Sistema de extracción de cuerpos de texto de la web para tareas lingüísticas

Web text corpus extraction system for linguistic tasks

Héctor Fabio Cadavid Rengifo[1] y Jonatan Gómez Perdomo[2]

RESUMEN

En este artículo se describe un sistema desarrollado para la extracción de grandes cuerpos de texto de Internet, teniendo como motivación el valor que ofrecen los ejemplos de lenguaje natural disponibles en la red para las tareas de aprendizaje no supervisado de dichos naturales, dado por características como su enorme volumen, permanente actualización respecto de las alteraciones del lenguaje, y bajo costo, en tiempo y recursos, en cuanto a los mecanismos tradicionales de construcción de *corpus* para esas tareas de aprendizaje. Se presentan las estrategias incorporadas al sistema con el fin de maximizar el aprovechamiento de los recursos de *hardware* y así reducir los tiempos de extracción, al igual que se presentan las características de extensibilidad para los formatos soportados, y adaptabilidad respecto a la manera como el sistema limpia los contenidos para obtener muestras de lenguaje natural puras. Al final del artículo se presentan los resultados experimentales obtenidos con uno de los dominios de contenido en español más grande de Internet: es.wikipedia.org, a través de los cuales se concluye sobre la validez y aplicabilidad de un *corpus* extraído directamente de la Internet para un eventual proceso de aprendizaje de morfología o sintaxis.

Palabras clave: *corpus web, crawler*, aprendizaje no supervisado de lenguajes, programación concurrente.

ABSTRACT

Internet content, used as text corpus for natural language learning, offers important characteristics for such task, like its huge volume, being permanently up-to-date with linguistic variants and having low time and resource costs regarding the traditional way that text is built for natural language machine learning tasks. This paper describes a system for the automatic extraction of large bodies of text from the Internet as a valuable tool for such learning tasks. A concurrent programming-based, hardware-use optimisation strategy significantly improving extraction performance is also presented. The strategies incorporated into the system for maximising hardware resource exploitation, thereby reducing extraction time are presented, as are extendibility (supporting digital-content formats) and adaptability (regarding how the system cleanses content for obtaining pure natural language samples). The experimental results obtained after processing one of the biggest Spanish domains on the internet, are presented (i.e. es.wikipedia.org). Such results are used for presenting initial conclusions about the validity and applicability of corpus directly extracted from Internet as morphological or syntactical learning input.

Keywords: Web Corpus, crawler, unsupervised language learning, concurrent programming.

Recibido: octubre 30 de 2008
Aceptado: octubre 23 de 2009

Introducción

Las líneas de investigación relacionadas con el procesamiento de lenguaje natural, en particular aquellas que estudian los mecanismos para el aprendizaje no supervisado del lenguaje natural, han tomado una relevancia importante en los últimos años por el interés que despierta tanto a nivel teórico como de aplicación. El principio de la pobreza del estímulo, de Chomsky (1986), y su afirmación de que teóricamente un niño no debería ser capaz de aprender la gramática de su lenguaje nativo dado lo limitado de los ejemplos que recibe de la misma –a menos que se cuente con una capacidad innata sólo existente en los humanos–, postulado de la teoría del nativismo, ha sido una motivación desde el punto de vista teórico de la psicología y neurolingüística para proponer modelos de aprendizaje de lenguajes a realizar por una máquina para, por un lado, poder demostrar la validez o invalidez del nativismo (Clark, 2002), y por otro, aproximarse a nuevas hipótesis de cómo se realiza el aprendizaje de los lenguajes (Parekh y Honavar, 2000). Por otra parte, a nivel de aplicación, la problemática que plantea el volumen de información disponible actualmente en Internet, al ser cada vez más complejo encontrar información relevante más allá de la obtenida con la coincidencia exacta de palabras, ha motivado investigaciones como las de la construcción automática de modelos de representación de conocimiento –ontologías– de cuerpos de texto disponibles en la red (Buitelaar *et al.,* 2005; Navigli *et al.,* 2003; Zhou, 2007), como una base para la construcción de la llamada "*web* semántica". Para las estrategias de aprendizaje no supervisado de lenguaje natural, un elemento fundamental es la muestra del lenguaje sobre la cual se van a generalizar, de forma aproximada, sus características (Church y Mercer, 1993). Se ha mostrado que la *web* es una fuente de datos para el análisis del lenguaje natural de una riqueza sin precedentes (Marianne Hundt y Biewer, 2007), y que algoritmos simples, basados

[1] Ingeniero de sistemas, Escuela Colombiana de Ingeniería. M.Sc., en Ingeniería de Sistemas, Universidad Nacional de Colombia. Profesor, Escuela Colombiana de Ingeniería. hfcadavidr@unal.edu.co
[2] Ingeniero de sistemas y M.Sc., en Matemáticas, Universidad Nacional de Colombia. Máster y Ph.D., of Sciences en Matemáticas con concentración en Computer Sciences, Universidad de Memphis, Estados Unidos. Profesor asociado, Universidad Nacional de Colombia. jgomezpe@unal.edu.co

en esta fuente de ejemplos de lenguaje, muchas veces superan el desempeño de aquellos más complejos basados en fuentes de datos más pequeñas –a pesar de que estas últimas son más depuradas– (Keller y Lapata, 2003). Otras motivaciones para el uso de la *web* a manera de *corpus* de texto, para tareas de aprendizaje de lenguajes, son:

-Elementos como las innovaciones léxicas, emergentes a través del tiempo en las diferentes culturas, o las características de las variantes "exóticas" de los lenguajes (el inglés australiano, o el español panameño, por dar algún ejemplo) no se hacen evidentes en las fuentes de ejemplos de lenguaje natural tradicionales (Kilgarriff y Grefenstette, 2003). En este sentido, las evidencias de comunicación dejadas en recursos típicos de Internet como los foros de discusión o los *blogs* representan un material sumamente valioso para la investigación del uso contemporáneo del lenguaje.

-El costo y tiempo que representan la construcción, de forma tradicional, de un *corpus* de texto suficientemente significativo de un lenguaje resulta sumamente alto (Marianne Hudt y Biewer, 2007), lo que por un lado restringe las posibilidades de experimentar con modelos de aprendizaje usando nuevos *corpus* en un tiempo razonable (por ejemplo, con nuevos lenguajes, variantes culturales o nuevos dominios temáticos), y por el otro da pie a la reutilización de *corpus* cada vez más antiguos.

-A pesar de que los contenidos de la *web* en su mayoría no tienen control sobre su calidad en cuanto al correcto uso del lenguaje, y pueden contener toda una variedad de errores, su enorme volumen permitiría detectarlos tomando una muestra suficientemente grande, y descartando aquellos elementos menos frecuentes. Adicionalmente, se cuenta con repositorios de contenidos digitales que en cierta medida sí garantizan un uso apropiado del lenguaje, a través de la colaboración masiva de sus mismos usuarios, de manera que dichos repositorios pueden considerarse como fuentes confiables de construcción rápida de *corpus*.

Este artículo presenta un sistema para la construcción automática de *corpus* de texto y el muestreo de palabras y frases a partir de contenidos de la *web*, para tareas de análisis o aprendizaje no supervisado de la morfología y la sintaxis de lenguajes naturales. El sistema descrito, entre otras características, cuenta con:

-Un mecanismo genérico de paralelización y sincronización de tareas utilizado en diferentes puntos del sistema.

-Extracción recursiva de contenidos de un dominio. A partir de la URL de un dominio, el sistema identifica todos los recursos relacionados directa o indirectamente a través de hipervínculos.

-Soporte extensible para múltiples formatos. Puede extraer muestras de lenguaje natural de contenidos disponibles en la red con formatos diferentes al tradicional HTML, y permite la inclusión transparente de nuevos extractores dentro del sistema.

-Manejo estadístico de los elementos lingüísticos encontrados, para tareas de detección de ruido y tareas relacionadas.

Inicialmente se describe la solución propuesta, la cual es especificada más en detalle en las secciones subsecuentes: estrategia general para la paralelización de tareas; extracción de URL del dominio y tipos de contenido; construcción de componentes expertos en extracción; extracción y persistencia de los cuerpos de texto. Finalmente, se muestran los experimentos y resultados obtenidos con dominios de acceso público, y las variaciones en los resultados de acuerdo con los cambios en los parámetros del sistema.

Descripción general del sistema

El sistema desarrollado tiene como propósito construir, en poco tiempo, grandes cuerpos de texto para realizar tareas de aprendizaje de lenguajes naturales. A diferencia de las herramientas presentadas en trabajos preliminares como el de Kehoe (2002) y Gelbukh y Sidorov (2006) donde a través de buscadores *web* y palabras claves se arman *corpus* de contextos o temas específicos, la herramienta aquí descrita se enfoca en la extracción de muestras de lenguaje de dominios *web* completos, ya que se busca obtener y analizar variantes regionales de un mismo lenguaje, independientemente del contexto o temática. Sobre la premisa de que hoy en día es fácil encontrar dominios *web* (portales, *wikis*, etc.) construidos por personas de una misma región o cultura, los *corpus* generados con este enfoque permitirán el análisis de los elementos lingüísticos como la morfología y la sintaxis particulares de dichas regiones.

A nivel funcional, la herramienta propuesta cuenta con dos características fundamentales: desempeño y extensibilidad. En cuanto a desempeño, se buscó que el proceso de extracción pudiera aprovechar al máximo los recursos de ancho de banda disponibles y de esta manera reducir los tiempos de construcción del *corpus*. En cuanto a extensibilidad, se buscó que el sistema pudiera adaptarse para extraer muestras de lenguaje natural de nuevos tipos de formatos digitales, a través de un esquema de componentes. En la figura 1 se presenta el funcionamiento general del sistema. Se parte de la raíz de un dominio para extraer, con un nivel de profundidad dado, los enlaces relacionados con dicho dominio (los contenidos de éste deben ser uniformes en su lenguaje, si se quieren obtener buenos resultados en las tareas de aprendizaje no supervisado). Posteriormente, el sistema identifica los tipos de contenido de cada enlace encontrado y localiza al componente más adecuado para su manipulación. Finalmente, y de forma concurrente, cada uno de estos componentes extrae, filtra y hace persistentes las muestras del lenguaje natural extraídas. Durante el proceso de persistencia se realizan cálculos de frecuencias, con el fin de poder realizar labores posteriores de eliminación de ruido.

Patrón de ejecución concurrente de tareas independientes

Durante el diseño del proceso de extracción de muestras de lenguaje natural de un dominio, se identificó una problemática común para varias de las etapas de dicho proceso: las tareas de alta latencia, independientes entre sí, que requieren la sincronización de su terminación para pasar a una siguiente etapa del proceso. Por ejemplo, la etapa de identificación de tipo de contenido MIME[3] requiere, para cada URL a explorar, conectarse al respectivo servidor, efectuar una petición de encabezado, y procesar la respuesta para identificar dicho tipo. Como el tiempo de ejecución de esta tarea depende del tiempo de respuesta de los servidores el cual en ocasiones puede ser relativamente alto, realizarla de forma secuencial desaprovecharía las capacidades de cómputo y de ancho de banda de la máquina y de la red donde se corra ese proceso.

Otro ejemplo, es la tarea de extracción de muestras de lenguaje como tal. Esta tarea, dado que requiere extraer la totalidad de los contenidos de cada dirección, tiene una latencia aún más alta, lo que crea un cuello de botella para las tareas de procesamiento intensivo que le siguen, donde se incluyen el procesamiento del contenido y la actualización de la información estadística del len-

[3] Multipurpose Internet Mail Extensions.

guaje. Para tener una solución genérica de ejecución concurrente y sincronización de los diferentes puntos funcionales que requieren la ejecución de múltiples tareas de alta latencia, se definió e implementó un patrón de diseño para el modelo de ejecución descrito en la figura 2, donde se ejecuta una tarea global compuesta por una serie de tareas independientes entre sí (es decir, donde la tarea global finaliza sólo hasta que la última tarea atómica se ejecute), y donde el número máximo de procesos a ejecutarse simultáneamente se puede ajustar, independientemente del número de tareas a realizarse, y sin afectar el cumplimiento de la totalidad de dichas tareas. Este patrón es un nuevo elemento para el conjunto de patrones de procesamiento en paralelo disponible en la literatura (Mattson *et al.,* 2004), el cual tiene como principal beneficio permitir la creación de esquemas de sincronización por barrera (Krishnamurthy y Yelick, 1995) de forma transparente para quien desarrolle algoritmos paralelos.

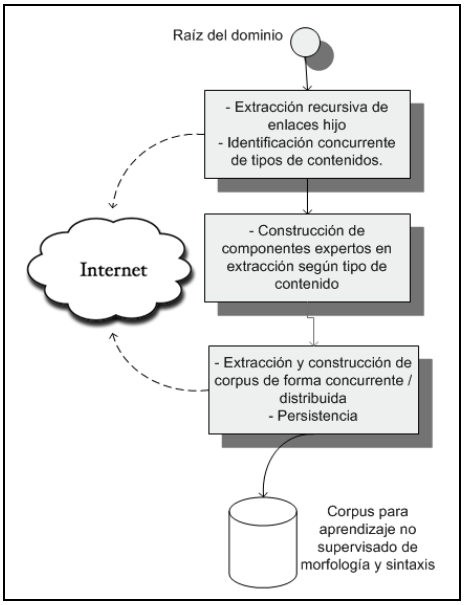

Figura 1. Funcionamiento general del sistema

La idea general del patrón se puede revisar en la figura 3: una vez se da el control al proceso global (proceso concurrente), se crea un conjunto de N hilos, los cuales toman, en la medida que se encuentren disponibles, cada una de las tareas a realizar. A continuación, el proceso global entra en suspensión. Las tareas notifican cuándo han terminado su ejecución al monitor de ejecución, el cual a su vez lleva el control de cuantas tareas han sido culminadas. Finalmente, cuando el monitor identifica que la totalidad de tareas han sido finalizadas, notifica al proceso global para que se reanude y retorne el control de la ejecución a quien lo invocó (Figura 4).

Extracción de enlaces hijo y de tipos de contenido

Para la extracción recursiva de los enlaces relacionados con una determinada URL raíz, con una profundidad h dada (Figura 5), se hizo uso del núcleo del *crawler* desarrollado en el proyecto Sphinxs (Miller y Bharat, 1998), el cual resultó ser muy eficiente y robusto para dicha tarea. Cabe resaltar que, como herramienta tipo *crawler*, la única funcionalidad para la que se le pudo reutilizar fue la de extracción de URL, pues por lo demás esta herramienta está enfocada, al igual que los *crawlers* tradicionales, en descargar localmente copias de los documentos disponibles en la red, y a lo sumo indexar a partir de las palabras claves definidas, exclusivamente para los documentos HTML. Una vez se construye la secuencia de objetos que representan todas las tareas de extracción, y siguiendo el mecanismo de ejecución concurrente descrito, a cada uno de éstos se les delega la responsabilidad de identificar su tipo de contenido (usando la convención MIME), para proveer la información necesaria que permita la construcción de un componente experto a quién delegarle la tarea de extracción y construcción de la muestra de lenguaje natural de dicha dirección.

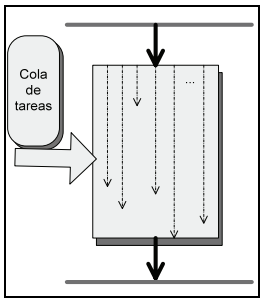

Figura 2. Escenario de aplicación del patrón de ejecución concurrente propuesto: se tiene un número arbitrario de tareas, y se requiere componer una operación global que ejecute dichas tareas con un número de procesos concurrentes parametrizable y que finalice sólo hasta que la última tarea atómica sea terminada.

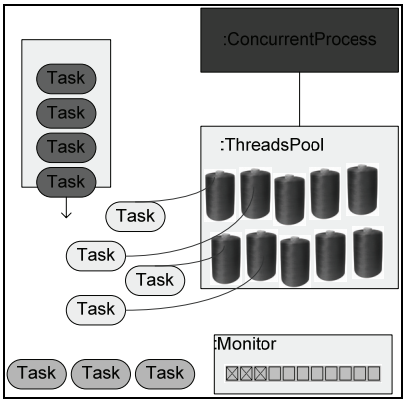

Figura 3. Ejecución concurrente y sincronizada de tareas independientes: el proceso global se detiene mientras el conjunto de N hilos ejecuta el conjunto de tareas. El monitor es notificado por las tareas cuando éstas han terminado, y lleva el control de cuál es el número de tareas restantes.

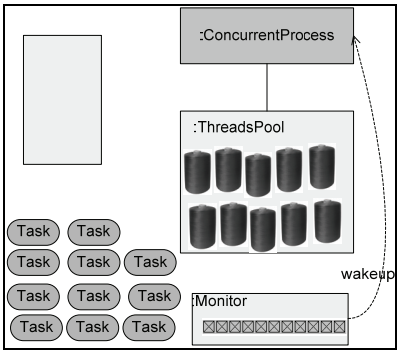

Figura 4. Ejecución concurrente y sincronizada de tareas independientes: una vez se ha ejecutado la última tarea, el monitor notifica a la tarea global para que reanude el control y continúe la ejecución del programa.

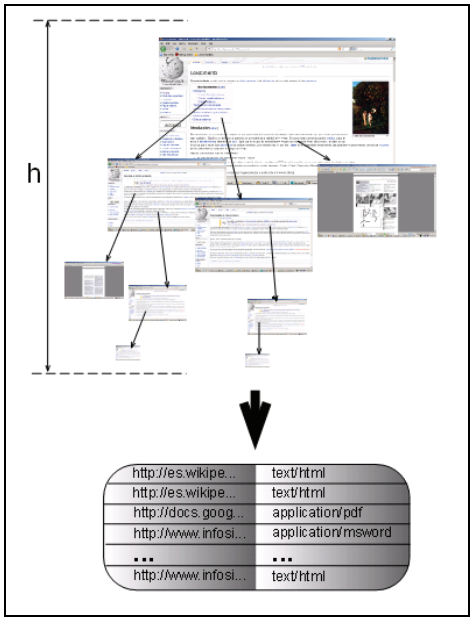

Figura 5. Construcción de la lista de tareas de extracción a partir de la inspección recursiva de los hipervínculos del dominio.

Construcción de componentes expertos en extracción

El patrón planteado, tal como se ve en el diagrama de la figura 6, se definió de tal manera que pueda aplicarse a cualquier tarea que se desee. En este caso, para habilitar las tareas de identificación de tipos de contenido (MIME) y su extracción para ejecutarse en paralelo y sincronizar la terminación de dichas tareas, bastó con definir la lógica de dichas tareas a través de la creación de subclases de la clase Task (en el diagrama MIMETypeExtractionTask y ContentExtractionTask).

Una de las características del sistema descrito es la extensibilidad en cuanto a capacidad de manejo de tipos de contenido. Es decir, en el futuro, a medida que se identifiquen nuevos tipos de contenidos a los cuales se les pueda extraer muestras de lenguaje natural (por ejemplo, medios audiovisuales), basta con desarrollar un componente con los nuevos mecanismos de extracción para que el sistema lo integre de manera transparente. Para tal propósito, se construyó un modelo de componentes de extracción de contenidos de medios digitales extensible, a partir de una metáfora de fábrica de extractores (patrón fábrica abstracta). Este modelo de fábrica al iniciarse realiza un proceso de introspección sobre todo el conjunto de clases públicas en tiempo de ejecución, identifica cuáles son capaces de manipular contenidos digitales (clases que cumplan con la interfaz DigitalMediaTextExtractor (Figura 7), y deja registrado dicho componente con el tipo de contenido que puede manejar, de manera que durante el proceso de extracción concurrente descrito, se tenga acceso inmediato a los componentes expertos en extracción, a medida que se identifiquen los tipos de contenido a extraer.

Extracción y construcción concurrente / distribuida de corpus y persistencia

Como se ve en la figura 8, a partir del conjunto de objetos "expertos" en extracción, y el modelo de ejecución concurrente descrito anteriormente, se inicia finalmente el proceso de extracción de muestras del lenguaje natural. El principal inconveniente en este proceso es que los medios digitales a los cuales se les extrae su contenido, tales como las páginas HTML o documentos de procesadores de texto en línea, llevan incrustados muchas veces una enorme cantidad de elementos adicionales al texto, como imágenes, hipervínculos o metadatos, lo cual genera una cantidad significativa de ruido para las muestras extraídas. Dado que el contar con ejemplos válidos de sentencias del lenguaje es fundamental para tareas tales como el análisis sintáctico, se integró al modelo un esquema de filtros encadenados, los cuales se encargan de depurar el cuerpo de texto hasta lograr que éste se componga sólo de sentencias y signos de puntuación válidos. El modelo de componentes descrito es igualmente aplicado, de tal forma que la incorporación de procesos de filtrado adicionales, y su encadenamiento, sólo requiera la definición de la lógica de filtrado (Figura 9). En esta etapa la herramienta incorpora un filtro de eliminación de símbolos inválidos (en el contexto de las sentencias de lenguaje natural tradicionales), y otro de unificación de signos de puntuación contiguos.

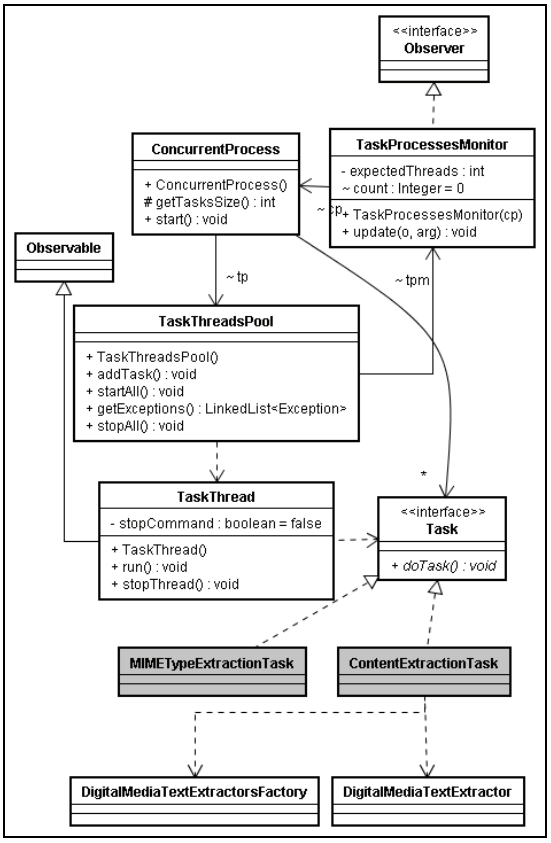

Figura 6. Diagrama de clases del patrón de ejecución concurrente y sincronizada de tareas independientes.

Resultados

Las herramientas propuestas en trabajos preliminares para el uso de la *web* como *corpus* (Kehoe, 2007; Gelbukh y Sidorov, 2006) no muestran análisis de desempeño, que es uno de los énfasis de esta propuesta, de manera que a los resultados aquí presentados no se les puede hacer análisis comparativos. Para la pruebas se utilizó un canal dedicado (sin más aplicaciones consumiéndolo) de 600 Mbs, y un computador Intel Core 2 Duo de 2GHz con 2GB de memoria RAM, con 1GB dedicado al *heap* de la máquina virtual de Java. Extracción de hipervínculos para el idioma español. Como fuente de ejemplos del lenguaje español se escogió el domi-

nio en español de *Wikipedia* (http://es.Wikipedia.org), por su enorme volumen de datos difícil de encontrar para lenguajes diferentes al inglés, de cerca de 120.000 páginas. Partiendo de la dirección raíz del portal en mención, se lograron extraer, a través del seguimiento de hipervínculos (con una profundidad máxima de 10), 76.000 enlaces. La totalidad de estos enlaces fueron procesados por una máquina dedicada, en aproximadamente dos días, generando un *corpus* de 690 MB y 44,5 millones de palabras, correspondientes a un conjunto de 370.000 palabras.

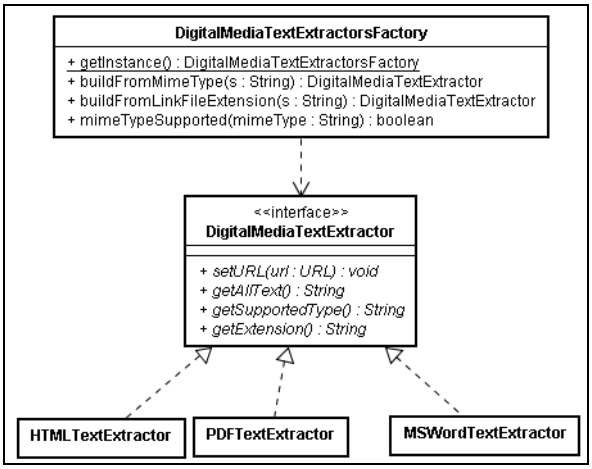

Figura 7. Modelo de fábrica de componentes de extracción.

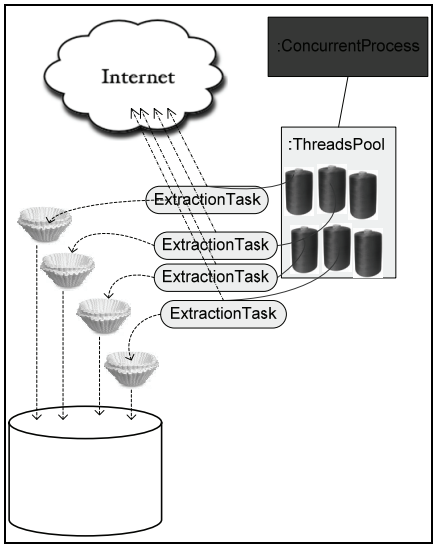

Figure 8. Extracción concurrente de contenidos y filtrado múltiple de éstos.

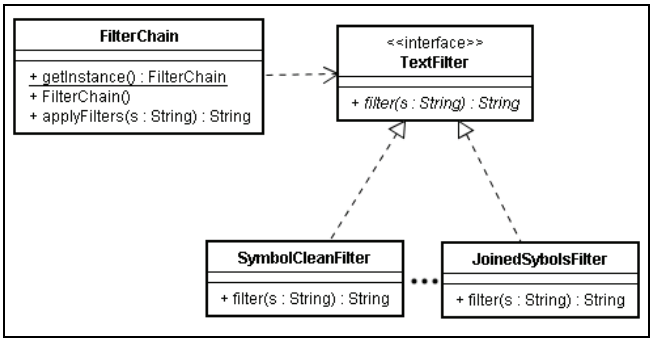

Figura 9. Modelo de filtros encadenados.

Observaciones

Como un soporte adicional a las técnicas que utilicen los cuerpos de texto generados a través de esta herramienta, la persistencia se realiza en un modelo relacional, haciendo persistente de forma independiente las palabras, las frases (identificadas a través de los signos de puntuación), y en un archivo de texto el *corpus* consolidado. La persistencia de las palabras, apoyados en el motor de base de datos, garantiza la no duplicidad de los vocablos almacenados, y adicionalmente lleva un registro del número de ocurrencias. Dado que el sistema no puede garantizar que no se incluyan palabras que no correspondan al lenguaje (sino por ejemplo a metadatos HTML o de otros tipos de contenido dejados accidentalmente como parte del cuerpo de texto), el control de dicha frecuencia podría servir para hacer un descarte de palabras (aquellas que tengan una frecuencia demasiado baja, en proporción al tamaño del *corpus* de texto extraído).

Se hizo la inspección de las palabras y sus frecuencias del *corpus* extraído cuando apenas se habían procesado 300 enlaces (un *corpus* de aproximadamente 2 Mb y 180.000 palabras), obteniendo como las más frecuentes las mostradas en el cuadro I. Como se observa, algunas preposiciones resultan predominantemente más frecuentes que el resto de vocablos, al igual que ciertas palabras recurrentes en *Wikipedia* (aunque vale resaltar que son válidas dentro del lenguaje). A pesar de los resultados anteriores, lo más importante es lo identificado en el cuadro II, donde se observa que palabras inválidas para el lenguaje español como "*ttulosi*" o "*internoregresa*" tienen una mayor frecuencia que términos válidos como "gramática" y "almacenaje". Esto hace evidente la necesidad, en el caso particular de cuerpos de texto obtenidos de Internet, de manejar *corpus* de dimensiones muy altas, para lograr una tendencia donde el volumen de ejemplos correctos del lenguaje supere significativamente el ruido existente en los medios digitales y se pueda hacer una filtración de palabras válidas dada su frecuencia (sin perder vocablos importantes que sean poco frecuentes).

En el cuadro III se muestran de nuevo las frecuencias de las palabras obtenidas, luego del procesamiento de 2.000 enlaces, y consultando específicamente las relacionadas con "gramática", la cual, en el experimento anterior, habría podido considerarse como ruido. En este punto se hace evidente que, entre mayor sea la muestra del *corpus* extraído, mayor será la proporción entre la frecuencia de las palabras correctas y las incorrectas del lenguaje.

Cuadro I. Palabras y sus frecuencias obtenidas al procesar 300 enlaces de es.wikipedia. org.

No.	WORD	FREQUENCY
1	de	33779
2	la	14427
3	en	11363
4	el	9613
5	y	8134
6	a	6481
7	del	5137
8	que	5067
9	los	4902
10	por	3664
11	un	2990
12	una	2852
13	las	2689
14	con	2521
15	enciclopedia	1302
16	este	794
17	sin	665
18	o	634
19	otros	633

Cuadro II. Palabras y sus frecuencias obtenidas al procesar 300 enlaces de es.wikipedia.org. Posiciones de las menos frecuentes.

63	giro	13
64	ciertas	12
65	componentes	11
66	ordenador	10
67	creative	7
68	cesin	5
69	ttulosi	5
70	internoregresa	5
71	corregirlo	5
72	apunte	5
73	licencias	4
74	infraestructuras	4
75	creaciones	3
76	receptores	2
77	libreradio	2
78	didáctico	2
79	derechoscomo	1
80	incondicionalart	1
81	licenciasanlogas	1
82	contenidocomo	1
83	originalliberacin	1
84	libreinfraestructuras	1
85	librelicencia	1
86	gnulicencias	1
87	tambinportalsoftware	1
88	librecontenido	1
89	libremsica	1
90	externoscreative	1
91	webel	1
92	culturacontenido	1
93	significarla	1
94	guarda	1
95	almacenaje	1
96	gramática	1

Cuadro III. Frecuencias de palabras relacionadas con "gramática", y el mejoramiento de la evidencia de cuáles son las correctas.

WORD	FREQUENCY
gramática	59
gramaticales	32
gramatical	29
gramaticacuando	6
gramaticalmente	5
gramáticas	4
gramaticapudiendo	3
esperantogramatica	2
islandesgramatica	2
gramaticailr	2
gramaticalesy	2
gramaticalvariando	2
gramaticaleslenguas	2
gramaticalpor	2
libregramatica	2
gramaticalesalfabeto	2
gramaticalcaracteristicas	2
gramaticalidad	1
isticogramatica	1
gramaticalesherramientas	1
gramaticalespor	1
gramaticalesindica	1
ogramatica	1
principalgramatica	1
italianogramatica	1
inglesgramatica	1
gramaticalesestos	1
idogramatica	1
hebreogramatica	1

Desempeño

La tasa promedio de extracción, con las características de infraestructura descritas, 200 procesos concurrentes, y los tiempos de respuesta propios del dominio *es.Wikipedia.org*, es de 12 Mb por hora, pero para hacer más evidentes las mejoras de desempeño de las estrategias de extracción concurrentes aplicadas se monitoreó el tráfico de la red mientras se realizaba el proceso de extracción (Figura 10). En el monitoreo 1 hay un límite de 10 procesos de extracción concurrente, mientras que en el 2 hay 200. Como se puede ver en la imagen, la tasa promedio de ancho de banda consumido para tareas de extracción aumenta en más de 13 veces.

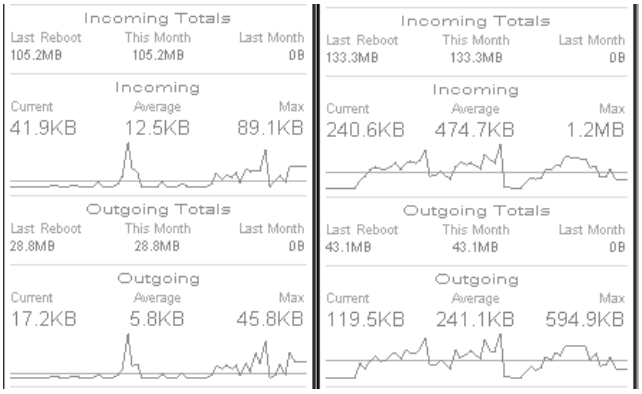

Figura 10. Diferencias de desempeño alternando entre 10 y 200 como número máximo de procesos de extracción concurrente.

Trabajo futuro

A partir de los resultados obtenidos, en el futuro este trabajo se aplicará y extenderá en los siguientes aspectos:

El *corpus* obtenido con *Wikipedia* en español será utilizado con una técnica de aprendizaje no supervisado de lenguajes naturales, que permita identificar elementos de la morfología y de la sintaxis del español. Una vez se mida el desempeño de esta técnica con el *corpus* obtenido, se podrán hacer nuevos experimentos con *corpus* de diferentes tamaños (generados con esta herramienta), de manera que sea posible determinar una función aproximada de desempeño de aprendizaje respecto del tamaño del *corpus*.

En cuanto a la herramienta como tal, en un futuro se incorporará una arquitectura distribuida para la extracción. De esta manera, la herramienta podrá hacer uso de varias máquinas, cada una con su ancho de banda, para la construcción, aún más eficiente, de *corpus* de texto.

Conclusiones

Desde hace tiempo se han documentado las posibilidades y ventajas que tendría la *web* vista como un repositorio enorme de cuerpos de texto. Una herramienta como la presentada permitirá, a quienes trabajen en el área de procesamiento de lenguajes naturales, obtener de forma ágil *corpus* de diferentes variantes de un mismo idioma, con tan sólo identificar dominios *web* construidos por personas pertenecientes a una determinada región o cultura.

Los esfuerzos dentro del área de procesamiento de lenguaje natural enfocados a la construcción de soluciones eficientes para la extracción de muestras del lenguaje, tal como se plantea en este trabajo, a largo plazo tendrían una aplicación dentro de los modelos de *web* semántica que a futuro se planteen, pues el mecanismo tradicional de los *crawlers*, donde simplemente se indexan y (en algunos casos) se replican documentos en línea, no sería suficiente para alimentar a los métodos de construcción automática de representaciones de conocimiento de los contenidos disponibles en línea.

A pesar de la enorme cantidad de ruido y errores existentes en los contenidos textuales de Internet, se pudo comprobar que cuanto mayor es el volumen del *corpus* construido más fácil será identificar los elementos válidos del lenguaje, usando como criterio diferenciador la frecuencia de las palabras. Es decir, con una herra-

mienta como la aquí presentada será posible, sin intervención humana, construir los lexicón de las diversas variantes de un idioma, tarea que antes resultaba impensable.

El sistema presentado es sólo un ejemplo de lo que se puede hacer con el enorme volumen de información textual que está acumulándose en Internet. A partir de esta información puede que se logren identificar, para cada idioma, más elementos y paradigmas lingüísticos que aquellos disponibles en la literatura. Esto será sumamente valioso para los lingüistas y para quienes trabajen en procesamiento de lenguajes naturales

Bibliografía

Chomsky, N., Knowledge of Language: Its Nature, Origin, and Use., Praeger, 1986.

Clark, A., Unsupervised Language Acquisition: Theory and Practice., Tesis presentada a la Universidad Génova, para optar al grado de Doctor of Philosophy, Dicembre, 2002.

Parekh, R., Honavar, V., Grammar inference, automata induction, and language acquisition., 2000.

Buitelaar, P., Cimiano, P., Magnini, B., Ontology Learning from Text: Methods., Evaluation and Applications, Vol. 123 of Frontiers in Artificial Intelligence, IOS Press, 2005.

Navigli, R., Velardi, P., Gangemi, A., Ontology learning and its application to automated terminology translation., IEEE Intelligent Systems, Vol. 18, No. 1, 2003, pp. 22-31.

Zhou, L., Ontology learning: state of the art and open issues., Information Technology and Management archive, Vol. 8 , No. 3, September, 2007, pp. 241-252.

Church, K. W., Mercer, R. L., Introduction to the special issue on computational linguistics using large corpora., Comput. Linguist., Vol. 19, No. 1, 1993, pp. 1-24.

Marianne Hundt, N. N., Biewer, C., Corpus Linguistics and the Web., Language and Computers 59, Kenilworth: Rodopi, 2007.

Keller, F., Lapata, M., Using the web to obtain frequencies for unseen bigrams., Comput. Linguist., Vol. 29, No. 3, 2003, pp. 459-484.

Kilgarriff, A., Grefenstette, G., Introduction to the special issue on the web as corpus., Computational Linguistics, Vol. 29, 2003, pp. 333-347.

Miller, R. C., Bharat, K., Sphinx: a framework for creating personal, site-specific web crawlers., in WWW7: Proceedings of the seventh international conference on World Wide Web 7, (Amsterdam, The Netherlands, The Netherlands), Elsevier Science Publishers B. V., 1998., pp. 119-130.

Kehoe, A. R., Webcorp: Applying the web to linguistics and linguistics to the web., in WWW2002 Conference, Honolulu, Hawaii, 2002.

Kehoe, A. M. G., New corpora from the web: making web text more 'text-like'., in Towards Multimedia in Corpus Studies, electronic publication, University of Helsinki, 2007.

Mattson, G., Sanders, B. A., Massingill. B. L., Patterns for Parallel Programming., Addison-Wesley Professional, 2004.

Krishnamurthy, A., Yelick, K., Optimizing parallel programs with explicit synchronization., SIGPLAN Not. 30, 1995, pp. 96-204.

Gelbukh, A., Sidorov, G., Procesamiento automático del español con enfoque en recursos léxicos grandes., IPN, Mexico, 2006.