Enriquecimiento parcial de la consulta para la recuperación de información de alta calidad y flexibilidad

Alexander Gelbukh

Centro de Investigación en Computación (CIC),
Instituto Politécnico Nacional (IPN),
Av. Juan de Dios Bátiz s/n esq. Mendizábal,
Col. Nueva Industrial Vallejo, CP 07738, DF, México

www.gelbukh.com

Resumen. En la recuperación de información con alta calidad se tiene que tratar con los fenómenos de la equivalencia no literal de cadenas de letras, desde la inflexión y derivación morfológica de las palabras (*dormir*, *duermo*, *durmió*; *dormitorio*) hasta las relaciones semánticas de varios tipos entre las palabras (*computadora*, *informática*, *algoritmo*). En todos estos casos, es importante que dada una de estas palabras en la petición del usuario, se encuentren los documentos que contienen otras palabras del mismo conjunto. Por otro lado, para ciertas aplicaciones es importante que el usuario tenga buen control sobre qué palabras el sistema considera como equivalentes para cada petición específica. Técnicamente, el fenómeno se puede manejar en el momento de la indización de la base de documentos o en el momento de aplicación de la petición del usuario. En este trabajo explicamos por qué conviene tratar el problema a través de la expansión de la petición del usuario, y proponemos una técnica que drásticamente reduce el tamaño de la petición así obtenida.

1 Introducción

En nuestra sociedad de información, los procesos de manejo de conocimiento y en particular de recuperación de información textual juegan un papel crucial. Actualmente tenemos tanta información disponible en formato electrónico que encontrar los documentos que nos interesan se vuelve en un problema técnico muy complejo. En particular, los usuarios usualmente no saben con precisión qué palabras puede contener el documento que les puede intere-

sar, y sólo pueden formular su necesidad informática en palabras generales que describen el tema de su interés pero no un documento específico que buscan.

Consecuentemente, cualquier sistema de recuperación de información debe tratar de alguna manera el problema de la comparación aproximada entre las palabras clave de la petición de búsqueda del usuario y el documento. Por ejemplo, dado la petición computadora, el sistema debe ser capaz de recuperar (o no recuperar, ya dependiendo de los ajustes definidos por el usuario y las opciones de la petición del usuario) los documentos que contengan las cadenas de letras Computadora (con mayúscula), computadoras, computación, monitor, algoritmo, Internet, etc. Hay dos momentos en el flujo del procesamiento de datos por el sistema de recuperación de información donde este problema puede ser tratado. Por un lado, se puede tratar en el momento de la indización de los documentos, es decir, con el enriquecimiento del índice. Alternativamente, puede ser tratado en el momento del procesamiento de la consulta (así llamaremos la petición del usuario) específica, es decir, a través del enriquecimiento de la consulta.

La primera técnica se utiliza con más frecuencia debido a que los problemas de la segunda parecen prohibitivamente graves. Específicamente, en la segunda técnica, el enriquecimiento de la petición del usuario, en parecer requiere agregar a la petición millones de palabras relacionadas con las palabras indicadas por el usuario.

En este trabajo demostramos, sin embargo, que el primer método tiene sus propias desventajas, y que por otro lado los problemas del último método se pueden resolver de manera computacionalmente eficiente. Nuestra principal motivación en este trabajo ha sido el desarrollo de un sistema de recuperación de información para corpus reducidos. Específicamente, nuestro interés fue el desarrollo de la metodología de acceso computacional y recuperación de documentos del Corpus diacrónico y diatópico del español de América, denominado CORDIAM[1], el cual está en desarrollo en el marco de una colaboración entre la Academia Mexicana de la Lengua y un número considerable de instituciones lingüísticas de los países latinoamericanos. Este corpus consiste de la transcripción de los documentos históricos escritos en diferentes países y en diferentes tiempos que comprenden el período de los siglos XVI a XIX.

También hemos experimentado con el corpus de la legislación mexicana del Senado de la República Mexicana. La base del documento del Senado contiene las leyes de la República Mexicana, los proyectos de ley en estudio en las comisiones del Senado, los protocolos de las sesiones, los discursos de los senadores, etc.

[1] Véase www.CORDIAM.Gelbukh.com/acerca-de.

Sin embargo, consideramos que las técnicas expuestas en este artículo se aplican a ortos corpus y otras situaciones similares en la tarea de la recuperación de información. Otra motivación muy importante para nuestro trabajo son los sistemas de búsqueda dentro de la computadora, tales como *Microsoft Desktop Search* o el descontinuado sistema *Google Desktop Search*. Este tipo de sistemas proporcionan a los usuarios la búsqueda dentro de los archivos personales guardados en el disco duro de la computadora, documentos personales, correo y otras aplicaciones. Como ejemplo se puede considerar la función de búsqueda incluida en el programa de correo Outlook, parte de Microsoft Office. Esta función se beneficiaría mucho de las mejoras que proponemos en este artículo, ya que no proporciona nada parecido a éstas. Igual como en el caso de los corpus lingüísticos y jurídicos, la tarea de búsqueda dentro de la computadora personal muestra los rasgos que la hacen apropiada para nuestra metodología.

A saber, en el desarrollo de nuestra metodología nos guiamos por las siguientes prioridades, o sea, desiderata para el método de la recuperación de información a desarrollar [4]:

- La calidad, la fuerza expresiva y la flexibilidad de la búsqueda eran nuestras prioridades principales debido a la importancia de los resultados de la búsqueda para los usuarios, tales como los investigadores de la lingüística diacrónica o los abogados y los senadores. A diferencia de las tareas más comunes de la recuperación de información, en tales casos de uso cada documento es importante.
- Tamaño del índice reducido y razonablemente baja carga de mantenimiento en el servidor eran nuestras segundas prioridades, de acuerdo a los recursos computacionales disponibles para el sistema y el número de consultas que se espera.
- Cambios mínimos requeridos en la estructura de bases de datos de los corpus y en el mantenimiento de estas bases de datos que.
- El sistema debe estar en funcionamiento mientras que los diccionarios y los recursos lingüísticos correspondientes están en la fase de desarrollo y las mejoras y correcciones a los mismos debe surtir efecto de inmediato sin repetir el proceso de indización de toda la base de datos.
- La eficiencia computacional de procesamiento de una sola consulta es de menor prioridad ya que el número de usuarios esperado fue menos de él de los sistemas de recuperación de información en Internet.

Las propiedades características de la base de documentos para la cual se aplica la metodología propuesta son las siguientes:

- Tamaño mediano a grande, en el orden de un gigabyte, es cual puede extenderse a varios gigabytes,
- Contenidos especializados con variedad limitada de léxico y sintáctico,
- Construcciones lingüísticas sin restricciones con la posibilidad de uso ocasional de casi cualquier palabra o forma morfológica de palabra.

En este trabajo, estamos interesados en una solución flexible y computacionalmente eficiente, de implementación simple y mantenimiento fácil, al problema de la adaptación no literal de la petición de búsqueda de usuario en virtud de los requisitos y las circunstancias descritos más arriba.

El artículo está organizado de la siguiente manera. En la sección 2 damos una discusión muy breve del estado del arte en el problema abordado. En la sección 3 consideramos a detalle varios tipos de la comparación no literal de las cadenas de caracteres. En las secciones 4 y 5 discutimos dos posibles aproximaciones al problema de la comparación no literal entre la petición del usuario y el documento: la expansión del índice y la expansión de la consulta, así como sus ventajas y desventajas, llegando a la conclusión de que el enriquecimiento de la consulta tiene ventajas importantes sobre el enriquecimiento del índice. En la sección 6 proponemos un tratamiento del enriquecimiento de la consulta que subsana sus desventajas existentes. En las secciones 7 y 8 describimos la implementación de nuestro sistema y los resultados experimentales obtenidos con ella. Finalmente, en las secciones 9 y 10 discutimos el trabajo futuro —el cual consiste en la combinación de las dos aproximaciones— y presentamos las conclusiones.

2 Estado del arte

Existe un gran cuerpo de literatura sobre la modelación computacional del lenguaje natural [17] y en específico sobre la comparación no exacta de cadenas de caracteres. En la literatura existente se usa para esta tarea una gran variedad de estructuras de datos, tales como los denominados *tries*, los árboles B, etc. [1, 8, 11]. Estos trabajos se basan en patrones implícitos o explícitos que describen la similitud entre la cadena de origen y las cadenas con las cuales ésta se compara (por ejemplo, la distancia mínima de edición) o el conjunto de las cadenas que se buscan (por ejemplo, las expresiones regulares), a nivel de las letras individuales. Por ejemplo, un patrón de *com** (donde el asterisco indica una cadena arbitraria de caracteres) se utiliza para buscar todas las formas de la palabra española *comer*, aunque este patrón también coincidirá con al menos 172 otras palabras en español tales como *cometa*.

Sin embargo, en nuestro caso consideramos el problema de la coincidencia de conjuntos arbitrarios de palabras que pueden no compartir ningún patrón simple de letras. Por ejemplo, las palabras *dormía*, *duermo* y *durmiendo* son formas del mismo verbo español *dormir*. Las cadenas de caracteres, *iglesia*, *sacerdote* y *peregrino* representan el mismo concepto *la religión* a pesar de que no existe ningún patrón específico de letras que se pueda usar para identificar estas cadenas en el texto.

El problema de la generación y coincidencia de las formas de palabras en varios idiomas, incluyendo el español es muy bien estudiado en la lingüística. Varios métodos y estructuras de datos se han sugerido en la lingüística computacional para el manejo de los diccionarios correspondientes y las listas de morfemas [3, 12, 13]. Sin embargo, en este artículo no discutimos los problemas lingüísticos de la morfología del lenguaje natural, sino estamos interesados en la aplicación de un analizador morfológico a la tarea relacionada puramente con la gestión de bases de datos, a saber, la tarea técnica de la recuperación de una palabra clave en todas sus formas morfológicas.

En este trabajo suponemos que la lista de las formas de las palabras ya es conocida, es decir, es generada por un analizador morfológico previamente desarrollado, mientras que estas formas no necesariamente admiten un patrón simple de letras para su descripción.

Por otro lado, el uso de las llamadas ontologías y jerarquías de conceptos para el análisis de documentos es también una tarea muy bien estudiado. La tarea de análisis temático de documentos con el uso de ontologías se estudia en [9], y las técnicas correspondientes se aplican a las tareas de recuperación de información en [7]. Han sido recopilados varios tesauros jerárquicos grandes [2, 10, 14].

De manera similar al tratamiento de la coincidencia morfológica de cadenas de letras, aquí no nos interesa el manejo de los pesos estadísticos de la relación semántica entre conceptos ni la compilación del diccionario de conceptos sino la forma en que los documentos relevantes para un nodo específico de la jerarquía semántica pueden en la práctica ser encontrados en un sistema de información grande existente. Este tema no ha sido cubierto por la literatura sobre la construcción de las ontologías ni los lenguajes de descripción de las mismas.

3 Tipos de coincidencia aproximada de cadenas de caracteres

En esta sección analizamos con más detalle los tipos de cadenas de caracteres los cuales los usuarios necesitan coincidir en los procesos de la búsqueda

por la necesidad informática expresada de manera aproximada o general en la petición de búsqueda. Un punto importante en cada caso es el alto grado de flexibilidad necesaria para satisfacer las necesidades del usuario específico o efectuar una búsqueda específica.

3.1 Coincidencia entre mayúsculas y minúsculas

Es el tipo más simple de la coincidencia no literal de cadenas, y pudiera parecerse que no representa ningún problema para su tratamiento en un sistema computacional. Por ejemplo, usualmente las cadenas tales como *computadora, Computadora*, *COMPUTADORA* y *ComPuTaDoRa* deben ser consideradas como equivalentes. Los diseñadores de los sistemas de recuperación de información tienden que pensar que es obvio que antes de la indización de la base de datos todas las palabras se convertirían automáticamente a, por ejemplo, minúsculas.

Sin embargo, en determinadas circunstancias el usuario podría querer buscar una cadena específica, como *Ángel*, *Victoria*, *Gigante, PAN* (nombres de personas, topónimos, nombres de empresas), pero no *ángel*, *victoria*, *gigante, pan*, etc. O bien, un usuario puede, al revés, estar interesado en la palabra *pan* y no *PAN*.

Sin embargo, los motores de búsqueda actuales los cuales no hacen diferencia en su índice entre mayúsculas y minúsculas no proporcionan la posibilidad de especificar las mayúsculas y minúsculas en la petición de búsqueda —ni siquiera Google proporciona tal posibilidad. En nuestro caso, especialmente cuando se trata de los corpus lingüísticos tales como el CORDIAM, esta aproximación no es apropiada, dado que para los investigadores lingüistas es importante distinguir el uso de las mayúsculas y minúsculas en el corpus. Más específicamente, a veces es importante distinguirlas y a veces, al revés, esta diferencia se debe ignorar, dependiendo de las necesidades del usuario específico y la consulta específica. Con esto es claro que el tratamiento de las mayúsculas en nuestra metodología se debe posponer hasta la fase de búsqueda y no puede efectuarse en la fase de la indización de la base de datos.

3.2 Morfología y formas de palabras

La segunda clase importante de cadenas de caracteres las cuales con frecuencia —aunque no para cualquier búsqueda— se consideran equivalentes son las palabras formas del mismo lexema: *computamos*, *computábamos, computaremos,* o sus variantes derivativas: *computadora*, *computación, computacional, computabilidad*.

En las aplicaciones prácticas computacionales esta equivalencia se indica por el software lingüístico llamado analizador morfológico [12, 13]. Para cada palabra el analizador morfológico genera (posiblemente con ambigüedad) un identificador, por ejemplo, la primera forma, como *computar,* una base o raíz, como *comput-*, un número identificador, etc. Tal identificador es común para todas las formas morfológicas de la palabra, las cuales se consideran equivalentes para la búsqueda específica si el usuario así lo indica en las opciones de la búsqueda. Entonces la comparación aproximada de las dos cadenas de caracteres consiste en la reducción —llamada también *normalización*— de ellos a tales identificadores y luego la comparación exacta —ya no aproximada— de los resultados de tal reducción.

Existen analizadores morfológicos de diferente grado de complejidad, dependiendo de la precisión deseada y de si sólo se toma en cuenta el paradigma de inflexión dentro de un sólo lexema (*computamos / computaban*) o se debe tomar en cuenta también la derivación morfológica (*computar /computadora*) o semántica (*computadora / informática*). Algunos de los analizadores morfológicos dependen de la lengua específica de los documentos y otros son independientes del lenguaje.

En el caso simple, un analizador morfológicos utiliza una lista de terminaciones, tales como: *-ar, -amos, -ábamos, -adora, -abilidad* y una pequeña lista de excepciones, tales como *ir / fue*. Para los idiomas altamente inflexivos —como es el español— la lista de terminaciones puede ser considerablemente grande. Por ejemplo, en el sistema que hemos desarrollado en el marco de este trabajo en la actualidad se usan 3 mil 451 terminaciones.

Un analizador morfológico más sofisticado —y por lo tanto más preciso— utiliza patrones complejos y posiblemente depende de un diccionario general. Sin embargo, incluso un sistema basado en un diccionario debe contener un algoritmo heurístico para el manejo de las palabras ausentes en su diccionario, entre éstas, términos especiales, neologismos, nombres propios, etc.

Se debe tener en cuenta que los algoritmos morfológicos basados en heurísticas funcionan mucho mejor en el análisis —normalización— que en la síntesis —generación— de las formas de palabras correspondientes a una base morfológica dada. Por ejemplo, un simple algoritmo basado en lista puede normalizar *incomputabilidad* a *-comput-*, sin embargo, dado una base *-comput-*, es difícil para un sistema computacional no sofisticado elegir entre las variantes **incomputibildad*, **descomputabilidad,* etc.

Coincidencia de las formas morfológicas de la misma raíz en la aplicación de la petición de búsqueda no siempre es deseable para el usuario. Por ejemplo, el usuario puede estar interesado en las *computadoras*, pero no en la *computabilidad*. La reducción morfológica de las formas ambiguas, especialmente en

idiomas altamente inflexivos como el español, a veces produce efecto muy molesto. Por ejemplo, el verbo español *comer* tiene más de setecientas formas morfológicas, tales como *comiste* o *comiéndotelas*, una de los cuales, a saber, *como*, coincide con una conjunción muy frecuentemente utilizada *como*. Por lo tanto, si el usuario quiere encontrar los documentos con el lexema español *comer* con una precisión razonablemente alta, tendrá que sacrificar un poco el *recall* (recuerdo, el porcentaje de los documentos relevantes para el usuario que el sistema realmente encuentra) formulando la consulta como "buscar todas las formas de la palabra *comer,* pero no la forma como".

Nótese que este efecto no se puede lograr con una expresión lógica tan simple como "todos los documentos que contienen el lexema *comer* en cualquiera de sus formas morfológicas, excepto los que contienen el lexema *como*", ya que su significado no es equivalente a la búsqueda deseada. A saber, el *recall* de una consulta de este tipo sería extremadamente bajo ya que casi cualquier documento en español efectivamente contiene la cadena de caracteres *como*.

De este modo, otra vez llegamos a la conclusión de que el usuario debe poder controlar el hecho y el grado de la aplicación de la normalización morfológica que es aplicada a su consulta de búsqueda por el sistema —lo que los buscadores existentes tales como Google o *Microsoft Office Search 2013* no permiten. En particular, no será adecuado aplicar la reducción morfológica al momento de la indización de la base de documentos, como lo hacen los motores de búsqueda existentes.

3.3 Ontología o jerarquía de conceptos

La tercera clase de las palabras que el usuario podría necesitar que se consideren equivalentes son los sinónimos y variantes dialectales (*computadora / ordenador*), los hipónimos e hiperónimos (*computadora / dispositivo*) y posiblemente las palabras unidas por otras relaciones semánticas, tales como meronimia / holonimia, etc. Dado que ningún algoritmo puede inferir dichas relaciones entre las palabras por las correspondientes cadenas de caracteres, se utiliza un diccionario —por ejemplo, una ontología o un tesauro jerárquico— para proporcionar esta opción al usuario.

En nuestros experimentos hemos utilizado un diccionario de 33 mil palabras organizadas en una jerarquía profunda de conceptos, similar al tesauro de Roget [2] y en parte derivada de él. En cuanto a los conceptos relacionados, nos referimos no sólo a los conceptos con la relación de hiponimia (llamada *es-un*), sino también a otras palabras que son de interés para el usuario para la búsqueda de los documentos sobre un tema dado [9]. Por ejemplo, la entrada del diccionario para *religión* contiene tales palabras como *Biblia, sacerdote,*

rezar, iglesia, peregrino, etc. Por lo tanto, al usuario que busca los documentos sobre la *religión* el sistema le ofrece también los documentos que mencionan la *Biblia*. Opcionalmente, para medir la relevancia del documento encontrado el grado de dicha relación entre las palabras y los tópicos puede ser ponderado cuantitativamente [7].

Para este fin también se puede utilizar otros diccionarios y ontologías, siendo WordNet la primera opción que viene a la mente. Sin embargo, WordNet no contiene las relaciones semánticas tan ricas como el tesauro de Roget.

Obviamente, el usuario debe tener control total sobre el conjunto de las palabras que deben considerarse equivalentes a la palabra clave o a las palabras clave de la consulta. Las opciones mostradas en la ilustración 1 son de interés particular.

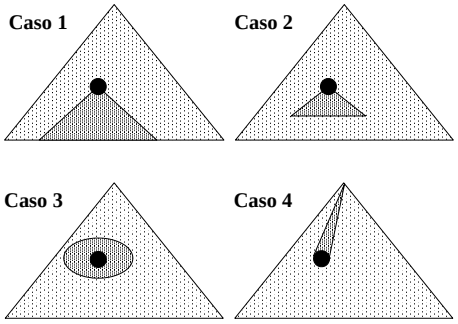

Ilustración 1. Los tipos de vecindades en una ontología.

En esta ilustración se muestran diferentes tipos de vecindades en una jerarquía semántica. La jerarquía se muestra en forma de triángulo, donde la punta simboliza la raíz de la jerarquía y la base simboliza los conceptos del más bajo nivel, siendo los puntos debajo de un punto dado sus hipónimos, los puntos al lado de él sus hermanos —cohipónimos— y los puntos arriba de él sus hiperónimos. Los cuatro tipos de vecindades se pueden caracterizar como sigue.

El caso 1 representa **todas las instancias** (hipónimos) de un concepto dado, es decir, todas las palabras por debajo de un nodo dado en la jerarquía. Por ejemplo, si el usuario elije este tipo de consulta, dada una consulta "¿qué hay de *matemáticas*?", el sistema recupera todos los documentos sobre *álgebra, geometría, cálculo* y dentro de estos temas, respectivamente, todo sobre la *álgebra lineal*, *teoría de grupos*, etc., *geometría diferencial*, *teoría de foliación*, etc., *cálculo diferencial*, calculo integral, etc., y finalmente recupera los documentos que mencionan las palabras *vector, variedad, diferencial,* etc. Otro ejemplo: ¿qué eventos ocurrieron en *Europa*? Esta consulta se debe interpretar

en de tal manera que se recuperen los documentos que citan *Inglaterra, Italia, Austria,* etc., *Londres, Manchester, Birmingham,* etc. En la ilustración 1, el conjunto correspondiente de palabras se muestra como todos los puntos debajo del punto indicado en la consulta, en este caso *Europa*.

El caso 2 representa los **hipónimos casi inmediatos** de un concepto dado, es decir, las palabras de abajo de un nodo dado, pero no más allá de un número n de niveles. Por ejemplo, un estudiante puede querer saber lo que son las matemáticas: "Encontrar documentos sobre las *matemáticas* en general". En este caso, el sistema recuperará los documentos que mencionan las palabras como *algebra* o *geometría*, luego *ecuación*, *desigualdad*, *teorema*, pero no *isotropía* ni *semirretículo*, ya que estas últimas palabras son demasiado especializadas para ser útiles para una consulta general. Otro ejemplo: "¿Qué hay de la política de los *países europeos*?" En este caso, se deben recuperar sólo los documentos que mencionan a *Inglaterra, Italia, Austria,* etc. y, probablemente, *Londres, Roma, Viena,* pero no *Birmingham*, *Manchester*, *Wolverhampton*, etc.

El caso 3 representa los **conceptos similares,** es decir, las palabras que se encuentran en el árbol de conceptos no más de n pasos de la palabra dada, ya sean los pasos para abajo (hipónimos), para arriba (hiperónimos), o en la dirección horizontal en el árbol (cohipónimos) [5]. Por ejemplo: "¿Qué disciplinas son similares a las *matemáticas*?" En este caso, las palabras relevantes son tanto *física* y *astronomía*, como *álgebra* y *geometría* o incluso *ciencia*, etc.

El caso 4 representa los **conceptos más generales,** es decir, hiperónimos: las palabras de las cuales el nodo dado es una instancia o un hipónimo. Por ejemplo: "¿En qué continente se encuentra *Morelia*?". En este caso, los documentos que podrían dar idea de este continente son los que contienen las palabras *Michoacán, México* o *América del Norte* (siendo Morelia una ciudad en el estado de Michoacán, México). Otro ejemplo: "¿Cuales derecho tiene un *Profesor Asociado*?". En este caso, los documentos útiles son los que mencionan los derechos de un *maestro*, *empleado*, *ciudadano*, *persona* o *humano*.

De hecho, las limitaciones indicadas en los casos de 1 a 4 a menudo tienen que combinarse. Por ejemplo, en una consulta de tipo 4, es útil imponer un límite sobre el número de niveles —como en la consulta de tipo 1— o sobre el nivel más general, ya que los conceptos demasiado generales aparecen en muchos documentos poco relevantes para una consulta específica y además es poco probable que provean conocimiento nuevo al usuario. O bien, las consultas de tipo 4 se pueden combinar con las restricciones de tipo 1 o 2. Por ejemplo, para la consulta "¿En qué continente se encuentra *Michoacán*?" se deben buscar tanto los conceptos más generales (como en el tipo 4) como los conceptos más específicos (como en el tipo 1).

Nótese que, al menos en la actualidad, el tipo deseado de la generalización de consultas no puede deducirse automáticamente por el sistema y debe ser elegido explícitamente por el usuario.

4 Enriquecimiento del índice

En la sección anterior hemos hablado de cuatro casos de la comparación aproximada de las cadenas de caracteres: con ignorar la diferencia en las mayúsculas y minúsculas, en las formas morfológicamente declinadas, en sinonimia y otras relaciones léxico-semánticas, y tomado en cuenta la estructura de un árbol de conceptos (las consultas de tipo 1 en la ilustración 1). Una aproximación ingenua —y la más utilizada— para representar los tres primeros casos de la comparación aproximada es *la reducción* del *índice*: al momento de la indización, todas las letras se reducen, por ejemplo, a minúsculas, todas las formas de las palabras o los conceptos derivados se reducen a la forma principal (*computando, computamos, computan, computación, computadora* → *computación*) y sinónimos y variantes dialectales son reemplazados por un representante elegido (*ordenador* → *computadora*). El último caso —un árbol de conceptos— puede ser manejado, además, por la indización de cada documento con los hiperónimos de las palabras las cuales este documento contiene (*PC* → *computadora, dispositivo, artefacto*). Con este método, con la consulta "*dispositivos*" se recuperará también *PC*.

Sin embargo, en este artículo proporcionamos los argumentos a favor de que esta aproximación simplista tiene serias desventajas. En primer lugar, como hemos mostrado en la sección anterior, en función de la relación de precisión vs. *recall* deseada, el usuario puede *no* querer que estas cadenas de caracteres se consideren idénticas. Por lo tanto, el proceso de indización no debe causar ninguna pérdida de información —es decir, todas las cadenas distintas de letras deben aparecer en el índice tal cual, sin cambio alguno, incluso en el caso de mayúsculas vs. minúsculas —es decir, las palabras no se deben reducir a minúsculas.

Para lograr esto, las cadenas reducidas a las letras minúsculas, reducidas morfológicamente, o bien reducidas con un diccionario de sinónimos deben aparecer en el índice además de las secuencias de letras originales y no en lugar de ellas. Por ejemplo, la palabra *Computadoras* debe causar la inclusión en el índice de las cadenas de letras *Computadoras, computadoras, computadora, dispositivo, artefacto*, etc. Dado que las nuevas palabras clave se agregan al índice en lugar de sustituir las palabras originales, este proceso se llama enriquecimiento del índice.

Para permitir cierta flexibilidad de las consultas, se pueden sugerir algunas mejoras a este esquema de indización. Por ejemplo, las palabras claves que se adicionaron se deben marcar de alguna manera para distinguirlas de las palabras originales. Por ejemplo, *Computadoras* → ORIGINAL: *Computadoras,* MINÚSCULAS: *computadoras,* MORFOLOGÍA: *computadora,* HIPERÓNIMO: *dispositivo,* HIPERÓNIMO: *artefacto*, etc.

Con esto, una consulta del usuario "exactamente *Computadoras*" se puede traducir internamente por el sistema a la consulta ORIGINAL: *Computadoras*; la consulta "la forma de palabra *computadoras*" se traduce a MINÚSCULAS: *computadoras*, lo que detecta tanto *Computadoras* como *computadoras* o *COMPUTADORAS*; la consulta "el lexema *computadora*" se traduce a MORFOLOGÍA: *computadora*, lo que detecta tanto *computadora* como *computadoras*. La consulta de tipo 1 para "*dispositivos*" se traduce a HIPERÓNIMO: *dispositivo*, lo que coincide tanto con *la unidad central* como con *impresora*. Otras mejoras serán presentadas en la sección 9 más abajo.

Sin embargo, el método de enriquecimiento del índice presenta algunos problemas, los cuales discutimos a continuación.

Primero, **falta de flexibilidad**. Sólo se pueden ejecutar los tipos de consultas para las que el índice fue diseñado específicamente. El usuario no puede elegir que las palabras de un determinado conjunto deben considerarse equivalentes, por ejemplo, "todas las formas de la palabra *comer* excepto *como* — véase la discusión de este ejemplo en la sección 3.2.

Segundo, el **índice más grande**. A diferencia de la reducción de índice, el método de enriquecimiento del índice puede aumentar el tamaño del índice muy significativamente —de dos veces a diez veces, dependiendo de la utilización de sólo morfología o también de un diccionario de sinónimos. En muchos casos, especialmente con bases de datos grandes, esta opción no es aceptable.

Tercero, **dificultad de mantenimiento**. El método del enriquecimiento del índice implica un demasiado fuerte acoplamiento del proceso de indización con el potencialmente complejo *lingware*, el software lingüístico tal como el analizador morfológico o el diccionario de sinónimos. Esto presenta al menos dos problemas técnicos y de organización.

Por un lado, el incluir el motor de búsqueda inteligente en una tecnología estable de mantenimiento de base de datos operativa que ya existe por un largo tiempo requiere cambios significativos en ésta última, lo que implica el cambio de software y la documentación existente, la formación o actualización de los ingenieros de mantenimiento, etc. A diferencia de esto, siempre es preferible conservar intacta la tecnología existente.

Por otro lado, a diferencia de los procedimientos estables de mantenimiento de las bases de datos, un *lingware* complejo basado en diccionarios tiende a

estar en constante desarrollo, al menos por un determinado período de tiempo inicial: nuevas palabras se añaden al diccionario, las tablas y algoritmos morfológicos se corrigen, se añaden nuevos enlaces al diccionario de sinónimos, etc. Ya sea con la reducción o enriquecimiento del índice, cada vez que se realiza un cambio en el *lingware*, toda la base de datos se va a volver a ser indizada. En muchos casos esto no es factible, sobre todo cuando el procesamiento lingüístico es lento y consume muchos recursos. Por otra parte, el posponer la reindización de la base durante mucho tiempo desalienta en gran medida cualquier mejora a *lingware*, desde el punto de vista tanto de los desarrolladores como los usuarios.

Estos argumentos nos indican que en el mejor caso el índice debe contener exactamente las cadenas de caracteres originales tal cual ocurren en los textos a indizar, sin alteración alguna.

5 Enriquecimiento de la consulta

Una alternativa viable al tratamiento de la comparación aproximada de las cadenas de caracteres al momento de la indización es su tratamiento en el momento del procesamiento de consultas de usuario.

Una aproximación ingenua a este método es la siguiente. Las cadenas de letras que se encuentran en los documentos se indizan tal cual, sin ninguna modificación. Luego, en el momento del procesamiento de la consulta del usuario, esta consulta se sustituye de forma automática por una expresión lógica apropiada. Por ejemplo, la consulta "*computar* y *matriz*" se ejecuta internamente como "(*computación* O *calcular* O *calculadas* O *informática*) Y (*matriz* O *matrices* O *matrices*)", donde O y Y son los operadores lógicos correspondientes. Este procedimiento se llamamos enriquecimiento de la consulta.

Este método no presenta ninguno de los problemas mencionados en el apartado anterior. Es decir, tiene las siguientes ventajas sobre el enriquecimiento o reducción del índice.

Primero, **flexibilidad**. El usuario puede editar la expresión resultante (por ejemplo, marcando o desmarcando las casillas de verificación junto a cada parte de la fórmula generada) para alcanzar cualquier combinación deseada. Por ejemplo, la consulta "todas las formas del verbo español *comer* excepto *como*" de forma fácil y natural puede ser expresada por el usuario y procesada por el sistema.

Segundo, el **índice más pequeño** en comparación con el enriquecimiento del índice. Sólo las cadenas literalmente presentes en el documento están presentes en el índice.

Tercero, **fácil mantenimiento**. El proceso de indización es trivial y no incluye ningún *lingware* ni depende de ningún *lingware*. No hacen falta ningunos cambios en la tecnología de indización no inteligente ya existente cuando se añade un motor de búsqueda inteligente a una base de datos ya operacional. No hace falta reindización cuando se realizan cambios en el *lingware*, y al mismo tiempo tales cambios están disponibles al usuario inmediatamente.

Sin embargo, las desventajas de este enfoque ingenuo son tan obvias que éste no se puede considerar como una alternativa viable en práctica. Los siguientes dos problemas hacen que tal método no sea factible.

Primero, **consultas demasiado grandes**. Como ya hemos mencionado, el verbo español *comer* tiene alrededor de 700 formas morfológicas, lo que resulta en la consulta enriquecida demasiado grande. Con un diccionario de sinónimos, el concepto *Europa* contribuiría a la consulta todos los países, ciudades, ríos, montañas, naciones, tipos de alimentos, etc. específicos para Europa. Además, cada una de estas cadenas debe capitalizarse de todas las maneras posibles.

Segundo, este método involucra la **generación lingüística**. Como ya hemos mencionado en la sección 3.2, la generación de todas las formas y derivadas semánticas de un lexema dado (*computar* → *computar*, *computación, incomputable*, *incomputabilidad*, etc.) es una tarea mucho más difícil que la tarea de análisis, es decir, de adivinar la forma principal correcta o la raíz de una determinada forma de la palabra (*computar*, *computación, incomputable*, *incomputabilidad* → *computar* o -comput-). En caso de un algoritmo morfológico basado en heurísticas, la cantidad de hipótesis en la generación de formas es usualmente mucho mayor que en la reducción de las palabras a la raíz morfológica.

En la siguiente sección presentamos cómo estos problemas pueden ser resueltos de manera práctica.

6 Enriquecimiento parcial de la consulta

La mejora que aquí sugerimos para el método de enriquecimiento de la consulta consiste en incluir en la consulta enriquecida sólo las cadenas que están presentes en al menos un documento de la base de datos sobre la cual se ejecuta la consulta. Ya que la diversidad lingüística en una base documental especializada (por ejemplo de textos jurídicos, médicos, etc.) es relativamente baja, sólo una pequeña fracción de todas las formas posibles de una palabra o subconceptos del concepto están presentes en la base de datos dada, lo que reduce en gran medida el tamaño de la consulta enriquecida. Al mismo tiem-

po, cuando se aplica a la base de datos específica, tal consulta parcialmente enriquecida es totalmente equivalente a la consulta enriquecida completa. Llamamos a esta modificación del procedimiento de enriquecimiento de enriquecimiento parcial.

El proceso de enriquecimiento parcial de consulta puede esbozarse de la siguiente manera.

Primero, se compila una lista de todas las cadenas que aparecen al menos una vez en la base de datos dada.

Segundo, esta lista relativamente pequeña es indizada como se describe en la sección 4, lo que produce una tabla de índices tales como los siguientes:

Cadena	Identificador
computadora	computadora
Computadora	computadora
COMPUTADORA	computadora
computación	computar
computable	computar
incomputabilidad	computar
PC	computadora
PC	dispositivo
PC	artefacto

Aquí, por el identificador nos referimos a una forma reducida, como en el caso de la reducción a las minúsculas, reducción morfológica a la forma principal, promoción vertical por el árbol en el diccionario de sinónimos e hiperónimos, etc., según lo descrito en la sección 3 (no mostramos en esta tabla las mejoras discutidas en las secciones 4 y 9).

Tercero, al momento de procesamiento de la consulta, cada palabra clave de la consulta es sujeto a un proceso de indización apropiado dependiendo de la opción definida por el usuario. Por ejemplo, éste puede ser la reducción morfológica a su forma principal, como en *calculable* $\rightarrow$ *calcular*, convirtiendo así la palabra a su potencial identificador. En caso de ambigüedad, se obtienen todos los identificadores potenciales.

Cuatro, los identificadores obtenidos para cada palabra clave de la consulta se buscan en la columna derecha de la tabla, y la palabra se sustituye con la lista de las cadenas correspondientes literales que se encuentran en la columna izquierda de la misma tabla.

Por ejemplo, con la tabla anterior, la pregunta "¿qué cosas son *computables*?" se transforma primero en la consulta "Identificador = computar" y después de la búsqueda en la tabla, en la consulta "*computación* O *computable* O *incomputabilidad*". Nótese que esta consulta parcialmente enriquecida no contiene tales cadenas de caracteres como *computar* o *computándose*. Ni siquiera contiene la forma de palabra *computables* de la misma consulta, ya que tal forma no se presenta en los documentos de la base de datos de nuestro ejemplo.

Para procesar una consulta compleja, tal como, por ejemplo, una consulta de tipo 3 como se discutió en la sección 3.3, el tesauro se recorre de manera correspondiente y la nueva consulta se construye primero como una disyunción de los lexemas o conceptos relevantes como se muestra en la ilustración 1. Luego la consulta de este tipo se enriquece aún más, o bien se filtra —lo cual puede incluso reducir su tamaño—, a través de la tabla de índice, como se describe más arriba.

La modificación que proponemos para el método de enriquecimiento de consulta no presenta ninguna de las desventajas del método original, y además presenta las siguientes ventajas en comparación con el enriquecimiento completo de consulta.

Primero, **consultas más pequeñas**. Sólo las palabras que realmente aparecen en la base de datos se incluyen en la consulta construida. La diferencia es especialmente sensible en el caso de las lenguas morfológicamente ricas tales como el español. Por ejemplo, de alrededor de 700 formas del verbo de uso frecuente comer, en la base de datos del Senado de la República Mexicana sólo aparecen 29, tales como *comiendose*, *comérselo*, etc. De más de 100 formas de *falsificar*, en la misma base aparecen sólo 11, tales como *falsificarla*, *falsificadas*, etc.

Segundo, uso de **sólo reducción**. El algoritmo no utiliza ningún tipo de generación sino sólo utiliza reducción —tal como la reducción a minúsculas o reducción morfológica. Esto en gran medida simplifica el *lingware*, lo que permite usar un bastante simple algoritmo del análisis morfológico basado en heurística.

Por supuesto, el método sugerido todavía tiene algunas desventajas en comparación con el enriquecimiento de consulta completo o los métodos de enriquecimiento de índice.

Primero, la **necesidad de mantener la lista de las cadenas** de caracteres. En comparación con el enriquecimiento de consulta completo, el método sugerido requiere mantener una estructura de datos adicional. Aunque en la siguiente sección vamos a demostrar que esto no presenta problemas serios de mantenimiento.

Segundo, todavía **aumenta el tamaño de la consulta**. En comparación con el enriquecimiento del índice, el tamaño de las consultas es mayor, aunque no a tal grado como con el enriquecimiento de la consulta completo.

Tercero, las **opciones pueden parecer extraño** al usuario. En comparación con el enriquecimiento de la consulta completo, la lista de las cadenas de caracteres que se presentan al usuario para la edición (véase la sección 5) puede parecer incompleta, sobre todo si el usuario no entiende cómo funciona el método y por qué algunas formas de la palabra, por ejemplo, *computan, computando* e *incomputabilidad* están presentes en la lista, mientras que otros, por ejemplo, *computar* o *computadas,* no están. Esto, sin embargo, no debería de ser un problema grave. Nótese que la lista no se puede completar con las palabras ausentes en la base de datos ya que el algoritmo morfológico basado en heurísticas que se utiliza para la reducción no está diseñado para la generación de todas las posibles formas de la palabra.

7 Arquitectura del sistema

En esta sección describimos la arquitectura del programa en el cual implementamos nuestra metodología para poder conducir experimentos en ella.

7.1 Actualización del índice

En la sección anterior, la necesidad de mantenimiento de la lista de cadenas y la tabla de índice fueron mencionadas como una fuente potencial de complicaciones o acoplamiento no deseable de la tecnología de indización con el *lingware*. Aquí vamos a mostrar cómo evitamos estos problemas en nuestra metodología. Hay dos fuentes potenciales de problemas:

- La actualización de la lista de cadenas caracteres y la tabla del índice cuando cambia la base de datos y
- actualización de la tabla del índice cuando cambia el *lingware*.

El último punto no presenta ningún problema real ya que la lista de las cadenas que se encuentran en la base de datos es muy pequeña en comparación con toda la base de datos. Así, la tabla del índice es simplemente reconstruida a partir de esta lista cada vez que el *lingware* se cambia, sin carga computacional significativa en el sistema.

El primer punto es sólo ligeramente más difícil. Para mantener la existente tecnología de mantenimiento de la base de datos independiente del módulo lingüístico que construye y utiliza la lista de las cadenas de caracteres, utili-

zamos un proceso independiente (un llamado agente computacional) el cual periódicamente, a intervalos de tiempo dependiendo de la carga actual del sistema, sincroniza la lista con la base de datos real. Hay dos maneras posibles de lograr dicha sincronización.

Con el primero método, el índice de la base de datos es accedido por el agente y enumerado en orden alfabético. El agente reconstruye la lista de palabras y la compara con la actual, así detectando que algunas palabras se han introducido y otras han desaparecido. Las entradas de las palabras desaparecidas se remueven de la tabla, mientras que las nuevas palabras se reducen —a las minúsculas, morfológicamente y con el diccionario de sinónimos— y se agregan a la tabla.

El otro método requiere una propiedad adicional del tipo booleano (lógico) del documento, "*indizado*", la cual se guarda en la base de datos junto con cada documento. Cuando se añade un nuevo documento a la base de datos, esta propiedad se establece en *falso*. El agente examina periódicamente la base de datos con la consulta "*indizado* = falso", recupera algunos de los documentos encontrados (dependiendo de la carga del sistema actual), extrae las cadenas de letras a partir de ellos, las agrega a la lista y la tabla si no están allá todavía y marca el documento como *indizado* = *verdadero*.

Los dos métodos tienen las siguientes ventajas y desventajas.

- El segundo método permite el tratamiento de los manejadores de las bases de datos como una caja negra, mientras que el primero requiere operación directa sobre sus estructuras internas.
- El primer método no requiere ningún cambio en la tecnología de mantenimiento de la base de datos utilizada antes de introducir la búsqueda inteligente, mientras que el segundo requiere un pequeño cambio en la estructura de la base de datos.
- El segundo método permite la indización de los documentos con las propiedades no relacionadas con las palabras individuales, sino más bien relacionadas con las combinaciones de palabras específicas o el conjunto de documentos completo, tales como el tema principal del documento [7, 9].
- El segundo método no proporciona una manera fácil para detectar documentos borrados y por lo tanto para eliminar del índice las palabras de tales documentos eliminados.

Esto último no es un problema grave ya que las cadenas que están presentes en la lista pero ausentes en la base de datos no afectan a los resultados de la búsqueda, aunque reducen el rendimiento del sistema. Una de las posibles soluciones a este problema es periódicamente —por ejemplo, una vez al mes— reconstruir toda la lista. Para ello, la propiedad *indizado* debe ser cambiada al

tipo de fecha en lugar de booleano. Para reconstruir la lista, si el proceso de reconstrucción se inició, por ejemplo, el 20 de mayo de 2014, el agente recupera los documentos con "*indizado* $<$ 20 de mayo de 2014", analiza los documentos encontrados y luego restablece la propiedad a "*indizado* = hoy".

7.2 Arquitectura general del sistema

Nuestro sistema está construido sobre la tecnología operativa existente tratada como una caja negra. El flujo de información es interceptado en los tres puntos siguientes.

Primero, la consulta del usuario es interceptada, analizada y sustituida por una consulta enriquecida mediante la técnica de enriquecimiento parcial de la consulta. La nueva consulta se presenta al usuario en un formato de uso fácil para una posible edición. Si es necesario, la consulta enriquecida se divide en una serie de consultas más pequeñas (véase más abajo).

Segundo, la respuesta del manejador de la base de datos es interceptada, analizada y —en caso de una consulta dividida en partes— una respuesta se compila de varios resultados de las consultas parciales.

Tercero, en los momentos de baja carga del sistema, el agente analiza periódicamente la base de datos para actualizar la lista de palabras y por lo tanto la tabla de sustitución utilizada para el enriquecimiento parcial de la consulta.

7.3 Enriquecimiento gradual

Para mejorar el rendimiento en los casos cuando la consulta enriquecida es demasiado grande, la consulta se enriquece sólo parcialmente con las palabras que están más estrechamente relacionadas con la consulta original del usuario. Por ejemplo, en la consulta del tipo 1 presentado en la sección 3.3, primero se efectúe la reducción a minúsculas y luego la reducción morfológica. Sólo en el caso de que una consulta así parcialmente enriquecida no resulta en un número suficiente de documentos encontrados, según lo especificado por el usuario, se efectúe el enriquecimiento de tipo 2 de la ilustración 1. Si esta consulta no es suficiente, se lleva a cabo el enriquecimiento completo de tipo 1.

Sin embargo tan pronto como la consulta parcial resulta en un número suficiente de los documentos recuperados (por ejemplo, diez) éstos se ordenan por relevancia y se presentan al usuario. Se lleva a cabo búsqueda adicional sólo si el usuario solicita más resultados. Con esta técnica, en la mayoría de los casos las consultas mucho más pequeños han demostrado ser suficientes.

Esta técnica se basa en la presuposición de que los documentos que contienen las palabras más cercanas a las palabras clave originales de la consulta del usuario (en la secuencia de minúsculas luego morfología luego ontología) son siempre más relevantes para el usuario. En realidad, el usuario debe poder controlar el orden exacto de la aplicación de las consultas parciales. Por ejemplo, en algunos casos la declinación morfológica podría ser más importante que la reducción a minúsculas: el *Estado* puede ser considerado más cercano a los *Estados* que a *estado* (cf. también el apartado 8.2).

8 Resultados experimentales

Hemos aplicado nuestra metodología a dos proyectos distintos: el buscador para el CORDIAM, Corpus diacrónico y diatópico del español de América, en el marco de colaboración con la Academia Mexicana de la Lengua, y el buscador de los textos legislativos para el Senado de la República Mexicana.

En particular, investigamos a detalle un subconjunto de 200 megabytes de la base de datos del Senado mexicano que contiene una mezcla representativa de los discursos de los senadores, leyes y otros documentos de trabajo del Senado. El corpus contiene 21 millón de palabras, de las cuales sólo 174 mil son diferentes (0.8%). Hemos reducido estas cadenas de caracteres de diversas maneras. Obviamente, la tasa de dicha reducción es igual a la tasa de expansión de la consulta cuando se usa enriquecimiento parcial de la consulta.

Primero, la reducción a minúsculas dio 102 mil cadenas diferentes, lo cual demuestra que con sólo tomar en cuenta la equivalencia de mayúsculas y minúsculas, el tamaño de la consulta se aumenta de manera no significativa, a saber, menos de dos veces. Los resultados para el enriquecimiento morfológico y el enriquecimiento basado en diccionario de sinónimos se discuten en las siguientes subsecciones.

8.1 Enriquecimiento morfológico de la consulta

Para nuestros experimentos hemos utilizado un procedimiento morfológico muy simple basado en una lista de todas las posibles cadenas de sufijos (sufijos morfológicos, terminaciones y clíticos) potencialmente usados en español, un total de 3 mil 578, por ejemplo: *-a*, *-aba*, *-abais*, *-abamos*, *-aban*, *-andoselas*, ..., *-eandoselo*, *-eandoselos*, *-eandoseme*, *-eandosenos*, ..., *-ismo*, *-ismos*, *-ista*, *-istas*, ..., etc. La reducción de una palabra consiste simplemente en remover tal sufijo. En caso de ambigüedad, todas las variantes posibles de reducción son considerados: *hablaba* → *hablab-* y *habl-*. Nótese que nuestra

reducción involucra sufijos significativas, por ejemplo, *comunismo*, *comunista*, *comunes* → *comun-*, lo que aumenta la tasa de enriquecimiento. Nuestra intención fue aumentar el *recall* con un método sencillo y robusto, sin usar diccionarios grandes.

Obviamente el método tan simplista que usamos produce cierto número de raíces incorrectas y a veces erróneamente considera diferentes palabras como si tuvieran una raíz común, por ejemplo, a *démosle* y *día* se les asigna una raíz común *d-*, véase más abajo. En las etapas posteriores del desarrollo de nuestro sistema, se usará un analizador morfológico basado en el diccionario [6] y además los sufijos significativos serán tratados en la ontología. El método basado en sólo en la lista de sufijos todavía se aplicará a las palabras ausentes en el diccionario morfológico. Esto mejorará aún más la proporción de expansión de consultas y la precisión de la búsqueda.

La reducción morfológica con nuestro método simple demostró que la base de datos utiliza 55 mil raíces diferentes. Por lo tanto, el número promedio de cadenas de caracteres por raíz —es decir, la tasa promedia del enriquecimiento parcial de consulta utilizando sólo la morfología— are aproximadamente de cuatro veces. En este proceso distinguimos las letras minúsculas y mayúsculas. Por ejemplo, la raíz *cultiv-* fue representada por tres cadenas de caracteres: *Cultiva, cultivo* y *cultivaron.*

Nótese que las "raíces" que forma nuestro método automático no necesariamente corresponden a las raíces lingüísticas de las palabras; además, las "palabras" que se encuentran en la base de datos no necesariamente son palabras correctas en español sino pueden ser erróneas o con errores de dedo. Con esto, el mayor número de cadenas (incluyendo erróneas) por raíz fue de 279 (la "raíz" *d-* según nuestro método simplista: *D, Dádme, Dé, Démos, Démosle, Démosles, Dénnos, Día, Días, Díza, DA, DADO, ..., duelo, duelos)*, el segundo mayor número fue 201 (la "raíz" *s-* según nuestro método simplista: *S, Sán, Sé, Sí, SA, SADAS, SAL, SALA, SALAS, SALES, SAN, ..., suelo, suelos*), luego 200 (*v-*), 190 (*c-*), 172 (*m-*), 171 (*p-*), 150 (*t-*), 140 (*r-*), 131 (*l-*), 125 (*est-: éstó, ésta, éstan, éstas, éste, éster, ésto, éstos, ESTA, ESTABLE, ESTADO, ESTADOS, ..., estira, esto, estos*), siendo este último el primer elemento de la lista que no fue de una sola letra. Para 183 raíces, es decir, sólo el 0.3% del total de las raíces representadas en la base de datos, el número de cadenas de letras por raíz fue mayor o igual a 50. El número total de cadenas correspondientes a estas raíces fue 12 mil, es decir, 7% del número total de diferentes cadenas de letras en la base de datos.

Como se puede ver, las palabras que causan una alta tasa de enriquecimiento de la consulta son las palabras cortas, en su mayoría formas de los verbos auxiliares, palabras con significado muy amplio o palabras incorrectamente

identificados por nuestro procedimiento como si fuera que tengan la misma raíz. Por lo tanto, aunque la tasa media del enriquecimiento parcial de la consulta con el procedimiento morfológico calculada por las cadenas de letras en nuestra base de datos es de 4, con la exclusión de las palabras con el significado muy amplio que no se utilizan en las consultas reales y la mejora del procedimiento morfológico esta cifra se disminuiría aún más. De hecho, en las consultas de los usuarios reales la tasa media que se observó (con nuestro procedimiento morfológico simplista basado en una lista de sufijos) fue alrededor de 3, que es un resultado muy alentador.

8.2 Enriquecimiento de consulta basado en ontología

Presentamos aquí dos ejemplos de consulta de enriquecimiento de la consulta a base de diccionario de sinónimos de tipo 1 según la ilustración 1. En el diccionario que usamos, el concepto *una ciudad mexicana* se compone de 2 mil 413 nombres. Cuando el nombre de la ciudad consiste de varias palabras, por ejemplo, *La Paz*, consideramos las dos cadenas de forma independiente, como si la lista incluyera ambas palabras *La* y *Paz*, lo que se tradujo en 2 mil 129 cadenas de letras (debido a las repeticiones de partes de los nombres). Sin embargo, la base de datos sólo menciona 1,130 de esas palabras si ignoramos la diferencia de mayúsculas y minúsculas, o bien 1,780 cadenas si distinguimos las mayúsculas y minúsculas.

De hecho sólo 1,077 de ellos eran los nombres de ciudades y el resto eran palabras que coinciden accidentalmente con el nombre de una ciudad o una parte de tal nombre, debido a la reducción a las minúsculas. Por ejemplo, la palabra *paz* coincide con una parte del nombre de la ciudad *La Paz*.

Este ejemplo demuestra una vez más la importancia del control por parte del usuario sobre los tipos de reducción que se aplican a la consulta: en este caso, la reducción basada en tesauro se debe aplicar sin reducción a minúsculas, a pesar de que esta última se percibe como más "básica" y usualmente se aplica de manera rutinaria sin siquiera ofrecer al usuario una opción de deshabilitarla. Con la técnica de enriquecimiento del índice, la decisión sobre el orden de aplicación de los tipos de reducción se toma en el momento de la indización —a pesar de que se puede hacer de una manera inteligente para cada palabra individualmente— y no puede entonces ser cambiada por el usuario.

El concepto *partes del cuerpo humano* en nuestro diccionario está representado por 97 palabras: *barba*, *barbilla*, ..., *mejilla*, *torso*, ..., *tripa*, etc. La base de datos sólo menciona 55 de ellos, o sea 86 si distinguimos mayúsculas y minúsculas. La mayoría de estas palabras fueron mencionadas en los discursos de

los senadores de estilo sonoro, por ejemplo: "¿Y será que ahora ponemos la otra mejilla?", "¡No vamos a caer de rodillas!".

Por lo tanto, con el enriquecimiento de la consulta a base de un diccionario de sinónimos, la tasa de expansión de la consulta es relativamente alta y podría no ser práctico para un sistema real. Sin embargo, como ya hemos mencionado, debido a la naturaleza de un diccionario de sinónimos que refleja el conocimiento "general" que a menudo resulta inadecuado para un usuario en particular, así como debido a la relativamente baja calidad de los diccionarios existentes, este tipo de enriquecimiento más que otros necesita el grado de control de usuario que no puede ser proporcionado por el método del enriquecimiento del índice.

Entonces, consideramos que la desventaja del método de enriquecimiento de la consulta es puramente técnica, es decir, temporal y fácilmente se subsana con mayor memoria y velocidad de las computadoras, mientras que su ventaja —mayor grado de control por parte del usuario— es fundamental y se refleja directamente en la calidad de los resultados de búsqueda. Nótese que a medida de que se está elaborando y expandiendo el diccionario la tasa de expansión de la consulta no se aumentará de manera significativa ya que las nuevas palabras que se añaden al diccionario son de baja frecuencia en los textos. En la siguiente sección, discutiremos una posible solución al problema de alta tasa de expansión de la consulta.

9 Trabajo futuro: una técnica combinada

A pesar de sus ventajas e incluso con la técnica propuesta, el enriquecimiento de consultas aún ralentiza el sistema debido a la inflación de la consulta del usuario, sobre todo en caso de enriquecimiento de la consulta a base de una ontología o un diccionario de sinónimos como se muestra en la ilustración 1. Por otro lado, el método del enriquecimiento del índice tiene otras ventajas, por ejemplo, la ventaja de poder utilizar el contexto de la palabra de una de las siguientes maneras.

Primero, las expresiones compuestas de varias palabras y las frases idiomáticas presentes en el diccionario de sinónimos, tales como *a duras penas* o *ser pan comido*, se pueden manejar de forma más natural en la etapa de la indización del texto completo del documento. Como una variante de las expresiones compuestas, planeamos usar el concepto de las llamadas n-gramas sintácticos [15], los cuales han mostrado buen comportamiento en otras tareas del procesamiento de texto [16]. Los detalles sobre la construcción de tales n-gramas se puede encontrar en el libro [18].

Segundo, las palabras se pueden desambiguar en el contexto: por ejemplo, con la consulta "*sobres*", el texto *la carta está en un sobre* se debe encontrar mientras que *la carta está sobre la mesa* no se debe proporcionar al usuario, siendo el contexto lo que da chance de desambiguar la categoría gramatical de la palabra.

Tercero, puede ser tomada en cuenta la estructura del documento. Por ejemplo, las palabras en el título o en el resumen de un artículo científico se pueden indizar de manera diferente a las palabras del texto principal.

Cuarto, se puede utilizar para la indización las propiedades globales del documento no relacionadas con ninguna palabra en específico en su texto, tales como el tema principal del documento [7, 9].

Para proporcionar estas mejoras sin sacrificar la flexibilidad del lenguaje de consulta, el enriquecimiento del índice puede ser utilizado en combinación con el enriquecimiento de la consulta. El primer paso para dicha combinación es el siguiente. Ambos métodos se implementan en el sistema; en particular, los documentos están indizados con el enriquecimiento del índice tal como se explica en la sección 4. Las consultas de los usuarios de los tipos estándares, tales como la reducción morfológica completa o una consulta basada en el diccionario de sinónimos de tipo 1 completo (véase sección 3.3), se procesan rápidamente con el índice enriquecido sin ningún enriquecimiento de la consulta. Por otro lado, en los casos relativamente poco frecuentes cuando el usuario modifica de alguna manera la lista de cadenas de caracteres las cuales para los efectos de esta consulta específica deben considerarse como equivalentes, se utiliza el enriquecimiento de la consulta.

La división del trabajo entre los dos métodos se puede optimizar. Por ejemplo, sólo los niveles profundos del tesauro pueden ser considerados para el enriquecimiento del índice (*matriz*, *ecuación*, desigualdad, etc. → HIPERÓNIMO: *matemáticas*, véase la sección 4), mientras que la jerarquía de nivel superior, si el usuario así lo indica, puede ser tomada en cuenta en forma del enriquecimiento de la consulta (*ciencia* → HIPÓNIMO: *matemáticas* O HIPÓNIMO: *física* O HIPÓNIMO: *química*).

Más aún, la forma en que los usuarios con mayor frecuencia modifican sus consultas puede ser aprendida y aplicada en el enriquecimiento del índice de forma automática, semiautomática o manual. Por ejemplo, si los usuarios con frecuencia no incluyen la forma *como* del paradigma morfológico del verbo español *comer* (véase la sección 3.2), entonces esta forma debe ser excluida del paradigma de este verbo en la fase del enriquecimiento del índice, mientras que la consulta con el paradigma completo (con la forma *como* incluida) será internamente —y de manera transparente para el usuario— tratado a través del enriquecimiento de la consulta como "MORFOLOGÍA: *comer* O MINÚSCULAS:

como" (obviamente aquí el "paradigma" morfológico del verbo *comer* no incluye la forma *como*; no estamos hablando del paradigma lingüístico real sino de una solución puramente técnica).

Además, el enriquecimiento de índice puede ser mejorado aún más cuando se utiliza en combinación con el enriquecimiento de la consulta. En la sección 4 hemos presentado las marcas para las palabras clave añadidas al índice durante el enriquecimiento, tales como MORFOLOGÍA, MINÚSCULAS, etc. Para permitir una mayor flexibilidad necesaria para las consultas de los tipos 2 y 3 basadas en un diccionario de sinónimos o una ontología (véase la ilustración 1), la distancia (en términos de los niveles) desde la palabra de origen hasta el concepto generalizado también se debe indicar en el índice.

Con esto, el ejemplo de la sección 4 se reescribe de la siguiente manera: *Computadoras* → ..., HIPERÓNIMO-1: *dispositivo,* HIPERÓNIMO-2: *artefacto*, HIPERÓNIMO-3: *objeto,* donde la notación HIPERÓNIMO-3 significa que la palabra *objeto* está a tres pasos en la ontología de la palabra original, en este caso siendo un hiperónimo del hiperónimo del hiperónimo. Ahora las consultas de tipo 2 "*dispositivos,* pero no más de 2 niveles hacia abajo" se implementan como un enriquecimiento de la consulta "HIPERÓNIMO-1: *dispositivo* O HIPERÓNIMO-1: *dispositivo*" y por lo tanto el sistema encontrará los documentos que contiene la palabra *Computadoras*, pero no *Compaq T100*, ya que éste último es un dispositivo demasiado específico y el usuario indicó que no está interesado en los hipónimos demasiado específicos.

Las consultas de los otros tres tipos se implementan de manera similar por la enumeración de los nodos correspondientes. Incluso si el usuario excluye algunas palabras o nodos de la subjerarquía, lo que genera una consulta no estándar, los nodos que mantienen intactos se pueden enumerar en la notación HIPERÓNIMO: ... en lugar de enumerar todas las palabras clave como se sugirió en las secciones 5 y 6 y como está implementado en nuestro sistema actual. Las palabras clave originales tales como HIPERÓNIMO: *computadora* se pueden mantener en el índice para ser utilizados sólo en el caso de las consultas del tipo más frecuente, es decir, tipo 1.

La técnica combinada aliviaría en gran medida el problema de la inflación de la consulta causada por el enriquecimiento de consultas, especialmente en el caso de las consultas basadas en el enriquecimiento a través del diccionario de sinónimos u ontología. Tiene la ventaja de mejor rendimiento debido a consultas más pequeñas, sin sacrificar la flexibilidad del lenguaje de consulta ni la calidad de las respuestas.

Obviamente, esto implica tanto ventajas y desventajas del enriquecimiento de índice. Específicamente, se da la ventaja de la posibilidad de considerar el contexto. Por otro lado, se introduce las desventajas metodológicas y técnicas

del enriquecimiento del índice mencionadas en la sección 4, por ejemplo los problemas de mantenimiento y acoplamiento indeseable del manejador de la base de datos con el *lingware*. Una investigación más profunda de la técnica combinada será la dirección de nuestro trabajo futuro.

10 Conclusiones

En este trabajo hemos considerado el problema de recuperación de información de las bases de tamaño mediano (no de tamaño de la web abierta) en la circunstancias que requieren alta calidad de la respuesta y consecuentemente gran flexibilidad del lenguaje de consulta. Para esto, analizamos dos aproximaciones al problema de comparación no literal de las palabras clave de la petición del usuario y del documento: el enriquecimiento del índice y el enriquecimiento de la consulta.

Hemos demostrado que la técnica usada de manera tradicional para tal tarea —la técnica llamada el enriquecimiento del índice— tiene desventajas inherentes importantes, y hemos propuesto una nueva técnica —el enriquecimiento parcial de la consulta— la cual permite una mayor flexibilidad de las consultas, mejor arquitectura general del sistema y un mantenimiento más sencillo del mismo.

Hemos probado nuestra metodología en el marco de dos proyectos: el buscador de los textos legislativos para el Senado de la República Mexicana y el buscador para el CORDIAM, Corpus diacrónico y diatópico del español de América, en el marco de una colaboración con la Academia Mexicana de la Lengua.

Nuestro método todavía tiene al menos dos problemas. Primero, las consultas enriquecidas construidas en algunos casos resultan significativamente más grandes y por lo tanto funcionan más lento. Segundo, no es obvio cómo en nuestro método se puede tomar en cuenta el contexto de la palabra clave para su desambiguación en el documento. Como una posible solución a estos dos problemas, hemos discutido la posibilidad de combinar los dos métodos: el enriquecimiento de la consulta y el enriquecimiento del índice, lo cual será nuestro trabajo futuro.

Agradecimientos. Este trabajo fue parcialmente apoyado por el Gobierno de México (SNI, SIP-IPN proyecto 20144534). En el trabajo se usaron los recursos léxicos y la experiencia de los proyecto apoyados por la Academia Mexicana de la Lengua (proyecto CORDIAM) y por el Senado de la República Mexicana.

Referencias

1. Aho, Alfred V. *Algorithms for finding patterns in strings*. En: J. van Leeuwen (ed.), Handbook of Theoretical Computer Science, capítulo 5, pp. 254–300. Elsevier Science Publishers B. V., 1990.
2. Cassidy P. *An Investigation of the Semantic Relations in the Roget's Thesaurus: Preliminary Results*. En: A. Gelbukh (ed.), Computational Linguistics and Intelligent Text Processing, Proc. of CICLing 2000, February 2000, IPN, Mexico City, 2000.
3. Gelbukh, A. *Exact and approximate prefix search under access locality requirements for morphological analysis and spelling correction*. Computación y Sistemas, Vol. 6, N 3, 2003, pp. 167–182.
4. Gelbukh, A. *Lazy query expansion*. Computación y Sistemas, Vol. 6, No. 1, July-September 2002, p. 13–24.
5. Gelbukh, A. *Ontology-based Semantic Relatedness Measures: Applications and Calculation*. Research in Computing Science, Vol. 47, 2012, pp. 117–138.
6. Gelbukh, A., G. Sidorov. *Approach to construction of automatic morphological analysis systems for inflective languages with little effort*. En: Computational Linguistics and Intelligent Text Processing, Proc. of CICLing 2003. Lecture Notes in Computer Science, N 2588, Springer, pp. 215–220.
7. Gelbukh, A., G. Sidorov, A. Guzmán-Arenas. *Use of a Weighted Topic Hierarchy for Document Classification*. En: Text, Speech, Dialogue. Lecture Notes in Artificial Intelligence, N 1692, Springer, 1999.
8. Gusfield, Dan. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
9. Guzmán-Arenas, A. *Finding the main themes in a Spanish document*. Expert Systems with Applications, Vol. 14, No. 1/2. Enero-febrero de 1998, pp. 139–148.
10. Fellbaum, C. (ed.) *WordNet as Electronic Lexical Database*. MIT Press, 1998.
11. Frakes, W., R. Baeza-Yates, editores. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, 1992.
12. Hausser, Roland. *Three principled methods of automatic word form recognition*. En: Proc. of VEXTAL: Venecia per il Tratamento Automatico delle Lingue. Venice, Italy, Sept. 1999.

13. Koskenniemi, Kimmo. *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. University of Helsinki Publications, N 11, 1983.
14. Lenat, D. B., R. V. Guha. *Building Large Knowledge Based Systems*. Reading, Massachusetts: Addison Wesley, 1990.
15. Sidorov, G. *N-gramas sintácticos no-continuos*. Polibits, vol. 48, pp. 67–75, 2013.
16. Sidorov, G. *Syntactic dependency based n-grams in rule based automatic English as second language grammar correction*. International Journal of Computational Linguistics and Applications, 4(2), 2013, pp. 169–188.
17. Sidorov, G. *Modelos formales en la lingüística computacional*. Universitat Autònoma de Barcelona, 2013, 220 pp.
18. Sidorov, G. *Construcción no lineal de n-gramas en la lingüística computacional: n-gramas sintácticos, filtrados y generalizados*. Sociedad Mexicana de Inteligencia Artificial, 2013, 166 pp.