos listas adicionales para la pregunta NPT: una con los modificadores en aposiciones, oraciones de relativo y sintagmas preposicionales (*list3*), y otra con los modificadores restantes (*list2*). Esto se realiza porque pretendemos distinguir entre el conocimiento semántico de cada clase de modificador. De este modo, el parámetro MOD_{NPT} toma el valor de [4]. Esta división, cuyos porcentajes se han obtenido experimentalmente, viene justificada principalmente por el alto porcentaje de errores ocasionados en el análisis por el enlace o anidamiento de las aposiciones, oraciones de relativo y sintagmas preposicionales. De este modo, los modificadores obtenidos de la *list3* se valorarán menos que aquéllos obtenidos de la *list2*.

[4] $MOD_{NPT} = 0,7 * MOD_{list2} + 0,3 * MOD_{list3}$

El parámetro lex de [3] depende del tipo léxico (categoría gramatical) devuelto por el etiquetador para cada núcleo de constituyente. Por ejemplo, en la versión inglesa si el núcleo del constituyente ha sido etiquetado como un nombre propio *lex* valdrá 1,4; si es un nombre común 1,0; si es un adjetivo 0,9; si es un verbo 0,7 y para las restantes etiquetas léxicas valdrá 0,3. Los valores asignados para este parámetro son bastante naturales, ya que los nombres propios son los que tienen más valor (los elementos más importantes en Recuperación de Información). Estos coeficientes también se han obtenido experimentalmente en la fase de entrenamiento para las preguntas en español, con una diferencia que hay que mencionar. En las preguntas en español los nombres comunes tienen menos valor que los adjetivos. Esto se justifica por los errores del etiquetador, específicamente en el etiquetado de nombres, adjetivos y verbos. Como ya se ha mencionado anteriormente, este parámetro nos permite realizar las rotaciones en las entidades de una manera óptima, con el objetivo de capturar diferentes referencias a la misma entidad.

El $NLPfactor_{PPT}$ se obtiene del mismo modo que el $NLPfactor_{NPT}$ pero es diferente por la tabla CCT ya que se obtienen muy malos resultados en la fase de entrenamiento. Tras analizar los resultados, concluimos que hay muy pocas coincidencias entre los verbos de la pregunta y los documentos, además de observar que en las preguntas CLEF había muy pocos verbos. Por lo tanto, decidimos cambiar la fórmula para el $NLPfactor_{CCT}$ como se muestra en [5], para devolver un valor más alto para las pocas coincidencias entre la pregunta y el documento. En esta fórmula, *common* es el numero de modificadores coincidentes y *maxCommon* es la longitud máxima de cualquier lista de modificadores del constituyente.

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

Con respecto al parámetro *proximity* de [2], se usa para capturar la proximidad entre las entidades y penalizar los documentos que tienen las mismas entidades pero más dispersas. Se muestra en [6], donde se usan la distancia media (*averageDistance*) entre los constituyentes (en número de oraciones) y la primera y la última oración (*firstSentence* y *lastSentence*) en la que una entidad de la pregunta aparece en el documento.

[6] $$proximity = \left(1 - slope * \frac{averageDistance}{lastSentence - firstSentence + 1}\right), slope = 0{,}3$$

Después de obtener los tres valores finales de la relevancia del documento (uno para cada tabla: NPT, PPT y CCT), es necesario mezclarlos en un único valor que represente la relevancia que nuestro sistema asigna a un documento. Hemos probado varios métodos, incluyendo los expuestos en Bartell et al. (1994) y Voorhess et al. (1995), sin observar mejoras en los resultados respecto a una suma simple de los pesos de cada tabla. A cada tabla se le asocia un factor de importancia, obtenido experimentalmente en la fase de entrenamiento. Los resultados empíricos revelan una gran importancia de NPT en ambos idiomas, para inglés: 85% (NPT), 5% (PPT) y 10% (CCT); para español: 75% (NPT), 12% (PPT) y 13% (CCT).

5. Evaluación

Hemos realizado distintos experimentos en dos idiomas diferentes (inglés y español) para comprobar el funcionamiento de nuestra propuesta. La misma fórmula [2] se ha usado para los dos idiomas, y para las preguntas cortas y largas, con el objetivo de probar la aplicabilidad del modelo a diferentes tipos de preguntas e idiomas. En ambos casos, los parámetros (*NLPfactor* y *proximity*) y sus coeficientes se han obtenido experimentalmente en la fase de entrenamiento; posteriormente, éstos se han usado en la fase de evaluación.

Para el inglés, se usaron las preguntas del CLEF 2000 y 2002 (de la 1 a la 40 y de la 91 a la 140) para el entrenamiento. Las preguntas del CLEF 2001 (de la 41 a la 90) se usaron para la evaluación[7]. El corpus utilizado es la colección de 113.005 noticias del periódico Los Angeles Times del año 1994.

Para el español, se usaron las preguntas del CLEF 2002 (de la 41 a la 140) para el entrenamiento y las preguntas de 2003 (de la 141 a la 200) para la evaluación. El corpus utilizado es la colección de 454.045 noticias de EFE de los años 1994 y 1995.

Todas las preguntas están en dos versiones: corta y larga. Por ejemplo, la pregunta en (4) tiene tres campos: *title (título), desc (descripción), narr (narrativa)*; para la versión larga se usan los tres campos, mientras que para la versión corta sólo se usan los dos primeros.

(4) **<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 with Scandinavia, the Baltic countries and Russia. Historical or tourist information is not important.

Exp	Descripción	P

[7] La distribución entre preguntas de entrenamiento y preguntas de evaluación se ha realizado de este modo porque teníamos resultados del CLEF 2001 de Llopis and Vicedo (2002); de este modo se pueden realizar comparaciones con nuestra propuesta.

tipo de pregunta (S: corta –short–, L: larga –long–), la cuarta representa la precisión media, la quinta la precisión de los 5 primeros documentos recuperados y la última la R-Precision. Todas estas cifras se han obtenido usando el paquete *trec_eval* con el juicio de relevancia creado por medio de la técnica de "pooling" de los participantes CLEF, con el problema inherente de esta técnica, en la que los documentos no juzgados (aquéllos que no han sido recuperados por ningún participante CLEF) se asumen que no son relevantes.

El experimento 1 muestra los resultados de nuestro baseline: el coseno utilizando los stems de los términos y usando la fórmula [1] (Kaszkiel et al. 1999). En los experimentos 2 hasta el 6, sólo se ha utilizado la tabla NPT, ya que ésta es la más importante en los resultados finales; por simplificar los resultados de entrenamiento las otras dos tablas (PPT y CCT) se han omitido. El experimento 2 muestra los resultados de usar el parámetro *NLPfactor* como *MOD*, tal y como se presenta en la Tabla 3. El experimento 3 muestra los resultados obtenidos con la inclusión del parámetro *lex*. El experimento 4 incluye el parámetro *depth*. En el experimento 5, se usan las dos listas de modificadores de la pregunta como aparece en [4]. En el experimento 6, se usa el parámetro *proximity*. El experimento 7 usa las tres tablas (de los experimentos 2 al 6 sólo se ha usado la tabla NPT). Todos los porcentajes presentados se calculan a partir del baseline (experimento 1).

Como se puede observar en la Tabla 5, se ha obtenido una mejora final de 14,60% en la precisión media en inglés para las preguntas cortas y una mejora del 5,59% para las preguntas largas con respecto al baseline del coseno. Es conveniente destacar que en los experimentos iniciales se obtienen porcentajes negativos, que se han mantenido para mantener el mismo modelo sobre diferentes clases de preguntas e idiomas. Por último, en los experimentos 2 hasta el 6 sólo se usa la tabla NPT (es decir, no se usan los verbos de los documentos), mientras que en el baseline del coseno sí se usan, consecuentemente, las mejoras son más pequeñas.

En la Tabla 6 se muestran los resultados de la evaluación: una mejora de 35,11% en la precisión media para las preguntas cortas, y 12,96% para las preguntas largas respecto al baseline del coseno.

Experimento	P	AvgP 11-p	Precision á 5 docs	R-Precision	Recall
Coseno	S	0,3506	0,4120	0,3301	0,9498
	L	0,4597	0,4640	0,4613	0,9556
Nuestra Propuesta	S	+35,11%	+16,72%	+35,81%	+0,91%
	L	+12,96%	+13,73%	+7,13%	+0,91%

Tabla 6. Resultados obtenidos en la fase de evaluación para inglés.

Resultados similares se obtienen para español, tal y como se observa en la Tabla 7. Hay que destacar que se obtiene una mejora considerable con el experimento 2 (que sólo usa la tabla NPT). Finalmente, se obtiene una mejora de 22,05% en la precisión media en las preguntas cortas y una mejora de 24,03% en las preguntas largas respecto al baseline del coseno.

En la Tabla 8 se muestran los resultados de la evaluación, en la que la mejora es de 27,42% en la precisión media (preguntas cortas), y 37,18% (preguntas largas) respecto al baseline del coseno.

Exp	Descripción	P	AvgP 11-p	Precision á 5 docs	R-Precision
1	Coseno	S	0,3837	0,5313	0,3919
		L	0,4087	0,5879	0,4132
2	$NLPfactor_{NPT} = MOD_{NPT}$	S	+16,50%	+9,13%	+11,71%
		L	+18,86%	+13,31%	+20,08%
3	$NLPfactor_{NPT} = MOD^{lex}$	S	+16,60%	+10,28%	+14,49%
		L	+15,86%	+11,03%	+17,91%
5	$MOD_{NPT} = 0,6* MOD_{list2} + 0,4* MOD_{list3}$	S	+16,91%	+10,28%	+12,58%
		L	+17,03%	+11,03%	+19,62%
6	Usando el parámetro *proximity*	S	+20,85%	+14,83%	+16,10%
		L	+22,00%	+14,83%	+16,10%
7	$0,75*sim_{NPT}+0,12*sim_{PPT}+0,13* sim_{CCT}$	S	+22,05%	+19,39%	+19,19%
		L	+24,03%	+25,09%	+23,78%

Tabla 7. Resultados obtenidos en la fase de entrenamiento para español.

Experimento	P	AvgP 11-p	Precision á 5 docs	R-Precision	Recall
Coseno	S	0,2852	0,3600	0,2963	0,8184
	L	0,3045	0,4100	0,2992	0,7965
Nuestra Propuesta	S	+27,42%	+36,11%	+22,65%	+0,90%
	L	+37,18%	+29,27%	+31,08%	+0,90%

Tabla 8. Resultados obtenidos en la fase de evaluación para español.

Experimento	P	AvgP 11-p	Precision á 5 docs	R-Precision	Recall
Coseno	S	0,3026	0,2723	0,3216	0,9498
	L	0,3557	0,3447	0,355	0,9556
Nuestra Propuesta	S	+15,23%	+34,41%	+4,98%	+0,90%
	L	+10,82%	+20,97%	+10,45%	+0,88%

Tabla 9. Resultados obtenidos en la fase de evaluación para inglés con un corpus adicional: Glasgow Herald.

Para comprobar la portabilidad de nuestra propuesta con nuevos corpus, hemos realizado un experimento adicional, en el que hemos usado nuestro modelo con los mismos parámetros obtenidos en la fase de entrenamiento (es decir, sin entrenamiento adicional para el corpus nuevo) sobre los mismos corpus en inglés más un corpus nuevo, el Glasgow Herald (1995). El corpus nuevo incorporado también se usó en la competición CLEF y está formado por 56.472 documentos. Los resultados se presentan en la Tabla 9, y aunque no se ha realizado un entrenamiento previo, también se han obtenido mejoras considerables.

Con los resultados presentados, queda demostrada la aplicabilidad de nuestra propuesta sobre diferentes idiomas y corpus. Posteriormente, hemos probado su aplicabilidad sobre diferentes fórmulas de similitud: el coseno pivotado (Singhal et al. 1996), en la que se usaron los mismos parámetros *NLPfactor* y *proximity*. Los procesos de entrenamiento y evaluación se llevaron a cabo de forma similar. Los resultados de la evaluación se muestran en la Tabla 10, en la que también se obtienen mejoras considerables respecto al coseno pivotado (para inglés 21,12% y 9,91%, para español 19,76% y 36,67%).

Experimento	P	AvgP 11-p inglés	AvgP 11-p español
Coseno	C	0,3506	0,2852
	L	0,4597	0,3045
Coseno Pivotado	C	0,4120	0,3725
	L	0,4640	0,3469
Nuestra Propuesta respecto al Coseno	C	+35,11% (0,4737)	+27,42% (0,3634)
	L	+12,96% (0,5193)	+37,18% (0,4177)
Nuestra Propuesta respecto al Coseno Pivotado	C	+21,12% (0,4990)	+19,76% (0,4461)
	L	+9,91% (0,5100)	+36,67% (0,4741)

Tabla 10. Resultados obtenidos con el coseno pivotado.

Este artículo se centra en analizar el modo en el que la introducción de entidades en vez de términos, y la introducción de los parámetros *NLPfactor* y *proximity* puede aumentar la precisión (hasta un 37,18%), cuando se usan diferentes fórmulas de similitud. Hemos obtenido importantes mejoras en la precisión media en experimentos realizados sobre las colecciones CLEF para inglés y español, mucho más altos que los obtenidos en trabajos previos. Por ejemplo, Strzalkowski et al. (1999b) usan pares núcleo-modificador para crear un nuevo indicador y ellos mejoran un 7% la precisión media en preguntas cortas y un 20% en preguntas largas (en vez de nuestros 35,11% y 12,96%), pero el componente más importante del sistema continúa siendo el modelo vectorial con stems, donde los pares se usan de forma secundaria. Sin embargo, nosotros sólo usamos nuestro modelo y no con una combinación del modelo vectorial. Con respecto a los corpus en español, Alonso et al. (2002), en el que los autores combinan stems, lemas y derivación junto con los pares núcleo-modificador, ellos sólo obtienen una mejora del 1,59%, frente a los 37,18% que llegamos a obtener con nuestra propuesta.

Con respecto al hecho de que la mejora para las preguntas largas en inglés no es tan buena como para las preguntas cortas, sugiere que hay que realizar un procesamiento especial, por ejemplo realizando una mayor comprensión de la pregunta para tratar las negaciones como ocurre en el ejemplo (5): *"not about…"*. Además, se debe detectar el NP principal, por ejemplo en la versión narrativa de la pregunta (4) (página 10) el NP principal es *political or economic information on Kaliningrad*, mientras que los restantes NP introducen información menos relevante.

(5) 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.

Para concluir, sólo unos comentarios acerca de la implementación computacional de nuestra propuesta. Se ha implementado en lenguaje PHP de forma eficiente para evitar el inconveniente presentado en la mayoría de las anteriores propuestas que incorporaban el uso de PLN en RI. Así, 75 preguntas cortas en español del CLEF realizadas sobre el corpus español descrito anteriormente (2,1 GB de texto) tardaron 7 minutos en un Pentium IV á 2,8 GHz, mientras que estas preguntas en su versión larga tardaron 16 minutos.

6. Conclusiones

En este artículo hemos presentado un modelo innovador de RI que usa exhaustivamente PLN.

Este modelo supera los problemas de las aproximaciones tradicionales que usan sacos de palabras (que asumen que los términos ocurren independientemente unos de otros, que no es cierto) indexando las "entidades" y las "relaciones" entre estas entidades en los documentos. También supera los problemas del uso de conocimiento extraído con el PLN para la RI, ya que la mayoría de este conocimiento se mezcla en el mismo modelo: información morfológica, sintáctica y resolución de la anáfora. La mayoría de este conocimiento no se ha usado previamente en otras propuestas. Además, nuestro modelo mejora otras propuestas indexando no sólo pares independientes núcleo-modificador, sino sintagmas, para considerar relaciones entre diferentes modificadores. Otra contribución de este trabajo es que el conocimiento de PLN se incorpora en el modelo de RI de una forma computacionalmente eficiente, con el fin de evitar el inconveniente computacional presentado en la mayoría de las propuestas que usan PLN en la RI.

Nuestra propuesta se ha usado en el modelo vectorial y se han realizado dos modificaciones. La primera es almacenar entidades en vez de términos. La segunda es la introducción de dos parámetros adicionales en la medida de similitud entre los vectores de la pregunta y los documentos: *NLPfactor* y *proximity*. Como herramientas de PLN se han usado un etiquetador de categorías gramaticales, un analizador sintáctico parcial y un módulo de resolución de la anáfora.

Hemos evaluado la aplicabilidad de nuestro modelo en dos idiomas (español e inglés), sobre dos versiones diferentes de una pregunta (larga y corta), sobre diferentes corpus CLEF y utilizando distintas medidas de similitud (coseno y coseno pivotado). De este modo, se ha probado la portabilidad y generalización del modelo obteniendo grandes mejoras en la precisión media con respecto al modelo vectorial. Para las preguntas cortas en inglés se ha alcanzado una mejora de 35,11% en la precisión media y en las largas un 12,96%. Las preguntas cortas en español tienen un aumento de 27,42% y las largas 37,18%. Se han obtenido porcentajes similares para el coseno pivotado: para inglés 21,12% y 9,91%; para español 19,76% y 36,67%. Estos incrementos son mucho más altos que los obtenidos en trabajos previos.

Como trabajos futuros, los autores continúan desarrollando nuevos módulos y optimizando los ya existentes para mejorar el sistema global. Aunque obtenemos buenos resultados, nuestro sistema aún no aprovecha al máximo todos los recursos que utiliza. Específicamente, las tablas PPT y CCT contribuyen con una mejora del orden de un 2% en ambos idiomas, razón por la que debemos mejorar el sistema mediante un uso más efectivo de estas tablas. Por último, el sistema se debe probar en otros idiomas y corpus, así como en tareas de Búsqueda de Respuestas (Question Answering).

7. Bibliografía

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.

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