Corpus morfológicamente representativo: preparación de datos y compilación para el español

Liliana Chanona-Hernández[1] y Alexander Gelbukh[2]

[1] ESIME-Zacatenco,
Instituto Politécnico Nacional (IPN),
México DF,
México

[2] Centro de Investigación en Computación (CIC),
Instituto Politécnico Nacional (IPN),
México DF,
México

lchanona@gmail.com, gelbukh@gelbukh.com

Resumen. El problema de compilación automática de los corpus es uno de los problemas importantes de lingüística computacional. En los corpus tradicionales algunas palabras tienen demasiada ocurrencia y algunas tienen poca o ninguna ocurrencia según la ley de distribución de palabras de acuerdo a su rango: la ley de Zipf. En el trabajo proponemos el concepto del corpus representativo morfológicamente —cuando para cada palabra de alguna lista se garantiza por lo menos algún número determinado de contextos— y se describe un ejemplo de aplicación al español: la compilación automática de tal corpus a través de Internet, incluyendo la preparación de los datos iniciales y el filtrado de los contextos.

Palabras clave. Corpus representativo, español, representatividad morfológica, Internet.

1. Introducción

El tesoro más valioso de la raza humana es el conocimiento, es decir, la información. Existe en el mundo volúmenes inmensos de información en forma de lenguaje natural: los libros, los periódicos, las revistas, etcétera. Pero la posesión verdadera de este tipo de tesoro implica la habilidad de hacer ciertas operaciones con la información:

- Buscar la información necesaria.
- Comparar las diferentes fuentes, hacer inferencias lógicas y concluir.
- Manejar los textos, por ejemplo, traducirlos a otros idiomas.

Todo parece estar preparado para el uso de las computadoras para procesar volúmenes grandes de información: los métodos lógicos ya son muy fuertes, los procesadores muy rápidos, muchos textos ya están disponibles en forma digital, tanto en las casas editoriales como en Internet. El único problema para la computadora al procesar los textos es que para manejarlos bien hay que entenderlos [1, 2]. Sin eso, éstos son solamente cadenas de letras sin ningún sentido y no una información útil para el razonamiento lógico, lo que es una de las tareas más importantes es la búsqueda y el uso de la información contenida en el texto. Los motores de búsqueda permiten encontrar una infinidad de documentos aquellos que satisfagan una necesidad del usuario descrita en su petición. En el caso simple, la petición contiene las palabras clave, como «pensar y futuro», que quiere decir que el usuario necesita los documentos que contengan ambas de estas palabras. Hasta en este caso simple, se necesita un análisis bastante complejo: los documentos que contienen, las palabras futuras, piensan y probablemente pensador y pensamiento son también relevantes. Si el usuario indica que necesita los documentos que dicen sobre la acción «pensar en futuro», entonces el documento contiene la frase como «piensan en las futuras aplicaciones» probablemente es relevante, mientras que «en el futuro próximo voy a pensarlo» probablemente no lo es. Para hacer esta decisión es necesario un análisis profundo del texto.

La búsqueda eficaz de documentos depende de las soluciones técnicas. No es posible analizar todos los documentos cada vez que el usuario hace su pregunta. Entonces, se hace y se maneja un índice de los documentos, es decir, una representación corta, simple y formal de los documentos. La representación de este índice, el tipo de información incluida en él y los métodos matemáticos que se emplean en la búsqueda en él afectan mucho los resultados y eficiencia del proceso.

Típicamente, un corpus se construye de tal manera que los textos de diferentes géneros, sobre diferentes temas, escritos por diferentes autores, etc., se representan en él en una mezcla balanceada, para reflejar las propiedades promedio del lenguaje. Esta propiedad del corpus también se llama representatividad (respecto a los temas, géneros, etc.) [7], aunque en este trabajo usamos otro significado de la palabra *representativo*, hablando del corpus representativo respecto a las palabras individuales.

El uso de los corpus en lingüística es de suma importancia [8]. Los métodos actuales para el análisis de lenguaje natural emplean ampliamente los conocimientos sobre el lenguaje, su uso, y el mundo real, es decir, diccionarios de una gran variedad de tipos. En la compilación de éstos, hay dos momentos críticos en que se usan los corpus:

- Extracción de información **simbólica**. En el diccionario de tipo simbólico, se almacenan los datos sobre la existencia (o no existencia) de alguna relación o propiedad. Por ejemplo, el hecho de que la palabra *ayuda* se usa tanto con el verbo *prestar* como con los verbos *dar* y *proporcionar*: *prestar ayuda*, *dar ayuda*, *proporcionar ayuda*. Incluso para las cantidades pequeñas de palabras, a una persona le cuesta trabajo recordar toda la información necesaria y llenar el diccionario apoyándose sólo en introspección. A diferencia, para los diccionarios del tamaño realista, estos datos se deben extraer del uso de las palabras en el corpus –en

nuestro ejemplo, buscando todos los verbos con los cuales se usa la palabra *ayuda*.

- Extracción de información **estadística**. Sin embargo, la información solamente simbólica no es suficiente en muchas aplicaciones donde hay que resolver incertidumbre y ambigüedad. En muchos casos se necesita saber no sólo con cuales verbos se usa la palabra, sino con cuáles se usa más que con otros: 60%: *prestar ayuda*, 30%: *dar ayuda*, 10%: *proporcionar ayuda*.

Esta información no se puede compilar manualmente con el método de introspección, sino sólo se puede extraer de un corpus amplio con los métodos estadísticos.

En el resto del artículo, primero presentamos el concepto del corpus representativo morfológicamente, después describimos el algoritmo de su compilación de manera general, y posteriormente su aplicación para el caso del idioma español (selección de la lista inicial de palabras y ponderación morfológica). Al final se presentan las características del corpus compilado para el español y se dan las conclusiones.

2. El corpus representativo morfológicamente

El problema de casi cualquier investigación estadística en lexicografía es el hecho de que la mayoría de las palabras se encuentran en el corpus muy pocas veces, aunque un número relativamente pequeño de las palabras ocurren muchas veces y constituyen casi todo el corpus. Esto se conoce como la ley de Zipf: la palabra con el rango estadístico n tiene aproximadamente la frecuencia C / n. (C es una constante) por consecuencia, los resultados estadísticos compilados para la mayoría de las palabras del diccionario con el corpus son de baja confiabilidad, aunque sea un corpus muy grande, véase [9, 10].

De hecho, esta distribución es aún más sesgada y se aproxima más al inverso del cuadrado de n. Es decir, hay un conjunto pequeño de palabras muy frecuentes y muchas que aparecen muy pocas veces o sólo una vez (sea cual sea el idioma usado).

Este significa que para la mayoría de las palabras no hay suficiente información estadística (suficiente número de contextos) aunque el corpus sea muy grande y ocupe mucho espacio.

Entonces, los corpus tradicionales no son perfectamente adecuados para las tareas de PLN. La ley de Zipf se refiere a que muy pocas palabras en cualquier lenguaje son muy frecuentes, mientras que la mayoría de las palabras son poco frecuentes y de hecho las palabras raras son totalmente carentes de frecuencia. Por consecuencia, cuando se quiere hacer un estudio sobre cuál es el contexto de cualquier palabra con poca frecuencia es muy difícil porque casi no se tiene ninguna (o se tiene muy poca) información acerca de sus contextos. Es decir, en el corpus tradicional la información sobre la mayoría de las palabras no es estadísticamente significativa.

Un peor problema se presenta por la ley de Zipf cuando se trata de la investigación de ocurrencias de palabras. Si cada una de las palabras (tal que *ayuda* o *proporcionar*)

tiene poca frecuencia, su ocurrencia (*proporcionar ayuda*) tiene la frecuencia en el corpus casi nula y totalmente insuficiente para cualquier conclusión estadística.

Por otro lado, casi todo el volumen de los datos (y entonces, espacio y tiempo de procesamiento de datos por la computadora) de un corpus tradicional se ocupa por millones de inútiles repeticiones de las mismas 100–1,000 palabras.

En nuestro trabajo proponemos la solución a este problema: un corpus en el cual cada palabra tiene una frecuencia suficiente para su investigación estadística, este trabajo es una extensión de nuestro trabajo anterior [3], también se hace un cambio del enfoque con respecto al trabajo anterior.

Como la solución del problema de compilación del corpus representativo, se propone usar Internet como una fuente inmensa de los contextos típicos de las palabras. Sin embargo, no es factible técnicamente bajar de Internet todos los documentos en español que en éste existen. Afortunadamente, no es necesario, pues sólo se necesita un número limitado (digamos, de 50 contextos) para un número limitado (aproximadamente 100 mil) de palabras que existen en español (o bien, un millón de formas de palabras).

Este tipo de diccionarios se conoce en literatura como *concordancias* o diccionarios tipo KWIC (por sus siglas en inglés: *key words in context*). Nuestra solución propuesta, entonces, es la compilación de una concordancia tipo KWIC muy grande a través de la extracción de los contextos necesarios de los textos en español que se encuentran en Internet y con las posibilidades de enriquecimiento automático (no realizadas en nuestro trabajo). Sería muy bueno tener los contextos suficientemente grandes para su posterior análisis sintáctico automático, por ejemplo, con el parser libremente disponible FreeLing [1, 6].

La implementación de esta idea se describe en las siguientes secciones.

La Web es un gran repositorio de datos y un nuevo medio de publicación al alcance de más de mil millones de personas. El hacer uso eficiente y adecuado de estos datos depende de las herramientas que existen.

La técnica es usar una máquina de búsqueda (*search engine*) como AltaVista, Fast, Inktomi, Northern Light, Lycos o Google, que usan el paradigma de recuperación en texto completo. Es decir, todas las palabras de un documento se almacenan en un índice para su posterior recuperación. Aunque en muchos casos las búsquedas en estas máquinas son efectivas, en otros son un total desastre. El problema es que las palabras no capturan toda la semántica de un documento. Hay mucha información contextual o implícita que no está escrita, pero que entendemos cuando leemos. Los problemas principales son la *polisemia*, es decir, palabras que tienen más de un significado, y por lo tanto encontramos páginas que no queremos; y la *sinonimia*, palabras distintas que tienen el mismo significado y por ende si no usamos la palabra correcta, no encontramos lo que queremos.

4. Algoritmo de compilación del corpus para el español

4.1. El algoritmo principal

El esquema general del método se presenta en la Fig. 1

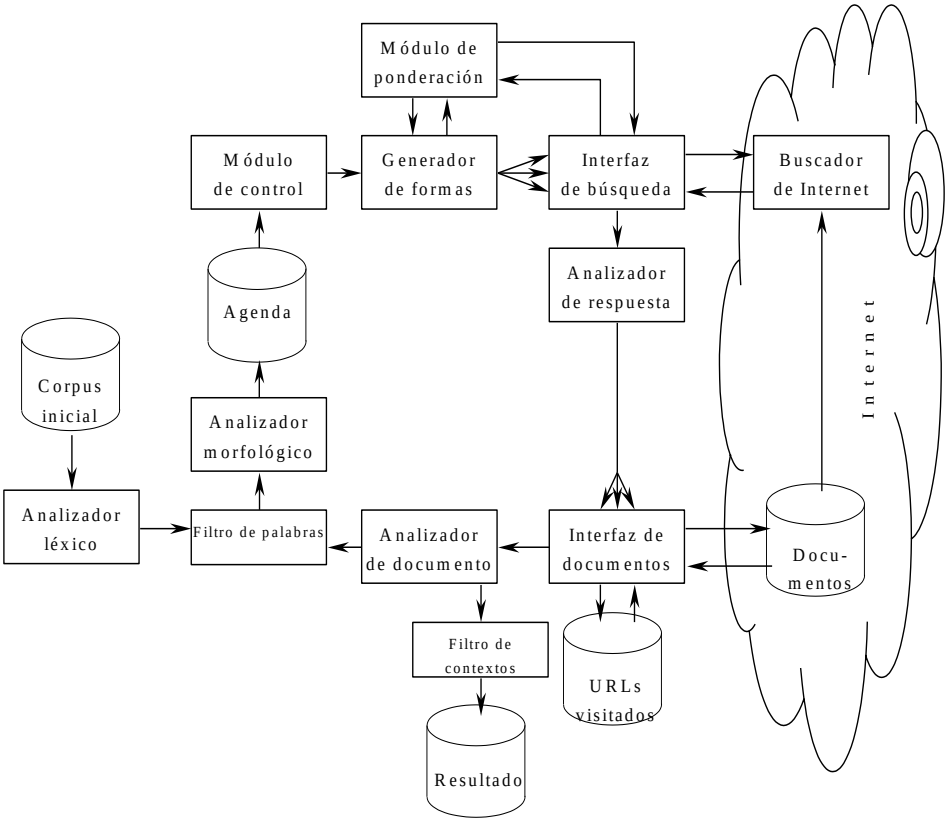

Fig. 1. Esquema general del método.

La idea de su funcionamiento es la siguiente. En la *Agenda* están las palabras que esperan su turno para ser procesadas. El *Módulo de control* las toma una tras otra de la Ag.enda y busca los contextos de la palabra en el Internet. Los resultados encontrados se guardan, pero antes se analizan para encontrar nuevas palabras, las cuales se agregan a la Agenda para que esperen su turno a ser procesadas. Inicialmente, la agenda se llena con una cantidad inicial de palabras en español.

Describimos en más detalle la relación entre sus partes.

Primero se tiene una fuente inicial de palabras, a la cual llamamos genéricamente como el *corpus* (empleamos los diccionarios existentes Anaya y el diccionario de la Universidad Politécnica de Cataluña y los corpus existentes LEXESP y el corpus compilado en el Laboratorio) de donde se sacan las palabras para empezar la búsqueda de los contextos en Internet.

Las palabras pasan por el *analizador léxico* que es el que se encarga de sacar todas las palabras de dicho corpus, con cierto tipo de estructura, un ejemplo (del diccionario Anaya) es:

trabajador | *, -ra*,

Este tipo de estructura no es permitido, así que la función del analizador léxico es la de transformar este renglón a dos palabras:

trabajador
trabajadora

admitidas más adelante en la *agenda* (que es la tabla donde se guardan las palabras).

Enseguida las palabras pasan por el *filtro* que es el que se encarga de verificar que las cadenas de caracteres que son mezcla de letras y números, no entren en la agenda (que es la tabla donde se guardan las palabras).

Una vez que las palabras ya fueron filtradas son pasadas por un *analizador morfológico* que se encarga de normalizar las palabras. La normalización se realiza porque, es más fácil y practico trabajar con las palabras normalizadas, así de este modo más adelante las palabras son pasadas por un *generador de formas* que como su nombre indica nos da todas las formas morfológicas de una palabra, asegurándonos que tendremos todas las formas de una palabra.

Conforme cada palabra ha sido filtrada y pasada por el analizador morfológico se almacena en una *agenda* que es la lista de palabras que ya cumplieron con ciertas normas.

Esto da pie a que las palabras sean tomadas por un *módulo de control* que es el que se encarga de pasar las palabras a un generador de formas morfológicas.

El *generador de formas* se encarga de generar todas las formas morfológicas de una palabra, un ejemplo simple es el sustantivo *mesa* y sus formas son *mesa* y *mesas*, este generador de formas es importante porque es necesario tener todas las variantes (formas morfológicas) de una palabra, pues de lo contrario el corpus representativo de palabras de español no estaría completo. Explicado de otra manera se busca tener una representatividad equilibrada de todas las formas morfológicas de una palabra.

Un punto importante que hay que mencionar, es el hecho de que existen tres formas de hacer un corpus representativo de palabras:

- El primero es el de compilar el corpus para las palabras normalizadas, cuando el número determinado de contextos en total se distribuye uniformemente entre sus formas morfológicas.
- La segunda forma, es la de tener todas las palabras normalizadas con todas sus respectivas formas, con un determinado número de contextos para cada forma.
- Finalmente, la que se está usando en este trabajo que es la de calcular para cada forma morfológica de dicha palabra el número de contextos que se desea obtener.

Cuando que ya se tiene todas las formas morfológicas de una palabra, cada una de las formas y la palabra normalizada se pasan a la *interfaz de búsqueda* que es el mediador entre el *generador de formas* y el *buscador de Internet*, una vez que el buscador le contesta a la interfaz, la interfaz de búsqueda a su vez le informa al *módulo de ponderación* (que es el encargado de hacer los respectivos cálculos de cuantos contextos se desea, tanto para la palabra normalizada, como para las formas morfológicas de dicha palabra), cuantos documentos existen en Internet con esas palabras. El módulo hace los cálculos y le dice al generador de formas cuantos

contextos tendrá cada forma de la palabra y la palabra misma. De este modo cuando ya el generador de formas sabe cuántos contextos exactamente tendrá cada palabra, vuelve a pasar las palabras a la interfaz de búsqueda para luego pasarlas al buscador de Internet.

El paso siguiente se da cuando, el buscador le envía todas las direcciones a la interfaz de búsqueda, quien ésta vez, las envía a un módulo *Analizador de respuesta* que se encarga de depurar todo el excedente que acompaña al URL.

Por ejemplo:

Descargue GRATUITAMENTE MSN Explorer en http://explorer.msn.es/intl.asp

Todo lo que se encuentra sombreado será eliminado, pues no es de utilidad.

La *Interfaz de documentos* es la que se encarga de conectarse a cada uno de los documentos con las direcciones que le pasa el *Analizador de respuesta.* Cuando ya se ha revisado cada uno de los documentos, sus direcciones son almacenadas en una tabla que es de los URLs visitados, esto es para no volver a visitar dos veces un mismo sitio.

El *Analizador de documento* es el que se encarga como su nombre lo indica de analizar cada uno de los documentos, para esto marca los párrafos que deben considerarse para el estudio y también auto enriquece el corpus, pasando al filtro las palabras nuevas que encuentra.

El *filtro de contextos* es el que se encarga de ver que el contexto no contenga menos de un número determinado de palabras y de que los contextos no sean repetidos, cuando ya la palabra pasa este filtro es almacenada en una tabla que se llama *Resultados*.

Para funcionar correctamente, el programa mantiene los siguientes datos principales:

1. Una lista de palabras para las cuales se pretende encontrar los contextos.
2. Una lista de los URLs encontrados para una palabra.
3. Una lista de contextos para cada palabra.

El algoritmo principal se esboza a continuación.

Paso 1. Hacer vacías todas las listas.

Paso 2. Agregar a la lista de palabras, todas las palabras encontradas en el corpus inicial, en este caso de la lista de palabras definidas en el diccionario Anaya.

Paso 3. Si todas las palabras ya se procesaron, terminar el trabajo.

Paso 4. Para una palabra todavía no procesada, buscar contextos en Internet.

Paso 5. Para cada contexto encontrado en el Paso 4, aplicar las heurísticas para determinar si el contexto contiene alguna irregularidad.

i. Si la contiene, ignorar el contexto y aumentar el número de contextos irregulares.

ii. Si no la contiene, agregar el contexto a la lista de los contextos encontrados para la palabra.

Paso 6. Si para la palabra ya se encontraron al menos 50 contextos válidos, marcar la palabra como ya procesada. Si no existen en Internet más contextos para esta palabra, también marcar la palabra como ya procesada. Ir al Paso 3.

4.2. Análisis y filtrado del texto obtenido

Al obtener el texto del documento en el formato HTML se buscan las ocurrencias de la palabra y se evalúa si los contextos son apropiados o no, como se describa a continuación.

El primer paso es quitar el marcado del formato HTML, para lo cual se usa el algoritmo desarrollado previamente en el Laboratorio de Lenguaje Natural. Es importante notar que esta función conserva la estructura de bloques de documento, por ejemplo, cada celda de una tabla pertenece a un bloque diferente. Esto se logra con sustitución de cada elemento del marcado con un símbolo especial, el cual es "#".

Después en el texto se busca la palabra, lo cual se hace con la función estándar de C `strstr()`. Para hacer esta búsqueda, se tiene que hacer una copia del documento en minúsculas. Ya que nos interesan solo palabras y no partes de palabras, se verifica que es una palabra completa (es decir, que no existen símbolos de letras inmediatamente antes y después del segmento encontrado).

El siguiente paso en el análisis es la búsqueda del contexto de la palabra que se agrega a la lista de contextos potenciales. Más tarde este contexto pasa por el filtro de contextos el cual rechaza los contextos "malos", véase la sección 4.3.

Como contexto se toma la misma oración donde se encontró la palabra, la oración anterior, y la oración siguiente (si las dos últimas existen). Pero al mismo tiempo se limita el número de palabras a la izquierda y a la derecha de no más de 25; las palabras más lejanas se ignoran. La limitación del número de palabras (25) se debe a la necesidad de limitar el tamaño del contexto para que la base de datos no sea demasiado grande. Además, las palabras más lejanas usualmente no tienen ninguna relación lingüísticamente interesante con la palabra en cuestión.

Para encontrar el contexto, se hace el siguiente análisis:

1. Se busca a la izquierda de la palabra el símbolo de fin de oración (punto, signo de exclamación o de interrogación) y se cuenta el número de palabras (no debe ser mayor que 25). También sólo se permite un símbolo de fin de oración a la izquierda para que sólo se tome la oración anterior. Se toman en cuenta los símbolos de fin de bloque ("#") para que el contexto no los rebase.

2. Se hace el análisis semejante, a la derecha de la palabra.

Como el resultado de este algoritmo se obtiene el contexto potencial el cual va a pasar al filtro y ser aprobado (se guarda como un resultado final) o rechazado (se ignora).

4.3. Filtros adicionales de contextos encontrados

Después del proceso de extracción del contexto, se tiene el contexto potencial y la lista de contextos encontrados para la palabra. Ahora se toma la decisión si el contexto es bueno para incluirlo en el corpus que se está compilando.

Primero se usa el criterio del **tamaño** del contexto: el contexto debe contener no menos que un número dado de palabras, usamos el valor 8. Aplicamos este criterio porque los contextos demasiado pequeños no contienen suficiente información lingüística y no son de gran interés. Más importante, los contextos pequeños frecuentemente no son expresiones de lenguaje natural sino otros tipos de datos (rótulos cortos de figuras, inscripciones en los controles de la pantalla como los botones, nombres de archivos, etc.).

Otro filtro que se usa tiene como propósito filtrar los apellidos y **nombres propios**. Los experimentos han mostrado que una palabra puede ser usada muchas veces como apellido (por ejemplo, casi todos los contextos encontrados para la palabra *abad* referían al apellido). Entonces, se verifica que la palabra no sea un apellido. Para eso se usa la siguiente heurística: si la palabra empieza con mayúscula y no tiene inmediatamente antes un signo de fin de oración, entonces consideramos que este contexto contiene apellido o nombre propio y por lo tanto es inaceptable.

El último filtro verifica que los contextos no **se repitan**. Para eso se verifican las dos palabras significativas que se encuentran inmediatamente a la izquierda y a la derecha de la palabra con la cual se trabaja. Estas palabras se comparan con todos los contextos ya encontrados; la cadena de las tres palabras no debe repetirse. Si se repiten, es decir, el contexto con estas palabras ya está en la base de datos, entonces el contexto se considera inaceptable. Nota: solo se verifican las palabras significativas, es decir, se ignoran los artículos, pronombres (*te, tí, se,* etc.) y preposiciones (*con, a, por, para,* etc.).

5. Preparación de lista inicial de palabras

Antes que el sistema pueda buscar en Internet los contextos para las palabras, es necesario preparar la lista de las palabras. Para prepararla se usó el diccionario explicativo del español desarrollado por el grupo español Anaya 1996. El diccionario se tiene en el Laboratorio de Lenguaje Natural del CIC-IPN en forma del archivo de texto. Lo que se hizo fue seleccionar las palabras encabezadas (para cuales hay definiciones en el diccionario) y formar la lista de estas palabras. En total el diccionario contiene alrededor de 30,000 palabras definidas.

Los problemas principales en la compilación de esta lista de palabras fueron relacionados con el análisis de la estructura del diccionario.

Hay que mencionar que las palabras definidas en el diccionario explicativo ya están normalizadas. Entonces, la etapa del análisis morfológico fue innecesaria en caso de este tipo de entrada. En caso que esta etapa fuese necesaria (digamos, cuando se usa algún corpus como la fuente de la lista de palabras), se utiliza la herramienta desarrollada en el Laboratorio de Lenguaje Natural: un analizador morfológico [2].

Ahora bien, al tenerse la lista de palabras normalizadas, es necesario para cada palabra normalizada obtener todas sus formas morfológicas y usarlas como peticiones a los motores de búsqueda en Internet. Para generar todas las formas morfológicas de cada una de las palabras se usó otra herramienta desarrollada previamente en el Laboratorio de Lenguaje Natural —un generador de formas de palabras (que es, de hecho, el mismo analizador morfológico invertido).

Aplicando esta herramienta, obtenemos para cada palabra normalizada (lema) la lista de sus formas gramaticales y guardamos todas estas formas en una base de datos.

Al terminar este procedimiento, la lista inicial está hecha. En el sistema desarrollado, el número de formas de palabras es alrededor de 100,000 (para 30,000 lemas).

6. Ponderación de las formas de palabras

Existen varias formas de hacer un corpus representativo de palabras, véase la discusión en [4]. Las más sencillas son:

- Un modo es el de compilar el corpus para las palabras normalizadas, cuando el número determinado de contextos en total se distribuye uniformemente entre sus formas morfológicas.
- La segunda forma, es la de considerar todas las palabras normalizadas con todas sus respectivas formas, con un determinado número de contextos para cada forma morfológica de la palabra dada.
- Finalmente, la que usamos en este trabajo es la de calcular para cada forma morfológica de dicha palabra el número de contextos que se desea obtener, de acuerdo con las estadísticas de uso de las formas específicas.

Ya que escogimos la opción de compilar el corpus usando la representación igual de lemas y hacer la ponderación de las formas de palabras, entonces necesitamos asignar las frecuencias a todas las formas gramaticales correspondientes a cada lema.

Decidimos que la ponderación se haga tomando en cuenta la frecuencia de la palabra en Internet. Entonces, eso se hace en el programa que busca las palabras en Internet.

El proceso consiste en 3 pasos:

1. Buscar en Internet el número de documentos correspondientes a cada forma de la palabra y guardar el resultado en una base de datos.
2. Para cada lema, calcular la sumatoria de los documentos que corresponden a cada de sus formas gramaticales,
3. Para cada forma gramatical, calcular su peso, usando la fórmula:

$$p = n / N$$

donde p es el peso, n es el número de los documentos que existen en Internet para cada forma gramatical, N es el número de documentos que existen en Internet

para el lema (la sumatoria de los números de documentos existentes para cada forma gramatical).

Este proceso tiene un parámetro el cual es el número de los contextos para cada lema. Se escogió el valor de 50 contextos, que da el número de contextos suficientemente grande.

El resultado de la ponderación es el número de contextos para cada forma gramatical del lema, siendo fijo el número de contextos para cada lema.

Por ejemplo, para el lema *ABAD* el cual tiene dos formas gramaticales: *ABAD* y *ABADES*, para la forma *ABAD* existen 57,900 documentos, y para la forma *ABADES* (la cual es mucho menos usada) existen tan sólo 3,310 documentos. Por eso, para la primera forma se buscó 47 contextos, y para la segunda solamente 3 contextos.

7. El corpus compilado

Como el resultado palpable del presente trabajo se obtuvo un corpus representativo de los contextos de palabras en español. El corpus obtenido tiene las siguientes características estadísticas generales, presentes en las Tablas 1 y 2.

Tabla 1. Características generales del corpus.

Propiedad	Valor
Tamaño del corpus, megabytes	221
Número total de palabras encabezado (lemas)	30,198
Número total de formas de palabras encabezado	99,938
Número total de lemas en los contextos (con cadenas no conocidas)	470,417
Número total de formas de palabras en los contextos	556,702
Número promedio de palabras por contexto	37
Número promedio de palabras significativas por contexto	18

El análisis estadístico muestra la siguiente distribución del número de ocurrencias de las palabras en el corpus obtenido, Tabla 2.

Tabla 2. Número de palabras y lemas en el corpus compilado.

Característica	Formas de palabras	Lemas (con cadenas no conocidas)
Total diferentes	556,702	470,417
Máximo	2,302,074 (*de*)	2,302,074 (*de*)
Más de 100	23,211	18,402

Característica	Formas de palabras	Lemas (con cadenas no conocidas)
11-100	77,676	38,826
2-10	194,460	162,834
1	261,356	250,357

En la Tabla 2 se presentan varios rangos de frecuencias de las palabras y sus totales respectivas en el corpus.

8. Conclusiones

En este artículo hemos propuesto un método para preparar un diccionario morfológicamente representativo. Nos basamos en la lista de formas gramaticales de palabras ponderadas según sus frecuencias en Internet. Posteriormente, a base de una lista de las palabras iniciales, se generan sus formas morfológicas, se ponderan, y se generan los contextos (concordancias). Hemos construido un corpus así para 30,000 lemas (100,000 formas gramaticales) de palabras de español, que tiene 50 contextos pos lema (y proporcional por cada forma).

En el trabajo futuro habrá que realizar el análisis del corpus construido y su comparación con otros corpus. También probar el método propuesto para otros idiomas u para otro número de contextos. Otra dirección de trabajo es aplicar el corpus en varias tareas de PLN.

Agradecimientos. Trabajo realizado con el apoyo parcial del gobierno de México (CONACYT,) e Instituto Politécnico Nacional, México (proyecto SIP 20121823), proyecto FP7-PEOPLE-2010-IRSES: Web Information Quality - Evaluation Initiative (WIQ-EI) European Commission project 269180.

Referencias

1. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA (1999)
2. Gelbukh, A., Sidorov, G.: Procesamiento automático del español con enfoque en recursos léxicos grandes. IPN, 307 p. (2010)
3. Sidorov, G., Gelbukh, A., Chanona-Hernández, L.: Corpus virtual virtual: un diccionario grande de contextos de palabras españolas compilado a través de Internet. In: Proc. of Workshop "Multilingual information access and natural language processing" of IBERAMIA 2002 (8th Iberoamerican conference on Artificial Intelligence), Sevilla, Spain, November, 12, pp 7-14 (2002)
4. Sidorov, G., Barrón-Cedeño, A., Rosso, P.: English-Spanish Large Statistical Dictionary of Inflectional Forms. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta. European Language Resources Association (ELRA), pp. 277-281 (2010)

5. Carreras, X., Chao, I. Padró L., Padró, M.: FreeLing: An Open-Source Suite of Language Analyzers. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04) (2004)
6. Padró, L., Stanilovsky, E.: FreeLing 3.0: Towards Wider Multilinguality. In: Proceedings of the Language Resources and Evaluation Conference (LREC 2012) ELRA. Istanbul, Turkey (2012)
7. Biber, D.: Representativeness in corpus design. Literary and linguistic computing, 8:243-257 (1993)
8. Biber, D., Conrad, S., Reppen, D.: Corpus linguistics. Investigating language structure and use. Cambridge University Press, Cambridge (1998)
9. Gelbukh, A., Sidorov, G.: Zipf and Heaps Laws' Coefficients Depend on Language. Proc. CICLing-2001, Conference on Intelligent Text Processing and Computational Linguistics, Mexico City. Lecture Notes in Computer Science N 2004, Springer-Verlag, pp. 330–333 (2001)
10. Zipf, G.K.: Human Behavior and the Principle of Least Effort. Addison-Wesley Publishers Co., Reading, MA. (1949)