# Experiencias del Grupo COLE en la aplicación de técnicas de Procesamiento del Lenguaje Natural a la Recuperación de Información en español

**Miguel A. Alonso**
**Jesús Vilares**

Departamento de Computación
Universidade da Coruña
Campus de Elviña s/n
15071 La Coruña (España)
{alonso,jvilares}@udc.es

**Francisco J. Ribadas**

Departamento de Informática
Universidad de Vigo
Campus de As Lagoas s/n
32004 Orense (España)
ribadas@uvigo.es

## Resumen

La aplicación de técnicas de Procesamiento del Lenguaje Natural a la Recuperación de Información ha sido objeto de estudio en numerosas ocasiones, si bien tales trabajos se han centrado, mayoritariamente, en el inglés. En el presente artículo repasamos la trayectoria de las investigaciones llevadas a cabo por nuestro grupo para el caso del español: desde nuestros experimentos iniciales, caracterizados por la falta de recursos de evaluación estándard, hasta nuestra reciente participación en el CLEF 2002, que ha supuesto un punto de inflexión en nuestras investigaciones.

**Palabras clave:** Recuperación de Información, Procesamiento del Lenguaje Natural.

## Abstract

The employment of Natural Language Processing techniques for Information Retrieval has been studied many times, but such works have mainly focused on English. In this article we describe the evolution of the research developed by our group for the case of Spanish: from our initial experiments, characterized by the lack of standard resources for evaluation, up to our recent participation in CLEF 2002, which has been a milestone in our research.

**Keywords:** Information Retrieval, Natural Language Processing.

# 1 Introducción

La *Recuperación de Información* (IR) es el área de la ciencia y la tecnología que trata de la representación, almacenamiento, organización y acceso a elementos de información. Idealmente, un proceso de IR produce como salida, ante una consulta dada por una necesidad de información del usuario, un conjunto de documentos cuyo contenido satisface dicha necesidad.

A lo largo de las últimas décadas se han desarrollado diversos modelos y técnicas para realizar el emparejamiento entre consultas y documentos [6, 33], debiendo tener presente que en casi todos estos desarrollos se ha tomado como premisa que tanto las consultas como los documentos estarían escritos en inglés. Sin embargo, dicho proceso de emparejamiento puede verse perturbado por múltiples factores, entre los cuales destacan la inherente ambigüedad de las lenguas naturales y la variación lingüística [5], ya que la forma de expresar conceptos utilizada en la consulta puede no corresponderse con la utilizada en todos o parte de los documentos. El caso peor vendría dado por la existencia de un conjunto de documentos con información relevante pero expresada de forma muy distinta a como aparece en la consulta, a la vez que existiese otro conjunto de documentos no relevantes pero que contuviesen todos o parte de los términos empleados en la consulta.

Puesto que los sistemas de IR deben tratar con textos en lenguaje natural, no siempre bien estructurados, y con ambigüedades a nivel léxico, sintáctico y semántico, parece lógico pensar que la utilización de técnicas de *Procesamiento del Lenguaje Natural* (NLP) debería ayudar a incrementar su rendimiento. Sin embargo, su uso supone un desafío importante, ya que para poder trabajar con grandes colecciones de documentos, las técnicas que se apliquen deben ser lo suficientemente rápidas. En particular, la aplicación de cualquier técnica cuyo coste no crezca a lo sumo linealmente con el tamaño de la colección es, por el momento, inviable desde el punto de vista práctico. Esta limitación deja fuera de juego un gran número de potentes algoritmos de análisis sintáctico completo, de desambiguación del sentido de las palabras y de análisis semántico en general. Nos vemos abocados por tanto a trabajar con técnicas basadas en tecnología de estado finito (autómatas y traductores de estado finito, expresiones regulares, etc.). En el mejor de los casos, podremos realizar aproximaciones de las costosas técnicas de análisis sintáctico y semántico [39].

Los primeros intentos de aplicación de técnicas de NLP en recuperación de información se realizaron sobre sistemas que procesaban textos escritos en inglés. Los resultados no fueron excesivamente halagüeños, ya que en la literatura sólo se informa de pequeñas mejoras con respecto a aquellos sistemas que no usan técnicas de NLP [23, 34, 24]. La posible causa de este rendimiento, menor del esperado, sería doble. Por una parte podemos argumentar que las técnicas utilizadas no son suficientemente potentes como para capturar la semántica de documentos y consultas más allá de lo que han conseguido las técnicas basadas en texto crudo. Por otra parte, podemos suponer que las técnicas de NLP mostrarán toda su potencia cuando sean aplicadas a lenguas con una estructura morfológica y sintáctica más compleja que la del inglés, como puede ser la de las lenguas latinas, entre ellas el español.

Bajo estas premisas, a finales de los años 90, el grupo de investigación de Compiladores y Lenguajes (COLE)[1], formado por investigadores de las universidades de La Coruña y Vigo, decidió iniciar una línea de investigación que tiene como fin incrementar el rendimiento de los sistemas de Recuperación de Información que trabajan sobre textos escritos en español y gallego, aplicando para ello técnicas de Procesamiento del Lenguaje Natural. En lo que resta de este artículo describimos las diferentes etapas en las que se ha desarrollado esta investigación. En la sección 2 describimos la evolución de nuestra investigación en dicho ámbito, la cual queda reflejada en nuestro sistema de IR. La sección 3 resalta la dificultad que supuso la carencia de un corpus estándar sobre el que evaluar el rendimiento de las herramientas desarrolladas. En la sección 4 se relata el cambio cualitativo y cuantitativo que conllevó la disponibilidad de un corpus estándar proporcionado por la organización del CLEF. Finalmente, en la sección 5 resumimos las conclusiones más importantes de nuestra actividad investigadora y esbozamos nuestras futuras líneas de trabajo.

# 2 Arquitectura del sistema

Los inicios de la investigación del grupo COLE en el área de Recuperación de Información se enmarcaron en el ámbito del proyecto ERIAL [7], que tenía como meta el desarrollo de un sistema de IR en español y gallego que hiciese uso del co-

[1] `http://www.grupocole.org`

nocimiento de las particularidades morfológicas y sintácticas de estas dos lenguas para incrementar el rendimiento de los sistemas disponibles comercialmente. En este proyecto, los investigadores del grupo COLE se encargaron del diseño de la arquitectura general del sistema, del desarrollo de las herramientas de NLP necesarias y de su integración en el sistema final, denominado LEIRA.

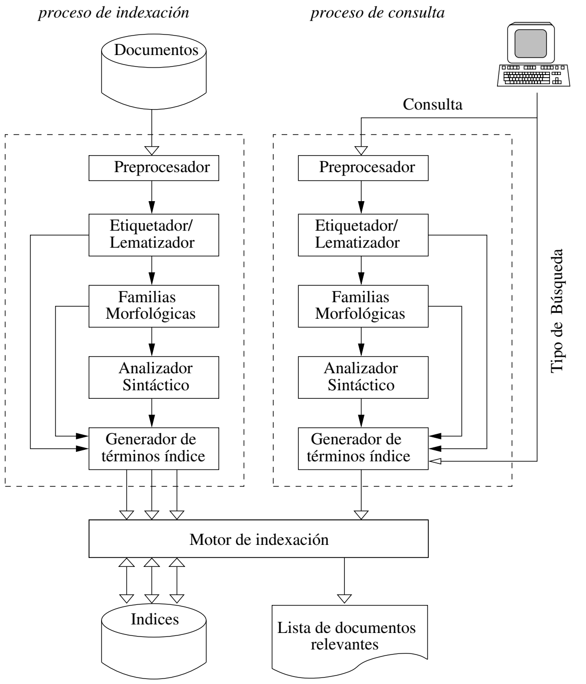


**Figura 1: Arquitectura de LEIRA**

En la figura 1 se muestra un esquema de la arquitectura general del sistema LEIRA, la cual se mantendrá, con ligeros cambios, en los sistemas de IR desarrollados por el grupo en años posteriores. Como se puede observar, el criterio de diseño principal consistía en la extracción, tanto de los documentos como de las consultas, de términos índice mediante técnicas de base lingüística. Como es habitual, coexisten dos flujos de procesamiento gemelos, uno para el tratamiento de los documentos y otro para el de las consultas. Describiremos a continuación los módulos involucrados en estos procesos.

## 2.1 Preprocesador

El primer paso en el procesamiento tanto de documentos como de consultas consiste en preprocesar el texto para, por una parte, segmentarlo adecuadamente, y por otra, para realizar diversas transformaciones en el mismo y así facilitar el trabajo de las fases posteriores [17]. Tales transformaciones comprenden el reconocimiento de números, fechas, abreviaturas, acrónimos, separación de contracciones, separación de pronombres clíticos, identificación de locuciones e identificación de nombres propios. Debemos señalar que la mayoría de sistemas de IR no suelen tratar tales fenómenos, lo cual en ocasiones redunda en normalizaciones erróneas que afectan negativamente el buen rendimiento del sistema.

De todas estas tareas, la que mayor conocimiento lingüístico requiere es la de separación de clíticos, ya que es preciso estudiar con detenimiento qué formas verbales permiten la concatenación de qué pronombres, así como las secuencias válidas de pronombres.

Sin embargo, la tarea computacionalmente más compleja es la de identificación de nombres propios, ya que ante la imposibilidad de disponer de un diccionario con todos los posibles nombres de personas, lugares y entidades, se optó por que el sistema aprendiese los nombres propios que aparecen en los documentos a indexar. Para ello es necesario realizar una primera lectura de dichos documentos, en la cual se extraen las secuencias de palabras con su primera letra en mayúscula y que estén ubicadas en posiciones en donde la utilización de mayúsculas indica sin ambigüedad alguna que estamos ante un nombre propio. Con el fin de poder identificar nombres complejos se permite que dichas secuencias estén conectadas por un conjunto de preposiciones y determinantes previamente establecido.

A pesar de que la correcta identificación de nombres propios es vista por algunos autores como una importante tarea de cara a conseguir sistemas de IR de alto rendimiento [27, 35], nuestra experiencia en este ámbito nos indica que el reconocimiento de nombres propios compuestos degrada el rendimiento ya que impide el emparejamiento de formas que, si bien se refieren a la misma entidad, están expresadas de forma distinta [8]. Es por ello que la identificación de nombres propios complejos como una unidad se desestimó finalmente en favor del reconocimiento separado de sus componentes simples.

## 2.2 Etiquetador-Lematizador

Tras segmentar y preprocesar el texto, el siguiente paso consiste en etiquetarlo y lematizarlo. Con ello conseguimos obtener los rasgos morfosintácticos más relevantes de cada palabra, información que será utilizada en las siguientes etapas, al tiempo que logramos eliminar la variación flexiva, normalizando las distintas realizaciones de verbos, sustantivos y adjetivos, a una sola forma, su lema: es decir, el infinitivo en el caso de los verbos, y la formas masculina singular en el caso de sustantivos y adjetivos. Nos centramos en estas tres categorías de palabras, denominadas *palabras con contenido*, porque existe un acuerdo general respecto a que es en ellas en las que reside la semántica de un documento [20], aunque se podría discutir si ciertos adverbios merecen ser tenidos también en consideración. Esta tarea podría llevarse a cabo mediante cualquier técnica de etiquetación de alto rendimiento. En nuestro caso se optó por integrar el etiquetador-lematizador `MrTagoo`, desarrollado en el seno de nuestro grupo de investigación [14], y basado en Modelos de Markov Ocultos (HMM). Esta decisión se debe a que presenta ciertas características que lo hacen especialmente adecuado para la tarea, como son una estructura de almacenamiento y búsqueda muy eficiente basada en autómatas finitos [16], el manejo de palabras desconocidas, la posibilidad de integrar diccionarios externos en el marco probabilístico definido por los HMM [18], así como la posibilidad de manejar segmentaciones ambiguas [15].

## 2.3 Familias morfológicas

Una vez eliminada la flexión del texto mediante la lematización, la variación morfológica queda reducida a la variación derivativa. Con objeto de eliminarla, la siguiente etapa consiste en agrupar las palabras derivables unas de otras mediante los mecanismos propios de la morfología derivativa. Cada una de estas agrupaciones recibe el nombre de *familia morfológica*. Ante la imposibilidad de crear a mano todas las familias, se optó por diseñar una herramienta automática que las generase a partir de las reglas más comunes de derivación, entre las cuales citamos la prefijación, la sufijación apreciativa, la sufijación no apreciativa, la parasíntesis y la derivación regresiva [22]. Si bien nuestra solución comete errores, su tasa de fallo resultó ser lo suficientemente baja como para permitir su aplicación práctica en entornos de IR [40].

## 2.4 Analizador sintáctico

En la siguiente etapa de procesamiento se trata la variación sintáctica presente en el texto. Para ello se procede a realizar un análisis sintáctico superficial de los documentos y las consultas, mediante el cual se pretende extraer aquellos pares de palabras que se encuentran ligadas por medio de algún tipo de dependencia sintáctica. En particular, nos centramos en las dependencias que se establecen entre el núcleo de un sintagma nominal y el núcleo de sus modificadores, entre el núcleo del sujeto y el verbo, y entre el verbo y el núcleo de sus complementos. Esta tarea se podría realizar de una forma bastante precisa aplicando cualquiera de los potentes algoritmos de análisis sintáctico descritos en la literatura [32] empleando una gramática de amplia cobertura del español. Desafortunadamente, el coste computacional de los algoritmos de análisis sintáctico nos hace desistir de su utilización. En este punto debemos recordar que los analizadores sintácticos para gramáticas independientes del contexto generales tienen un coste temporal que crece cúbicamente con respecto al tamaño de los textos a analizar. Esta complejidad es incluso mayor en el caso de formalismos gramaticales más adecuados a la descripción lingüística de los fenómenos sintácticos [2]. Por si esto no fuese obstáculo suficiente, no se dispone de ninguna gramática de amplia cobertura del español, ni tampoco de un banco de árboles a partir del cual extraerla. Con estas limitaciones en mente, se decidió aplicar la siguiente metodología para la extracción de las dependencias sintácticas [38, 4]:

- En una primera etapa se estudiaron las construcciones sintácticas a considerar, tomando como base la estructura de los sintagmas nominales del español. Para ello se levantaron los árboles correspondientes a cada una de las posibles formas de construir un sintagma nominal, teniendo en cuenta sus posibles complementos, y tratando de generar un árbol lo más bajo posible. Una vez creados los árboles, se procedió a aplicar sobre ellos las transformaciones sintácticas y morfosintácticas más frecuentes en español [21, 20], dando lugar a nuevos árboles, cuyo alcance puede llegar a abarcar la oración completa en el caso de entrar en juego verbos derivados de los términos del árbol original.

- En una segunda etapa los árboles resultantes fueron aplanados con el fin de obtener una expresión regular, en base a las categorías gramaticales de los términos involucrados, que representase de forma aproximada los sintagmas y frases generados por los árboles obtenidos en la etapa precedente. Finalmente, se estableció la correspondencia entre las palabras involucradas en las dependencias descritas por los árboles y las palabras de la expresión regular asociada.

- Finalmente, nuestro analizador sintáctico superficial emplea dichas expresiones regulares para producir una lista de los pares de palabras ligadas por dependencias sintácticas presentes en el texto a analizar. Para hacer frente a las restricciones de rendimiento comentadas anteriormente, nuestro analizador fue implementado empleando técnicas de estado finito.

### 2.5 Generador de términos índice

Por último, tal y como se indica en la figura 1, los términos índice obtenidos mediante cualquiera de las técnicas citadas (lematización, familias morfológicas, y dependencias sintácticas) o una combinación de ellas, son utilizados como entrada por un motor de indexación. Precisamente uno de los criterios de diseño fue el de mantener la independencia con respecto al motor de indexación, lo que nos permite comprobar la variación de rendimiento introducida por la utilización de técnicas de NLP en diversos sistemas y modelos de IR.

De la descripción realizada se observa que en la arquitectura diseñada la mayor parte de la carga computacional recae en la fase de indexación, alejándonos en cierta medida de otros enfoques que propugnan una indexación sencilla de los documentos y un tratamiento computacional intensivo de la consulta mediante diversas técnicas de expansión de la misma.

## 3 Evaluación sobre una colección no estándar

Cuando se trabaja sobre documentos escritos en inglés, la evaluación de un sistema de IR puede realizarse sobre colecciones de documentos bien conocidas y de libre acceso, empleando un conjunto de consultas previamente establecidas y analizadas por la comunidad científica. Esto permite establecer comparaciones homogéneas entre diferentes sistemas y así determinar las circunstancias en las que dichos sistemas se comportan bien o mal (consultas cortas o largas, documentos pequeños o grandes, variabilidad de los términos involucrados en las consultas, etc.). De todas las colecciones disponibles, sin duda la más importante es la utilizada en la competición TREC[2].

Sin embargo, cuando tratamos de realizar una evaluación empírica de nuestro sistema de IR, nos encontramos con un panorama completamente distinto, ya que se carecía de una colección estándar para español sobre la que realizar las medidas. En consecuencia, nos vimos obligados a construir nuestra propia colección de prueba, conscientes de las limitaciones que ello suponía, pues al carecer la comunidad científica de conocimiento de las características de los documentos y consultas utilizados, y al no ser reproducibles los resultados por otros grupos, el interés de los resultados podía ser fácilmente cuestionado. Para superar en lo posible estas limitaciones, decidimos realizar nuestros experimentos siguiendo lo más fielmente posible los criterios establecidos en TREC, a pesar del trabajo que ello suponía.

### 3.1 Creación de la colección

El primer paso consistió en crear una colección de documentos. Se decidió recopilar artículos periodísticos, alcanzándose finalmente un conjunto de 21.899 documentos, con una longitud media de 447 palabras, que cubrían secciones dispares como nacional, internacional, economía y cultura, aunque no de deportes, y siempre dentro del año 2000. En la colección se utilizaban 154.419 palabras distintas, que aparecían un total de 9.870.513 veces. Como dato interesante, señalamos que una vez lematizado el texto, obtuvimos 111.982 lemas únicos correspondientes a palabras con contenido, los cuales ocurrían 4.625.579 veces en los documentos.

La creación del conjunto de consultas sobre el que realizar la evaluación resultó ser más dificultosa. El proceso seguido se relata a continuación:

- En primer lugar, se realizó una lectura superficial de gran parte de los documentos con el fin de detectar temas recurrentes, ya que se trataba de realizar consultas para las que se

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

obtuviese un número significativo de documentos.

- Una vez determinados los temas de interés se creó, en base a ellos, un conjunto de consultas de referencia. Dicho conjunto estaba formado por 14 consultas, de una longitud media de 7,85 palabras, de las cuales 4,36 eran palabras con contenido.

- Cada consulta se utilizó como entrada a diferentes versiones del sistema, correspondientes a la utilización individual de las siguientes técnicas de normalización basadas en NLP: normalización mediante lemas, mediante familias morfológicas, y mediante pares de dependencia. Adicionalmente, se utilizaron las consultas en un sistema que trataba con textos sin procesar lingüísticamente. A su vez, cada una de estas técnicas se probó utilizando tres motores de indexación diferentes: el comercial Altavista SDK[3], el vectorial SMART [10] y el booleano SWISH-E[4].

- Para cada consulta y técnica de normalización se tomaron los 100 primeros documentos devueltos por el sistema, determinando manualmente cuáles eran relevantes y cuáles no.

- Una vez obtenido así el conjunto de documentos relevantes para cada consulta, se consideró que cualquier otro documento era no relevante para dicha consulta.

Establecido ya un conjunto de consultas de referencia y sus correspondientes documentos relevantes, procedimos a evaluar el comportamiento de las técnicas de normalización consideradas. Las pruebas mostraron que la lematización obtenía un buen rendimiento general para todos los niveles de cobertura, mientras que la utilización de pares de dependencia incrementaba la precisión en los niveles bajos, un resultado interesante al implicar una mayor probabilidad de que los primeros documentos devueltos por el sistema sean relevantes para el usuario. También observamos que la utilización de dependencias sintácticas arrojaba mejores resultados cuando el emparejamiento consulta-documento se realizaba de un modo aproximado, como en SMART (modelo vectorial) o en Altavista (aproximación al modelo vectorial). En cambio, cuando se requería la presencia de todos los términos índice extraídos de la consulta, como en el caso del SWISH-E (modelo booleano), el rendimiento decaía notablemente.

[3]`http://solutions.altavista.com`

[4]`http://sunsite.berkeley.edu/SWISH-E/`

Debemos también señalar que el rendimiento de cada técnica de NLP varía según las características de cada consulta. Un estudio más detallado de los resultados obtenidos puede encontrarse en [41].

# 4 Evaluación sobre la colección CLEF

El Cross-Language Evaluation Forum (CLEF)[5] proporciona la infraestructura necesaria para la prueba y evaluación de sistemas de IR, tanto multilingües como monolingües, siempre que estos últimos utilicen una lengua europea distinta del inglés. Para ello el CLEF dota a sus participantes de colecciones estándar de documentos y consultas sobre las que trabajar. En su primera edición, celebrada en el año 2000, se encontraban disponibles colecciones para inglés, francés, alemán e italiano. Afortunadamente, en el CLEF 2001 [26] se incorporó la colección de documentos en español. La aparición de esta colección estándar supone un hito importante en el desarrollo cuantitativo y cualitativo de la investigación sobre IR en español. Como hemos señalado anteriormente, no es sino a partir de ese momento en el que se puede hablar de una evaluación en igualdad de condiciones con respecto a los sistemas que trabajan en lengua inglesa[6]. Nuestro sistema no estaba lo suficientemente maduro a finales del año 2000 como para participar en la edición del 2001, pero sí estuvo preparado para participar en CLEF 2002 [25].

## 4.1 Características de la colección

El corpus español empleado en el CLEF 2002 estaba formado por 215.738 documentos SGML correspondientes a los despachos de noticias de 1994 proporcionados por la Agencia EFE[7], con un espacio en disco de 509 MB, 438 de ellos de texto. La evaluación del año 2002 se realizó sobre un conjunto de 50 consultas formadas por un breve título, una somera frase de descripción y un pequeño texto especificando los criterios que utili-

[5]`http://www.clef-campaign.org`

[6]A este respecto, debemos indicar que el tamaño de las colecciones TREC para inglés es todavía varias veces el de la colección CLEF para español y que el número y variedad de consultas es mucho más elevado, resultado de una década de trabajo. Sin embargo, tenemos fundadas esperanzas de que con el paso de los años las diferencias sean cada vez menos significativas.

[7]`http://www.efe.es`

zarán los revisores para establecer la relevancia de un documento respecto a la consulta. Asimismo, se encontraban también disponibles las 50 consultas utilizadas en CLEF 2001, junto con las listas de documentos relevantes asociados a cada una de ellas. De este modo se pudieron realizar pruebas preliminares del sistema antes de la competición oficial.

Llegados a este punto, debemos indicar que existe una importante diferencia entre la colección CLEF y la colección propia que habíamos venido utilizando hasta entonces. Ambas están formadas por textos periodísticos, si bien en nuestra colección los textos correspondían a artículos editados en un periódico, y que por tanto habían superado diversas etapas de revisión, por lo que apenas se encontraban palabras mal escritas o frases mal construidas. Por contra, la colección CLEF está formada por despachos de noticias, escritos a veces muy rápidamente y que, en consecuencia, contienen numerosos errores ortográficos y de sintaxis [13]. Además, ciertas partes del texto, habitualmente el título y otras partes sobre las que se desea atraer la atención del lector, se encuentran escritas completamente en mayúsculas y sin signos ortográficos. Nuestras herramientas de NLP, en particular el preprocesador y el etiquetador-lematizador, no estaban preparadas para hacer frente a este tipo de entradas, por lo que con frecuencia los títulos de los documentos eran mal etiquetados. Para solventar este problema hubo que desarrollar un módulo adicional para el procesado de entradas de tal naturaleza, recuperando, de ser necesario, su forma en minúscula y con signos ortográficos. El comportamiento del módulo y su impacto en los resultados se describen en mayor profundidad en [36, 37].

## 4.2 Resultados

Empleando el corpus del CLEF 2002, se repitieron los experimentos realizados anteriormente sobre nuestra propia colección, con la salvedad de que en este caso se empleó únicamente el motor de indexación SMART, ya que al ser el Altavista SDK un sistema comercial, su modelo de emparejamiento no es completamente público, y por lo tanto se desconoce si los resultados obtenidos son portables a otros modelos, mientras que el tipo de emparejamiento completo forzado por SWISH-E parece no resultar muy adecuado en tareas generales de IR.

Con motivo de nuestra participación en el CLEF 2002, decidimos también probar una nueva técnica, esta vez encaminada al tratamiento de la variación léxica en los términos simples, y basada en la expansión de consultas mediante relaciones de sinonimia ponderada. Para ello utilizamos una versión electrónica del diccionario de sinónimos de Blecua [9], del que se extrajeron todos los posibles pares de sinónimos, asignándoles a cada uno de ellos un valor indicativo de su grado de sinonimia: un valor de 1 implica una sinonimia perfecta de acuerdo con el diccionario, mientras que aquellos valores cercanos a 0 indican una relación marginal de sinonimia [12].

Los resultados obtenidos [36] confirmaron que la normalización mediante lemas constituía un buen método de partida en la aplicación de técnicas NLP para el tratamiento de la variación lingüística en IR. La lematización superó claramente los resultados obtenidos mediante *stemming*, aun cuando el *stemmer* empleado, perteneciente al motor de búsqueda de código abierto Muscat [28] y basado en el algoritmo de Porter [6], pretendía resolver a mayores la variación derivativa, y no sólo la flexiva como ocurre para la lematización. Nuestro intento por tratar la variación derivativa mediante familias morfológicas resultó introducir más ruido de lo que suponíamos, lo que se tradujo en una degradación de los resultados. De forma similar, la expansión de consultas mediante la utilización de sinonimia ponderada tampoco logró mejorar los resultados obtenidos por la lematización inicial, aunque la distorsión introducida fue menor. Por otra parte, la indexación de pares de dependencia sintáctica mostró un comportamiento peor de lo esperado, no logrando batir a la indexación mediante lemas.

Tras la presentación de resultados oficiales en el congreso CLEF 2002 [36], decidimos modificar el esquema de pesos utilizado en nuestros experimentos con SMART, pasando de un esquema `ltc-lnc` [11] a un esquema `atn-ntc` [30], logrando así mejorar significativamente los resultados [37]. Ello nos lleva a considerar que las técnicas de NLP aplicadas a sistemas de IR escalan bien cuando son aplicadas sobre modelos de indexación cada vez más sofisticados. En esta misma línea, y como novedad, mostramos en la Tabla 1 los resultados conjuntos obtenidos para el corpus CLEF de las ediciones 2001 y 2002, empleando el motor de indexación ZPrise[8] con un esquema de pesos Okapi BM25 [29].

El rendimiento de las diferentes técnicas se ha me-

[8] `http://www.itl.nist.gov`

Tabla 1: Experimentos con Okapi BM25

| | *stm* | *lem* | *sin* | *fam* | *f-pds* |
|---|---|---|---|---|---|
| Docs. | 99k | 99k | 99k | 99k | 99k |
| Rlvs. dev. | 5282 | 5320 | 5315 | 5296 | 5270 |
| R-pr. | .5312 | .5406 | .5397 | .5367 | .4893 |
| Pr. no int. | .5571 | .5733 | .5710 | .5636 | .5122 |
| Pr. doc. | .6238 | .6416 | .6377 | .6215 | .5597 |
| Pr. 11-pt. | .5653 | .5793 | .5766 | .5706 | .5237 |
| Pr. 3-pt. | .5778 | .5961 | .5941 | .5833 | .5283 |
| $N$ | Precisión | | | | |
| a 5 | .7172 | .7475 | .7495 | .7374 | .6667 |
| a 10 | .6364 | .6525 | .6525 | .6465 | .6192 |
| a 15 | .5838 | .6020 | .5987 | .5906 | .5569 |
| a 20 | .5470 | .5641 | .5621 | .5586 | .5157 |
| a 30 | .4872 | .5040 | .5007 | .4906 | .4545 |
| a 100 | .3139 | .3212 | .3191 | .3085 | .2881 |
| a 200 | .2105 | .2122 | .2114 | .2072 | .1956 |
| a 500 | .1007 | .1020 | .1019 | .1010 | .0982 |
| a 1000 | .0534 | .0537 | .0537 | .0535 | .0532 |

dido en base a los parámetros indicados en cada fila. En primer lugar, una serie de medidas globales: número de documentos devueltos, número de documentos relevantes devueltos (5548 esperados), *R-precision*, precisión media no interpolada para todos los documentos relevantes, precisión media por documento para todos los documentos relevantes, precisión media interpolada en 11 puntos de cobertura y precisión media interpolada en 3 puntos de cobertura. A continuación, se muestra la precisión obtenida por el sistema a los $N$ documentos devueltos.

A la vista de los resultados obtenidos, podemos confirmar la escalabilidad de nuestras aproximaciones de normalización NLP. Sin embargo, Okapi parece sacar mayor partido de nuestras técnicas basadas en términos simples: lematización (`lem`), expansión de consultas mediante sinonimia ponderada (`sin`), y empleo de familias morfológicas (`fam`). En los resultados obtenidos hasta entonces con SMART, sólo la lematización superaba al *stemming* (`stm`); sin embargo, podemos apreciar que en estos nuevos resultados el *stemming* se muestra claramente inferior respecto a las tres técnicas citadas, superando de nuevo únicamente al empleo de dependencias sintácticas (`f-pds`).

Por otra parte, el comportamiento relativo entre sí de las técnicas NLP se mantiene. La lematización (`lem`) sigue mostrándose superior respecto al resto, si bien las diferencias respecto a las otras dos técnicas basadas en términos simples (`sin` y `fam`) se han reducido en parte, sobre todo en el caso de la expansión mediante sinónimos. El empleo de familias (`fam`) sigue mostrándose como el peor en términos de cifras globales, pero perdiendo efectividad respecto a la expansión mediante sinonimia (`sin`) en términos de precisión a los $N$ documentos. Parece, por tanto, que el ruido introducido por la expansión mediante sinónimos tiene menor efecto en el nuevo marco de trabajo.

En lo que respecta al empleo de términos complejos basados en dependencias sintácticas (`f-pds`), sigue mostrando deficiencias respecto a los términos simples, por lo que deberemos seguir profundizando en su estudio.

## 5 Conclusiones y trabajo futuro

En este artículo hemos tratado de ilustrar el camino recorrido por el Grupo COLE en la creación de sistemas de Recuperación de Información para español basados en la aplicación de técnicas de Procesamiento de Lenguaje Natural. Puesto que nuestros sistemas permiten ser aplicados independientemente del motor de indexación utilizado, nuestro interés a la hora de evaluar sistemas de IR en español no es tanto un afán competitivo por obtener mejores resultados que otros equipos, como comprobar la variación de rendimiento inducida al utilizar componentes de NLP con respecto a una *línea base*, en nuestro caso la obtención de términos índice mediante *stemming*. Lo que pretendemos es observar qué técnicas de NLP proporcionan mejores resultados, para que de este modo puedan ser incorporadas en los sistemas de otros grupos, propagando de esta manera la utilización de este tipo de técnicas en sistemas de IR. Sin embargo, cuanto más alto esté el nivel establecido por la línea base de comparación, más difícil se convierte el reto de mejorar esos resultados.

En esta tarea, a la carencia histórica de recursos lingüísticos que sirvan de base y soporte para el desarrollo de herramientas de NLP para el español, en el caso de la investigación sobre IR se ha de añadir la falta de una colección estándar sobre la que realizar la evaluación. Hemos visto cómo este hecho nos obligó a dedicar un considerable esfuerzo a crear una pequeña colección propia sobre la que realizar nuestros experimentos hasta la incorporación en el CLEF del corpus para español, el cual se ha convertido en el estándar de hecho para la evaluación de los sistemas de IR que trabajan en esta lengua. La conclusión más importante que podemos obtener de nuestra experiencia es que la utilización de algunas técnicas de NLP permite incrementar el rendimiento de los sistemas de IR, sin incrementar significativamente el coste computacional. Sin embargo, quedan todavía aspectos por mejorar, y nuevos métodos por investigar. A continuación presentamos las líneas de trabajo futuro más importantes:

- Incrementar la efectividad del etiquetador. Nuestro etiquetador-lematizador fue entrenado con un pequeño corpus de textos periodísticos, de aproximadamente 16.500 palabras, etiquetado manualmente. Próximamente prevemos utilizar para el entrenamiento una parte del corpus Lexesp [31], aquella proveniente de textos periodísticos y que haya sido verificada manualmente.
- Creemos que la efectividad del empleo de familias morfológicas podría verse incrementada de aumentar su grado de corrección. Para ello es preciso revisar manualmente el conjunto actual de familias obtenidas automáticamente, tarea que consumirá mucho tiempo.
- Incrementar la efectividad del analizador sintáctico superficial. Para ello estamos desarrollando actualmente un analizador que trata de aproximar el análisis de gramáticas independientes del contexto mediante la utilización de cascadas de expresiones regulares [1, 19]. Los resultados preliminares son alentadores, ya que muestran un incremento de la precisión general del sistema.
- Estamos estudiando la posibilidad de enriquecer los términos índice con la inclusión de los pares de palabras involucradas en las *colocaciones* que aparecen en textos y consultas [3]. Aunque la definición de colocación es ciertamente controvertida, podemos decir que dos palabras forman una colocación si coocurren regularmente en una lengua y el significado de la aparición conjunta es diferente a la simple suma de significados de las palabras individuales. Por ejemplo, el verbo *dar* suele coocurrir con el sustantivo *respiro*, y *darse un respiro* no es lo mismo que respirar ni quiere decir que nos entreguemos un respiro. El problema que surge con el tratamiento de las colocaciones es que no existen unas reglas fijas que nos indiquen qué palabras coocurren con qué otras, sino que vienen determinadas por el uso de la lengua. Por ello, para extraer colocaciones primero se debe disponer de un conjunto de textos en los cuales se hayan identificado manualmente las colocaciones existentes, para a continuación tratar de extraer ciertos patrones estadísticos que nos ayuden a identificar colocaciones en textos que no hayan sido previamente tratados.

## Reconocimientos

El trabajo mostrado en este artículo ha sido financiado en parte por el Ministerio de Ciencia y Tecnología (proyecto TIC2000-0370-C02-01; acciones integradas HF2002-81 y HP2001-0044), por una beca FPU de la Secretaría de Estado de Educación y Universidades, por la Xunta de Galicia (proyectos PGIDT01PXI10506PN, PGIDIT02PXIB30501PR y PGIDIT02SIN01E) y por la Universidade da Coruña. Queremos también agradecer al Sr. Darrin Dimmick, del NIST, que pusiera a nuestra disposición el sistema ZPrise, y al Sr. Fernando Martínez, de la Universidad de Jaén, su ayuda para hacerlo operativo.

## Bibliografía

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

[2] Miguel A. Alonso, David Cabrero, Eric de la Clergerie, and Manuel Vilares. Tabular algorithms for TAG parsing. In *Proc. of EACL'99, Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 150–157, Bergen, Norway, June 1999. ACL.

[3] Margarita Alonso and Begoña Sanromán. Construcción de una base de datos de colocaciones léxicas. *Procesamiento del Lenguaje Natural*, 24:97–98, September 2000.

[4] Miguel A. Alonso, Jesús Vilares, and Víctor M. Darriba. On the usefulness of extracting syntactic dependencies for text indexing. In M. O'Neill, F. F. E. Sutcliffe, C. Ryan, and M. Eaton, editors, *Artificial Intelligence and Cognitive Science*, volume 2464 of *Lecture Notes in Artificial Intelligence*, pages 3–11. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[5] A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. Linguistically motivated information retrieval. In *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., New York and Basel, 2000.

[6] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley and ACM Press, Harlow, England, 1999.

[7] Fco. Mario Barcala, Eva M. Domínguez, Miguel A. Alonso, David Cabrero, Jorge Graña, Jesús Vilares, Manuel Vilares, Guillermo Rojo, M. Paula Santalla, and Susana Sotelo. Una aplicación de RI basada en PLN: el proyecto ERIAL. In Emilio Sanchís, Lidia Moreno, and Isidoro Gil, editors, *Actas de las I Jornadas de Tratamiento y Recuperación de Información (JOTRI)*, pages 165–172, Valencia, Spain, 2002.

[8] Fco. Mario Barcala, Jesús Vilares, Miguel A. Alonso, Jorge Graña, and Manuel Vilares. Tokenization and proper noun recognition for information retrieval. In A Min Tjoa and Roland R. Wagner, editors, *Thirteen International Workshop on Database and Expert Systems Applications. 2-6 September 2002. Aix-en-Provence, France*, pages 246–250, Los Alamitos, California, USA, September 2002. IEEE Computer Society Press.

[9] J. M. Blecua, editor. *Diccionario Avanzado de Sinónimos y Antónimos de la Lengua Española*. Vox, Barcelona, Spain, 1997.

[10] Chris Buckley. Implementation of the SMART information retrieval system. Technical report, Department of Computer Science, Cornell University, 1985. Source code: `ftp://ftp.cs.cornell.edu/pub/smart`.

[11] Chris Buckley, James Allan, and Gerard Salton. Automatic routing and ad-hoc retrieval using SMART: TREC 2. In D. K. Harman, editor, *NIST Special Publication 500-215: The Second Text REtrieval Conference (TREC-2)*, pages 45–56, Gaithersburg, MD, USA, 1993.

[12] Santiago Fernández, Jorge Graña, and Alejandro Sobrino. A Spanish e-dictionary of synonyms as a fuzzy tool for information retrieval. In *Actas del XI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF-2002)*, pages 31–37, León, Spain, September 2002.

[13] Carlos G. Figuerola, Raquel Gómez, Angel F. Zazo Rodríguez, and José Luis Alonso Berrocal. Stemming in Spanish: A first approach to its impact on information retrieval. In Carol Peters, editor, *Working notes for the CLEF 2001 workshop*, Darmstadt, Germany, September 2001.

[14] Jorge Graña. *Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural*. PhD thesis, Universidad de La Coruña, La Coruña, Spain, 2000.

[15] Jorge Graña, Miguel A. Alonso, and Manuel Vilares. A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. In Petr Sojka, Ivan Kopecek, and Karel Pala, editors, *Text, Speech and Dialogue*, volume 2448 of *Lecture Notes in Computer Science*, pages 3–10. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[16] Jorge Graña, Fco. Mario Barcala, and Miguel A. Alonso. Compilation methods of minimal acyclic automata for large dictionaries. In Bruce W. Watson and Derick Wood, editors, *Implementation and Application of Automata*, volume 2494 of *Lecture Notes in Computer Science*, pages 135–148. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[17] Jorge Graña, Fco. Mario Barcala, and Jesús Vilares. Formal methods of tokenization for part-of-speech tagging. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 240–249. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[18] Jorge Graña, Jean-Cédric, and Manuel Vilares. Integrating external dictionaries into stochastic part-of-speech taggers. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nocolov, and Nokolai Nikolov, editors, *EuroConference Recent Advances in Natural Language Processing. Proceedings*, pages 122–128, Tzigov Chark, Bulgaria, 2001.

[19] Gregory Grefenstette, Anne Schiller, and Salah Aït-Mokhtar. Recognizing lexical patterns in text. In Frank Van Eynde and Dafydd Gibbon, editors, *Lexicon Development for Speech and Language Processing*, volume 12 of *Text, Speech and Language*,

pages 141–168. Kluwer Academic Publishers, Dordrecht/Boston/London, 2000.

[20] Christian Jacquemin. *Spotting and Discovering Terms through Natural Language Processsing*. The MIT Press, Cambridge, Massachusetts, USA, 2001.

[21] Christian Jacquemin and Evelyne Tzoukermann. NLP for term variant extraction: synergy between morphology, lexicon and syntax. In [34], pages 25–74.

[22] Mervyn F. Lang. *Spanish Word Formation: Productive Derivational Morphology in the Modern Lexis*. Croom Helm, Routledge, London and New York, 1990.

[23] David D. Lewis and Karen Sparck Jones. Natural language processing for information retrieval. *Communications of the ACM*, 39(1):92–101, 1996.

[24] Jose Perez-Carballo and Tomek Strzalkowski. Natural language information retrieval: progress report. *Information Processing and Management*, 36(1):155–178, 2000.

[25] Carol Peters, editor. *Results of the CLEF 2002 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2002 Workshop*, Rome, Italy, September 2002.

[26] Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors. *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001*, volume 2406 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[27] Ulrich Pfeifer, Thomas Poersch, and Norbert Fuhr. Retrieval effectiveness of proper name search methods. *Information Processing and Management*, 32(6):667–679, 1996.

[28] M. Porter, and R. Boulton. Object Muscat, an open source search engine. In volume 34 of *ACM SIGIR Forum*. ACM Press, New York, 2000.

[29] S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In E. Voorhees and D. K. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, NIST Special Publication 500-264, pages 151–161, 2000.

[30] J. Savoy, A. Le Calve, and D. Vrajitoru. Report on the TREC-5 experiment: Data fusion and collection fusion. In *Proceedings of TREC'5*, NIST publication #500-238, pages 489–502, Gaithersburg, MD, 1997.

[31] Nuria Sebastián, Fernando Cuetos, María Antonia Martí, and Manuel F Carreiras. *LEXESP: léxico informatizado del español*. Universitat de Barcelona, Bercelona, Spain, 2000.

[32] Klaas Sikkel. *Parsing Schemata — A Framework for Specification and Analysis of Parsing Algorithms*. Texts in Theoretical Computer Science — An EATCS Series. Springer-Verlag, Berlin/Heidelberg/New York, 1997.

[33] Karen Sparck-Jones and Peter Willett, editors. *Readings in Information Retrieval*. Morgan Kaufmann Publishers, Inc., San Francisco, California, USA, 1997.

[34] Tomek Strzalkowski, editor. *Natural Language Information Retrieval*, volume 7 of *Text, Speech and Language Technology*. Kluwer Academic Publishers, Dordrecht/Boston/London, 1999.

[35] Paul Thompson and Christopher C. Dozier. Name recognition and retrieval performance. In [34], pages 261–272.

[36] Jesús Vilares, Miguel A. Alonso, Francisco J. Ribadas, and Manuel Vilares. COLE experiments at CLEF 2002 Spanish monolingual track. In [25], pages 153–160.

[37] Jesús Vilares, Miguel A. Alonso, Francisco J. Ribadas, and Manuel Vilares. COLE experiments at CLEF 2002 Spanish monolingual track. In *Cross-Language Information Retrieval and Evaluation: Results of the CLEF 2002 Evaluation Campaign*, volume 2785 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin-Heidelberg-New York, 2003.

[38] Jesús Vilares, Fco. Mario Barcala, and Miguel A. Alonso. Using syntactic dependency-pairs conflation to improve retrieval performance in Spanish. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2276 of *Lecture Notes in Computer Science*, pages 381–390. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[39] Jesús Vilares, Fco. Mario Barcala, Miguel A. Alonso, Jorge Graña, and Manuel Vilares. Practical NLP-based text indexing. In Francisco J. Garijo, José C. Riquelme, and Miguel Toro, editors, *Advances in Artificial Intelligence — IBERAMIA 2002*, volume 2527 of *Lecture Notes in Computer Science*, pages 635–644. Springer-Verlag, Berlin-Heidelberg-New York, 2002.

[40] Jesús Vilares, David Cabrero, and Miguel A. Alonso. Applying productive derivational morphology to term indexing of Spanish texts. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2004 of *Lecture Notes in Computer Science*, pages 336–348. Springer-Verlag, Berlin-Heidelberg-New York, 2001.

[41] Jesús Vilares, Manuel Vilares, and Miguel A. Alonso. Towards the development of heuristics for automatic query expansion. In Heinrich C. Mayr, Jiri Lazansky, Gerald Quirchmayr, and Pavel Vogel, editors, *Database and Expert Systems Applications*, volume 2113 of *Lecture Notes in Computer Science*, pages 887–896. Springer-Verlag, Berlin-Heidelberg-New York, 2001.