Procesamiento del Lenguaje Natural, núm. 30 (2003), pp. 75-87 recibido 1-03-2003; aceptado 20-05-2003

DAWeb: Un descargador y analizador morfológico de páginas web

Octavio Santana Suárez
Universidad de Las Palmas de Gran Canaria
Edificio de Informática y Matemáticas
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
osantana@dis.ulpgc.es

Zenón J. Hernández Figueroa
Universidad de Las Palmas de Gran Canaria
Edificio de Informática y Matemáticas
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
zhernandez@dis.ulpgc.es

Gustavo Rodríguez Rodríguez
Universidad de Las Palmas de Gran Canaria
Edificio de Informática y Matemáticas
Campus Universitario de Tafira
35017 Las Palmas de Gran Canaria
grodriguez@dis.ulpgc.es

Resumen: DAWeb es una aplicación informática desarrollada como parte de un proyecto consagrado a la realización de herramientas capaces de facilitar el aprovechamiento para la realización de estudios lingüísticos del enorme caudal de información que ofrece Internet. Es una herramienta orientada al análisis morfosintáctico de grandes volumenes de información —dominios completos— a los que se accede por una o varias URL de partida. Está dotada de una sencilla interfaz que permite establecer las acciones pertinentes sobre la información accedida. Como resultado de los análisis realizados, se genera un conjunto estructurado de datos que pueden estudiarse con facilidad.
Palabras clave: morfología, análisis de textos, Internet, lingüística computacional

Abstract: DAWeb is a computer application developed as part of a project oriented to produce tools designed to get at the big flow of linguistic information of Internet documents. It is a tool for morphosyntactic analysis of great volumes of information —whole domains— reached by its URLs. The simple application interfaz facilitates the configururation of how to accessing and analysing the information obtained. The results of the process are organized in a suitable way for posterior research.
Keywords: morphology, text analysis, Internet, computational linguistic

1 Introducción

El presente trabajo es proyección natural de los esfuerzos realizados por el Grupo de Estructuras de Datos y Lingüística computacional de la Universidad de Las Palmas de Gran Canaria en los últimos años. Estos trabajos se han centrado en el ámbito de la lingüística computacional y han dado lugar, entre otros resultados, al desarrollo de herramientas de reconocimiento y gestión morfológica, algunas de las cuales se encuentran disponible para su utilización en línea en la página web del grupo (http://gedlc.ulpgc.es). Se propone la utilización de dichas herramientas como parte de nuevas aplicaciones cuyo objetivo es obtener provecho del enorme caudal de información lingüística que supone Internet.

DAWeb se orienta al estudio conjunto de grandes vólumenes de documentos de forma desasistida y adopta el formato de un descargador de páginas con la diferencia de que en vez de bajar las páginas que accede, las analiza y almacena sólo los resultados.

Las modalidades de análisis que puede realizar abarcan: (1) la detección de neologismos, entendiendo como tal, en primera instancia, cualquier palabra que las herramientas de reconocimiento morfológico incorporadas no identifiquen —luego habrá que filtrar si se trata de entidades tales como

ISSN 1135-5948

nombres propios, secuencias especiales o incluso simples errores ortográficos—, (2) el estudio del uso de las palabras, por medio de diversas medidas cuantitativas y cualitativas, y finalmente, (3) aspectos cercanos a la sintáxis tales como el estudio de colocaciones léxicas o regímenes preposicionales.

2 Arquitectura de DAWeb

DAWeb se halla estructurado, figura 1, en tres módulos principales: 1) *módulo de configuración*, 2) *módulo de recuperación de documentos* y 3) *módulo de análisis* en línea; se complementa con una aplicación externa —programa mostrador— que se encarga de presentar los resultados —generalmente voluminosos— en formas adecuadas para su estudio eficiente.

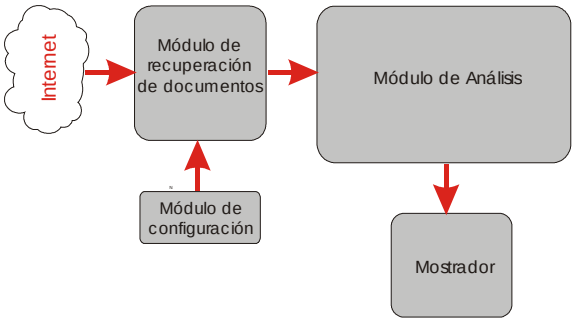

Figura 1: arquitectura de DAWeb

En las siguientes secciones se habla pormenorizadamente de cada uno de estos módulos, se exponen sus funciones, se detallan las interrelaciones que se establecen entre los mismos y se justifican las políticas de funcionamiento adoptadas.

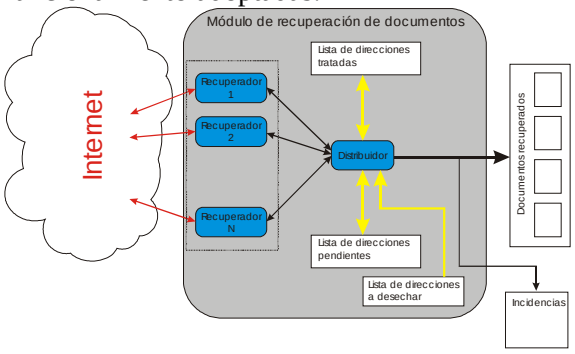

Figura 2: módulo de recuperación de documentos

2.1 El Módulo de recuperación de documentos

El *módulo de recuperación de documentos* está compuesto, figura 2, por un *módulo distribuidor* y un número variable de *módulos recuperadores* que interactúan con Internet.

2.1.1 El módulo distribuidor

El módulo distribuidor se encarga de repartir y coordinar el trabajo entre los recuperadores y de recibir los resultados que obtengan; les da forma y los deja preparados para su entrega al módulo de análisis o a cualquier otro que pudiera realizar algún tipo de tarea con los mismos.

El módulo distribuidor toma direcciones de la lista de "direcciones pendientes" —creada inicialmente por el *módulo de configuración de la recuperación*, submódulo del *módulo de configuración*—, y entrega una a cada recuperador hasta que todos tengan la suya o hasta que la lista de direcciones pendientes esté vacía. A partir del momento en que todos los recuperadores tengan una dirección o todas las direcciones hayan sido asignadas, el quehacer del módulo distribuidor consiste en esperar hasta que alguno de los recuperadores obtenga resultados de su gestión; cuando ocurre, el módulo distribuidor requiere el documento que obtiene el recuperador en el acceso a la dirección encomendada y lo incluye en la lista de documentos recuperados —funciona como una cola de documentos pendientes de analizar—, también interpela al recuperador acerca de la lista de direcciones asociadas a los hiperenlaces del documento conseguido.

Las direcciones que el recuperador se ha encargado de extraer deben confrontarse por triplicado:

1. Con la lista de direcciones pendientes para no duplicar una dirección incluida en la petición inicial u obtenida como resultado de otros accesos.
2. Con la lista de direcciones ya recuperadas, para evitar redundancias en la recuperación.
3. Con los criterios de direcciones desechables —los establece el *módulo de configuración*— para comprobar que constituyen candidatos aceptables de cara a posteriores expansiones de la búsqueda de documentos en curso.

Las direcciones que salven este triple filtro se añadirán a la lista de direcciones pendientes

y contribuirán a engrosar el conjunto de materiales que se obtengan por desarrollo de la petición inicial hasta el límite posible.

Una vez que se ha extraído toda la información que el recuperador es capaz de proporcionar, el distribuidor asigna al recuperador una nueva dirección a partir de la lista de direcciones pendientes y vuelve al estado de espera.

El trabajo del módulo de recuperación de documentos concluye cuando la lista de direcciones pendientes está vacía y ningún recuperador se halla navegando o intentando navegar —circunstancias de las que se apercibe el módulo distribuidor. Si la lista de direcciones pendientes está vacía, pero algún recuperador está ocupado, podría ocurrir que obtuviese algún documento con hiperenlaces, y habría que esperar para saber si el proceso aún debe continuar.

De fracasar el encargo asignado a un recuperador, el *módulo distribuidor* evaluará las circunstancias que han provocado tal situación y optará por volver a intentar la misma dirección con posterioridad o la desestimará por considerar que no va a ser posible acceder a la página —dirección errónea o acceso fallido y se anota en *Incidencias*—; en cualquier caso, si hay direcciones pendientes le entregará una nueva al recuperador malogrado —en ningún caso reintentará la misma dirección inmediatamente, ya que el recuperador lo habrá probado hasta el límite establecido en la configuración antes de decidirse a comunicar su fallo.

El módulo distribuidor genera un informe de *Incidencias* en el que se relacionan: las páginas solicitadas, las páginas obtenidas y las páginas no recuperadas —indica la causa aparente del fallo.

2.1.2 Los módulos recuperadores

Cada módulo recuperador es un gestor de transacciones HTTP (HiperText Transfer Protocol) que se ejecuta en un hilo independiente. En paralelo pueden ocurrir: 1) otras recuperaciones, 2) el distribuidor obtiene datos de los recuperadores que hayan finalizado con éxito su gestión y 3) el analizador procesa documentos recuperados con anterioridad.

Se pretende aprovechar al máximo el tiempo de uso del procesador, de forma que se minimicen los "tiempos muertos" de espera de unas tareas por la finalización de otras.

El trabajo del recuperador se inicia cuando recibe una dirección por parte del distribuidor; el gestor de transacciones HTTP iniciará una petición de conexión con la dirección indicada y pasará al estado de espera mientras llega una respuesta. Si la respuesta llega en forma de establecimiento de la conexión, comienza la transferencia de paquetes de información hasta completar la recepción del documento asociado a la dirección especificada. En caso de que ocurriera un fallo —por ejemplo que no se encuentre el servidor especificado, que se interrumpa la transferencia, etc.— o se agotara el tiempo de espera, el recuperador volverá a intentar la conexión un número limitado de veces antes de reportar un mensaje de error.

Si se logra completar la recepción de un documento, el recuperador emitirá un mensaje para informar de tal circunstancia al distribuidor y quedará a la espera de instrucciones que normalmente consistirán de dos requerimientos del distribuidor. El primero será para que le sean entregadas las direcciones referenciadas por los hiperenlaces del documento recuperado —extraídas por el recuperador—; se desestimarán todas aquellas que conduzcan a enlaces ajenos al dominio cuyo análisis se solicitó en un principio, a menos que se trate de un redireccionamiento —única medida que se estima válida para "podar" la navegación, ya que otras que limitan su profundidad no son suficientemente adaptables a la diversidad de los dominios web. Lo siguiente que solicitará el distribuidor al recuperador será el documento completo —incluye los códigos de formato HTML (HyperText Markup Language), ya que podrían necesitarse según el uso que se fuera a dar al documento y no es trabajo del recuperador eliminar dichos códigos.

Cuando el recuperador ya ha entregado toda la información de que dispone, lo único que queda es esperar una nueva dirección, o bien la señal de desactivación, si el proceso ha concluido.

El número de recuperadores es configurable entre uno y diez —al inicio de un nuevo proyecto se establece por defecto en cinco. El usuario puede cambiar su número en cualquier momento y cuantas veces estime oportuno mientras que el proyecto no esté en ejecución. Con esta configurabilidad se consigue una alta flexibilidad para adaptarse a aspectos tales como las condiciones de la red en un momento

determinado o a las necesidades concretas de un estudio: con un sólo recuperador se asegura un orden determinista en la llegada de los documentos que no variará entre ejecuciones distintas del proyecto a menos que los hiperenlaces sufran modificaciones.

La figura 3 muestra el efecto de la variación del número de recuperadores en el estudio de una web no demasiado voluminosa y de acceso relativamente cercano: la del Grupo de Estructuras de Datos y Lingüística Computacional del Departamento de Informática y Sistemas de la ULPGC. Están representados el tiempo de descarga y el tiempo total que dura la ejecución del proyecto —incluye el de descarga y el de análisis de documentos. Dado que el analizador funciona en un hilo de ejecución separado, los tiempos no son aditivos —se solapa la descarga de un documento con el análisis de otro.

Se observa que cuando el número de recuperadores es bajo —menor que cinco—, el tiempo total coincide en la práctica con el de descarga; se debe a que la recuperación es altamente lineal: todo documento que llega se analiza inmediatamente, pero cuando este análisis termina y aún no han llegado otros documentos el analizador debe quedar en situación de espera —en la que pasa la mayor parte del tiempo. La adición de un nuevo recuperador tiene un efecto drástico en la reducción del tiempo de descarga, y en consecuencia, del tiempo total de ejecución.

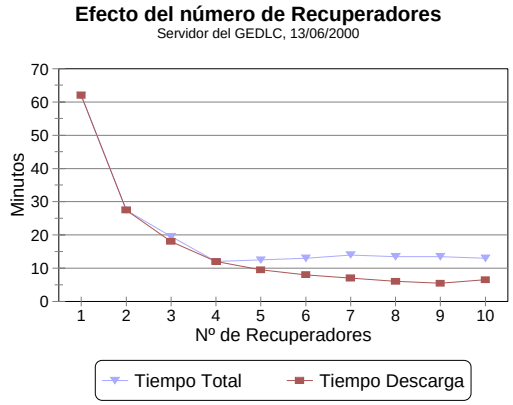

Figura 3: efecto del número de recuperadores

Cuando el número de recuperadores es superior a cinco, un recuperador más no mejora sustancialmente el tiempo de descarga; la competición de un mayor número de hilos entraña un aumento del tiempo —los recursos empleados en la concurrencia empiezan a consumir los tiempos muertos que su gestión intentaba repartir entre los recuperadores. Dado que la velocidad de análisis es casi constante, un determinado volumen de documentos no podrá analizarse en un tiempo inferior al resultante de dividir la cantidad de palabras que contengan por el número de palabras que el analizador es capaz de resolver por unidad de tiempo; tal barrera no puede rebajarse por medio de los recuperadores que, al fin y al cabo se mueven en otra dimensión; en consecuencia, con suficiente número de recuperadores, los documentos deben esperar para ser tratados por un analizador que no está nunca ocioso y el tiempo total viene condicionado por el tiempo de análisis.

Ya que el freno parece ponerlo la velocidad de análisis, podría pensarse en disponer de más de un analizador en paralelo, pero tal disposición carece de sentido; por actuar en la máquina local y a su máxima velocidad, el analizador no deja tiempos muertos significativos que pudieran aprovecharse para mejorar el rendimiento como ocurre en el caso de la recuperación, que sí depende de factores externos no controlables localmente. Las mejoras que puedan obtenerse en cuanto a velocidad de análisis han de conseguirse por otras vías: principalmente, por la optimización del uso del analizador en la dirección de aprovechar el trabajo ya procesado gracias a la frecuente repetición de palabras en cualquier texto.

2.2 Módulo de análisis de documentos

Como muestra la figura 4, el *módulo de análisis de documentos* está integrado por un total de ocho submódulos: 1) de *extracción de texto*, 2) *gestor de análisis*, 3) *selector de palabras*, 4) *gestor de segmentos*, 5) *optimizador de búsquedas morfológicas*, 6) *reconocedor morfológico*, 7) *recuentos* y 8) *entrega de resultados*.

El módulo de recuperación pone en cola los documentos recuperados a la espera de que les llegue el turno para ser analizados. El *módulo de análisis* toma los documentos de esta cola pero deberá eliminar las marcas de formato que confieren al documento de Internet su aspecto visual —tarea de la que se ocupa el *submódulo de extracción de texto*. En este momento entra en juego el *gestor de análisis* para conducir el proceso: mediante invocaciones al *selector de palabras* separa las palabras y echa mano del

reconocedor morfológico —complementado con un módulo *optimizador de búsquedas morfológicas*— para lematizarlas y poder realizar los análisis requeridos. La actividad del *gestor de análisis* está condicionada por la configuración establecida, mediante la cuál se determina qué se analiza —se pueden establecer listas de "palabras vacías", que no se tendrán en cuenta, o "listas de palabras significativas", cuyas apariciones se buscarán—, qué análisis se hacen —reconocimiento de palabras, estadísticas, segmentos, etc.— y qué resultados se proporcionan. Los submódulos de *gestión de segmentos* y *recuentos* intervienen en el análisis: el primero, localiza los segmentos cuya búsqueda se haya dispuesto y el segundo, realiza los cálculos estadísticos pertinentes.

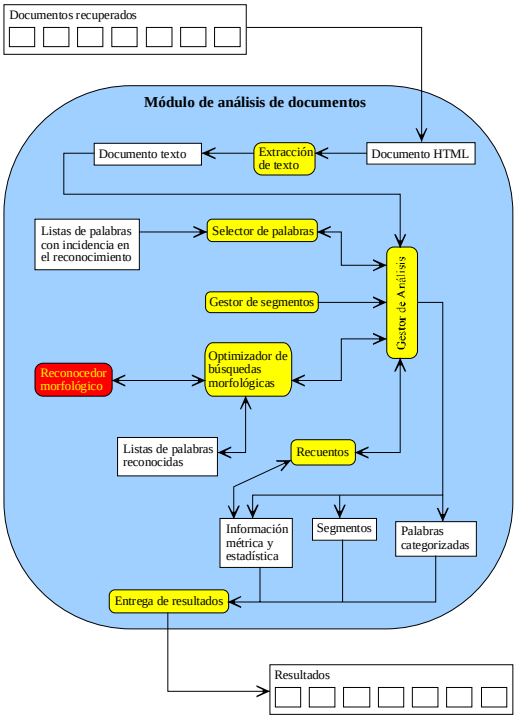

Figura 4: módulo de análisis de documentos

2.2.1 Módulo de extracción de texto

Una página o documento web es básicamente un texto etiquetado utilizando lenguaje HTML. Las etiquetas determinan el aspecto visual que tomará la página al ser mostrada en la ventana de un navegador —fija los colores, la disposición y la estructura del texto y otros muchos detalles. Para el tipo de análisis que se pretende realizar —morfológico y morfoestadístico—, las etiquetas HTML no suelen ser relevantes y cuando aportan información útil lo hacen desde un punto de vista sintáctico o de análisis de las partes del texto. En ningún caso son susceptibles de ser analizadas, dado que no forman realmente parte del texto. Las etiquetas HTML se dividen en: 1) las que definen elementos embebidos en el texto sin romper el flujo del mismo —la etiqueta de estilo de fuente negrita pertenece a esta categoría— y 2) las que definen elementos de ruptura del texto —saltos de línea o cambio de párrafo, entre otras muchas. Las primeras deben eliminarse. Las que definen elementos de ruptura se sustituyen por una marca especial que el gestor de análisis emplea como separador —proporcionan información sobre la estructura del texto que puede resultar útil. Hay que tener en cuenta además, las marcas de caracteres especiales —permiten usar acentos o alfabetos nacionales— que al desaparecer deben sustituirse por el carácter correspondiente.

El módulo de extracción de texto es un analizador sintáctico de una pasada, constituido por un autómata de transición de estados gobernado por la secuencia finita de caracteres del documento. En el estado inicial, E0, el autómata avanza por la secuencia de caracteres mientras no encuentre uno de los símbolos "&" o "<" —el primero señala un código de carácter especial incrustado en el texto y el segundo indica el inicio de una etiqueta HTML. Si se encuentra un "&", se pasa al estado Ea y si el carácter siguiente es "#" —significa que el código incrustado está expresado como un valor numérico— se pasa al estado Ea1 y si no —secuencia alfabética— se pasa al Ea2; en ambos casos se acumula la secuencia de caracteres que sigue hasta encontrar ";" —el símbolo de finalización del código de carácter especial— y se accede con esa secuencia a una tabla que muestra el carácter que corresponde a la misma —a cada uno de estos dos estados le corresponde una tabla diferente ya que con secuencias distintas se debe obtener un mismo carácter—; la secuencia comprendida desde el "&" hasta el ";" —ambos inclusive— se sustituye por el carácter indicado en la tabla correspondiente y se regresa al estado E0; si no se encuentra el ";" se entiende que el carácter "&" debe aparecer tal cual en el texto por no disponerse de una secuencia de sustitución correctamente construida.

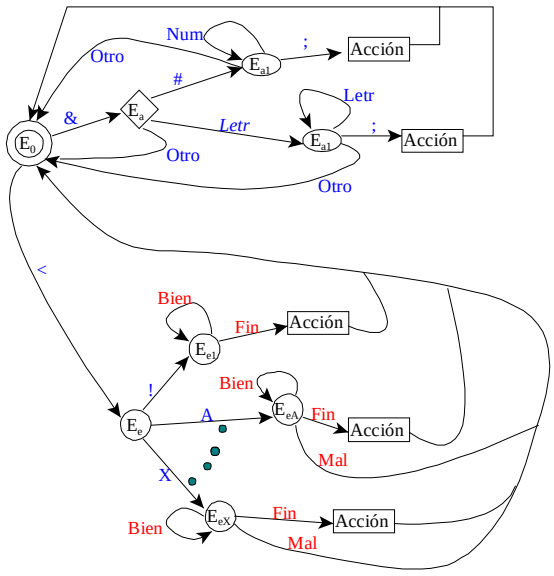

Figura 5: esquema idealizado del autómata

Cuando en el estado E0 se encuentra el símbolo "<" se pasa al estado Ee en el que se inicia el análisis del tipo de etiqueta para determinar la actuación que se va a realizar. Si el carácter que sigue al símbolo "<" es "!", en el estado Ee1 se discrimina entre un comentario —se pasa al estado Ee2 en el que se avanzan todos los caracteres hasta el final del comentario— o una definición de tipo de documento SGML —se elimina en el estado Ee3. Si el carácter que sigue al símbolo "<" es una de las letras con las que comienza una etiqueta —"A", "B", "C", "D", "E", "F", "H", "I", "K", "L", "M", "N", "O", "P", "Q", "S", "T", "U", "V", "X"—, se pasa al estado correspondiente —EeA, EeB, EeC, etc.— en el que se examinan los siguientes caracteres hasta que sea posible determinar de qué etiqueta se trata y qué acción hay que tomar respecto a ella: eliminar sólo la propia etiqueta o eliminar todo el texto comprendido entre ella y la etiqueta de cierre que la empareja —por ejemplo, si la etiqueta es <APPLET> todo lo que se encuentre hasta la etiqueta </APPLET> no es realmente texto, sino una llamada a una aplicación elemental en lenguaje Java y desde la óptica del análisis textual planteado carece de valor. Cuando en el estado Ee, el símbolo "/" sigue al "<" se trata de una etiqueta de cierre que empareja con otra de apertura ya eliminada, sin que se requiera la desaparición del texto encerrado entre el par de etiquetas; la solución radica en forzar una llamada reentrante en el estado Ee con el siguiente carácter para eliminar sin esfuerzo adicional la etiqueta de cierre. Tras eliminar los códigos etiquetados se regresa nuevamente al estado E0 para continuar la exploración del texto.

2.2.2 Módulo selector de palabras

El selector de palabras realiza un proceso de exploración progresiva del texto que proporciona el submódulo limpiador: en la primera invocación avanza desde el principio seleccionando los caracteres hasta formar la primera palabra; en las subsiguientes, retoma la exploración desde el carácter en que se detuvo la vez anterior y avanza hasta completar otra palabra. El proceso se repite mediante peticiones del gestor de análisis hasta que todo el texto haya sido recorrido. Para extraer las palabras se hace distinción entre cinco clases de caracteres: alfabéticos, numéricos, signos de puntuación, terminadores y otros. Algunos símbolos pueden cumplir papeles diferentes dependiendo del contexto en que se encuentren: así un punto puede servir de signo de puntuación —actúa al mismo tiempo de terminador de palabra— o bien como conector —clase otros— en una dirección URL, por ejemplo. Lo que el selector de palabras extrae pertenece a una de tres posibles categorías: secuencia alfabética —las palabras propiamente dichas—, secuencia alfanumérica —formada por letras y números, como los identificadores típicos en informática— y otras secuencias —incluyen caracteres especiales, como el punto en las direcciones URL. Además, acompañan a estas secuencias informaciones concernientes a los signos de puntuación que haya en su entorno.

2.2.3 Módulo de análisis morfológico

Las secuencias de caracteres producidas por el selector de palabras se hallan etiquetadas según su categoría; de ellas sólo las secuencias alfabéticas se consideran palabras por la herramienta de reconocimiento morfológico y en consecuencia, sólo ellas son sometidas a su acción (a las otras las cataloga como no reconocidas).

La herramienta de reconocimiento morfológico es un módulo externo que trabaja tomando una palabra y dando en respuesta la lista de formas canónicas de las que podría provenir y las categorías gramaticales que le serían aplicables. Para obtener este resultado se empieza por descomponer la palabra en sus

posibles pares raíz_terminación, prefijos y, en el caso de los verbos, pronombres enclíticos. La raíz pasa a un módulo de índices que determina su localización para que un módulo de accesos externos compruebe si la raíz admite la terminación, determine a qué flexión o derivación corresponde, deduzca su forma canónica y proporcione su categoría gramatical.

2.2.4 Módulo optimizador de la búsqueda morfológica

El análisis morfológico de los textos obtenidos constituye una parte fundamental de las aplicaciones desarrolladas; en consecuencia, la eficiente utilización del módulo que lleva a cabo dicho análisis tiene una gran influencia en el rendimiento global. El analizador morfológico consta en realidad de dos submódulos que realizan por separado la lematización de formas verbales y formas no verbales. Cada uno de los módulos es capaz de reconocer alrededor de 450 formas por segundo, según pruebas realizadas con un procesador Pentiun II a 300MHz con 128 Mb de memoria RAM; como una palabra puede pertenecer a cualquiera de las dos categorías, es necesario efectuar siempre los dos procesos de reconocimiento, con lo que cabe esperar que la velocidad media obtenida estuviese en torno a la mitad —entre 220 y 250 palabras por segundo.

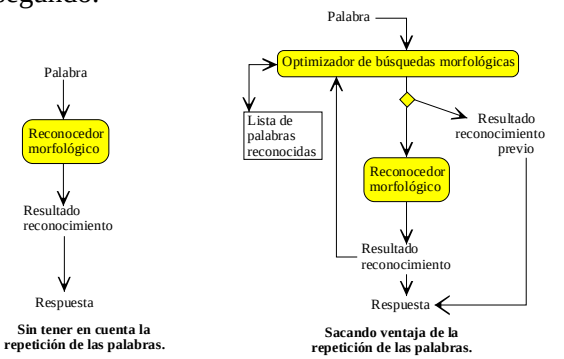

Figura 6: proceso de reconocimiento sin y con optimizador

En un texto, las palabras no se distribuyen de manera uniforme, sino que habitualmente un número muy reducido de ellas se repiten mucho y un grupo no muy grande aparece una sola vez. Cuando una palabra aparece por segunda o sucesivas veces en el texto, se reincide en el esfuerzo de análisis que se realizó en su primera aparición. Dado que hay palabras que se repiten mucho, aparece como una alternativa interesante la posibilidad de evitar las sucesivas llamadas al reconocedor que volverían a obtener los mismos datos de la primera vez. La solución consiste en implantar algún tipo de estructura de acceso rápido en la que se almacenarían los datos resultantes del reconocimiento de cada palabra que aporte el analizador morfológico —tal estructura estaría tanto más justificada cuanto mayor sea el grado de repetición de las palabras. La arquitectura del reconocimiento se modificaría de forma que cuando se obtenga una palabra del texto, no se invoque directamente al analizador morfológico, sino que se consulte previamente la *lista de palabras reconocidas* para averiguar si se ha lematizado con anterioridad; la sobrecarga que representa la consulta de palabras que aparecen por primera vez y que de todas maneras hay que lematizar, quedará ampliamente compensada por la superior velocidad de acceso a la estructura frente al proceso de reconocimiento morfológico.

La estructura de la *lista de palabras reconocidas* es una tabla de dispersión de claves —las palabras— implementada en memoria principal y dimensionada a 199 999 entradas —la diferencia se justifica por sus respectivas orientaciones. Se utiliza una función de dispersión de doble rotación binaria con listas encadenadas separadas para la resolución de colisiones —la longitud promedio inferior a 1,3 nodos por lista es debida tanto a la bondad de la función de dispersión como a la holgura de la tabla.

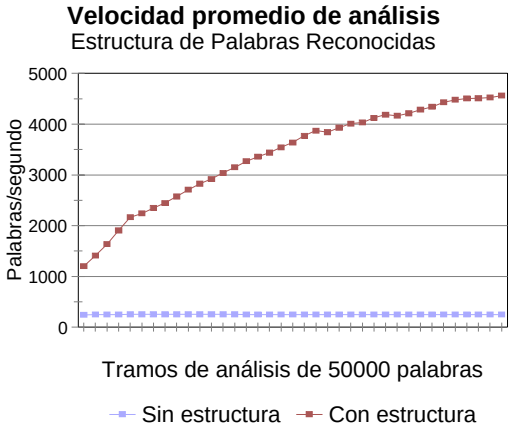

Figura 7: velocidad de análisis

La ganancia queda de manifiesto en la figura 7, donde se observa que la velocidad promedio de reconocimiento llega a experimentar una

mejora superior al 1 800% con los datos de un experimento consistente en analizar 50 documentos seleccionados aleatoriamente de entre los 500 más visitados de la Biblioteca Virtual Miguel de Cervantes —estos 50 documentos se distribuyen en 605 páginas web y contienen un total de 1 815 208 palabras. Al utilizar la *lista de palabras reconocidas*, el claro perfil creciente de la curva que representa la velocidad promedio acumulada se corresponde con la disminución de las llamadas al reconocedor morfológico a medida que se amplía la lista; la pendiente decreciente se explica porque también disminuye la probabilidad de que un tramo aporte palabras nuevas cuyo reconocimiento pueda ser aprovechado en los siguientes tramos.

2.2.5 Módulo de entrega de resultados

El submódulo de *Entrega de resultados* se encarga de elaborar los informes sobre los resultados de los análisis —quedan almacenados para su estudio posterior fuera de línea. La generación de estos informes se lleva a cabo de manera incremental —se añade la información que aporta el análisis de un documento en cuanto concluye—, de forma que se minimice el riesgo de pérdida de información ya disponible si ocurriera una interrupción abrupta del proceso que no supusiera la destrucción del soporte de la información.

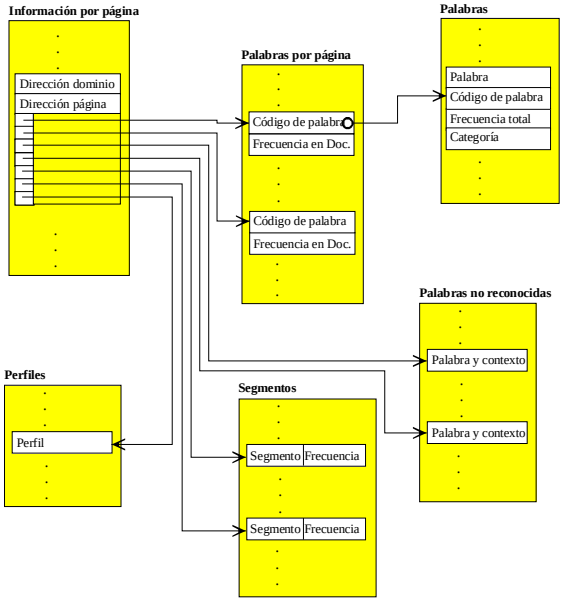

Figura 8: estructura de los resultados de DAWeb

Tal como se muestra en la figura 8, los resultados se organizan según una jerarquía dimanante del binomio dominio_página —por dominio se entiende la dirección de partida de un proceso—. El listado de *Información por página* está compuesto por un registro de información por página analizada, y consta de seis campos: 1) la dirección del dominio desde el que se ha accedido a la página —es necesario debido a que las páginas se recuperan y analizan en un orden desconocido, de manera que la información de dominios diferentes aparecería entremezclada—, 2) La dirección de la página a la que se refiere la información del registro, 3) una pareja de enlaces que identifica la región que corresponde a la página en el listado de *Palabras por página*, 4) otra pareja de enlaces que identifica la región que corresponde a la página en el listado de *Palabras no reconocidas*, 5) otra pareja de enlaces más que identifica la región que corresponde a la página en el listado de *Segmentos* y 6) un enlace que relaciona la página con el listado de *Perfiles*.

El listado de *Palabras por página* está formado por un conjunto de pares de valores numéricos —un *Código de palabra* y su frecuencia de aparición en la página. El *Código de palabra* sirve para localizar la información de la palabra que representa en el listado general de *Palabras*; tal listado contiene la siguiente información acerca de todas las palabras encontradas y reconocidas: la *Palabra*, su *Código* de identificación, su *Frecuencia de aparición total* y su *Categoría* gramatical. Las palabras no reconocidas se incluyen en el listado de *Palabras no reconocidas*, tantas veces como aparezcan y acompañadas en cada ocasión por un extracto del contexto en que aparecen —una región de unos 80 caracteres en torno a la palabra. El listado de *Segmentos* contiene los segmentos que cumplan con los requisitos de longitud y frecuencia mínima que se establecieron en la configuración junto con su frecuencia de aparición. El listado de *Perfiles* contiene para cada página un *Perfil* de la forma en que se incorporan las palabras al texto: se divide el texto en partes y se calcula para cada parte el número de palabras que aparecen por primera vez frente al número total de palabras que se usan —dato que permite hacer comparaciones, incluso gráficas, entre textos, particularmente si tienen una extensión parecida.

Toda la información generada queda almacenada en un conjunto de ficheros agrupados bajo una carpeta cuya denominación se forma mediante la unión del nombre dado por el usuario en el momento de configuración del proyecto con la fecha y hora en la que éste fue ejecutado —los resultados obtenidos en instantes diferentes quedan así perfectamente identificados y separados.

3 Interfaz de DAWeb

La interfaz de DAWeb, figura 9, consta de un menú y un área de trabajo dividida en tres secciones: 1) *Configuración*, 2) *Proceso* y 3) *Programación*. El menú presenta sólo dos opciones: 1) *Proyecto* permite crear un proyecto nuevo, abrir un proyecto previamente almacenado, guardar un proyecto recién creado, guardar un proyecto con un nombre distinto del que tenía y salir de la aplicación; 2) *Acerca de* visualiza información acerca de DAWeb. DAWeb trabaja con el concepto de proyecto definible como la especificación de un conjunto de opciones para realizar el análisis de un determinado grupo de documentos web de una cierta manera; tal especificación recibe un nombre —nombre del proyecto— y se puede almacenar para utilizarla en cualquier momento —incluye la posibilidad de activarse automáticamente en tiempos preprogramados.

La sección de configuración es donde el usuario de DAWeb define las características del proyecto en cuanto a descarga y análisis que plantea. De arriba abajo, el primer elemento que se encuentra sirve para atribuir un título a un proyecto de nueva creación o visualizar —y cambiar si se desea— el título de un proyecto —todo proyecto debe tener un título.

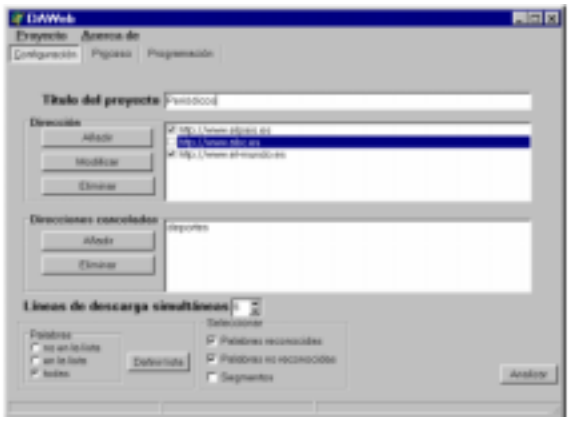

Figura 9: interfaz de DAWeb

Por debajo del título se muestra una zona destinada a configurar las direcciones cuyo contenido se quiere analizar; consta de cuatro elementos: la lista de direcciones introducida por el usuario que permite seleccionar las que se quieren utilizar en cada momento —no es necesario navegar siempre por todas las direcciones incluidas en un proyecto— y tres botones para añadir, modificar o eliminar una dirección de la lista. Las opciones *Añadir* y *Modificar* permiten, como muestra la figura 10, indicar o corregir una dirección y seleccionar un fichero para almacenar los datos recogidos —se pueden elegir ficheros diferentes para direcciones diferentes. La dirección especificada no designa únicamente una página web, sino que sirve como punto de partida para analizar el conjunto de páginas del mismo dominio accesibles por los hiperenlaces que contenga —con las restricciones que se impongan en la zona de direcciones canceladas. Tanto el botón *Modificar* como el *Eliminar* requieren que haya una dirección seleccionada en la lista de direcciones.

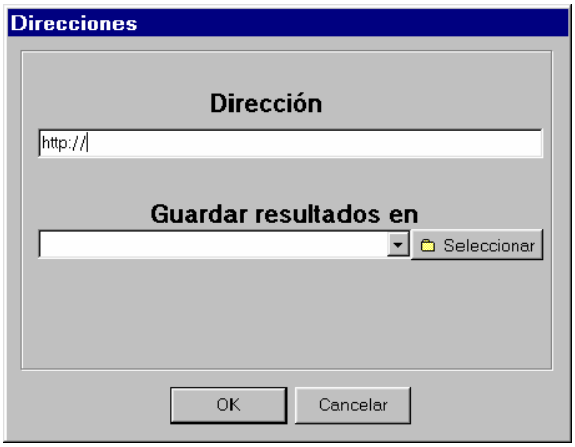

Figura 10: diálogo de añadir y modificar

La zona de direcciones canceladas se utiliza para designar una lista de direcciones que no se quieren analizar, aunque formen parte del dominio de alguna de las direcciones de partida activas. Se proporciona un botón llamado *Añadir* —lanza un cuadro de diálogo con tal propósito, figura 11 — y otro *Eliminar* que descarta la entrada seleccionada en la lista de direcciones canceladas. Se pueden indicar tanto direcciones completas como cadenas de caracteres que puedan formar parte de una dirección; por ejemplo, puede que no se desee analizar la sección de deportes de un periódico,

la cual está distribuida en varias páginas con direcciones diferentes pero todas contienen la palabra "deportes": en lugar de especificar todas las direcciones diferentes de las páginas de deportes, se puede indicar "deportes" —no se accede a ninguna dirección que contenga esta secuencia de caracteres.

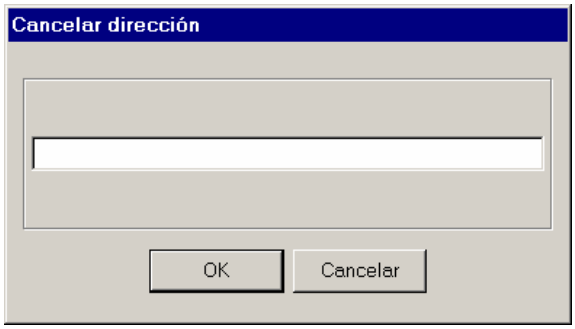

Figura 11: diálogo para añadir a la zona de direcciones canceladas

En la opción *Líneas de descarga simultáneas* se configura el número de hilos de ejecución que se activarán con módulos recuperadores en paralelo; inicialmente está puesta a 5 y puede bajarse hasta 1 —produce una descarga lineal, aunque lenta— o subirse hasta 10 —el máximo razonable según los experimentos realizados.

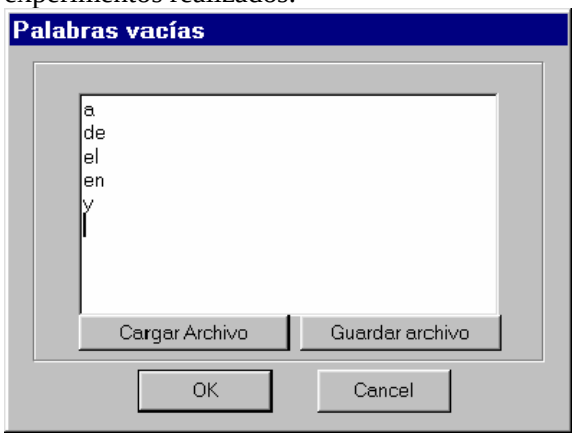

Figura 12: diálogo de definir lista

La parte inferior de la sección de configuración se dedica a definir cómo se llevará a cabo el análisis de los documentos accedidos según las direcciones fijadas en las secciones anteriores. Se puede definir una lista de "palabras vacías" o una lista de "palabras significativas"; en el primer caso —***no*** *en la lista*— las palabras no entran en el análisis y en el segundo —***en*** *la lista*— el análisis se restringe a esas palabras. Ambas listas se configuran en un cuadro de diálogo —se lanza en el botón *Definir lista*—, figura 12, que permite editarlas, guardarlas para su uso en otros proyectos y recuperarlas de un fichero.

Se puede configurar la descarga de información que se efectúa: las palabras reconocidas, las no reconocidas —suficiente si sólo se están buscando neologismos— y las secuencias frecuentes de palabras (*Segmentos*) —útil en estudios más generales.

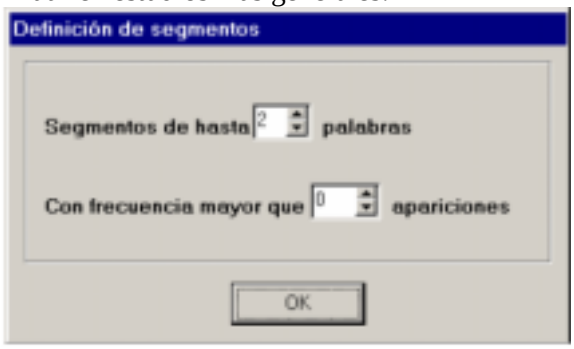

Figura 13: diálogo definición de segmentos

En la esquina inferior derecha de la pestaña de configuración se encuentra el botón *Analizar* que pone en marcha el análisis configurado y cambia la presentación de la aplicación —activa la sección de proceso mientras dure.

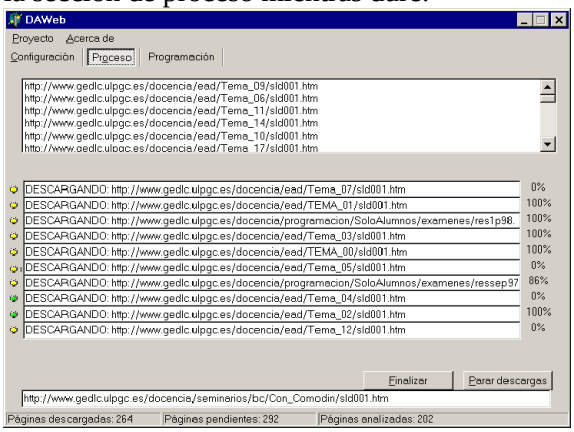

Figura 14: sección de proceso durante la ejecución de un proyecto

La sección de proceso se activa cuando se inicia la ejecución de un proyecto —puede accederse siempre, pero no da información si no se está ejecutando un proyecto. La primera área que muestra contiene la *lista de direcciones pendientes*; se modifica a medida que se toman direcciones para intentar su acceso y que se incluyen las que se encuentran en los

documentos accedidos —dentro de los límites establecidos en la configuración del proyecto. Debajo del área de direcciones aparecen tantas líneas de información como módulos recuperadores se activen; cada línea muestra el estado del recuperador —preparado, descargando, redireccionado, terminado, falló la dirección, falló la recuperación, no existe el *servidor* y se necesita autentificación—, la dirección a la que está accediendo o intentando acceder y el porcentaje de la respuesta que se ha obtenido. Una línea de información situada más abajo informa acerca del documento que está en curso de análisis —de entre los que ya se han recuperado. En la parte inferior, la línea de estado de la aplicación muestra información general acerca de cuántas direcciones se han accedido, cuántas están pendientes de ser accedidas y cuántas han sido analizadas —el primer y el tercer número son siempre crecientes, mientras que el segundo fluctúa en función de las direcciones que aporte cada documento accedido y del ritmo de trabajo de los recuperadores. Los dos botones de acción que ofrece esta sección tiene por objeto detener la ejecución del proyecto en marcha si se considera necesario; esta parada no es en ningún caso inmediata, ya que se han de abortar las conexiones en curso, terminar los análisis iniciados y vaciar las listas de direcciones y documentos pendientes. La diferencia entre *Parar descargas* y *Finalizar* es que el primero analiza las páginas que ya estén descargadas antes de terminar y el segundo no lo hace.

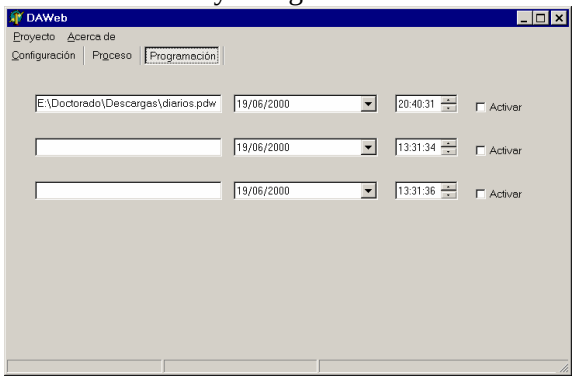

Figura 15: sección de programación

La sección de programación, figura 15, tiene por objeto la selección de una lista de proyectos a los que se les asigna una hora para su ejecución diferida. No hay problema por asignar la misma hora a varios proyectos, pero estos no se ejecutan en paralelo. Se ejecuta el primer proyecto que alcance o sobrepase su hora de ejecución, y el resto queda pendiente hasta que éste haya terminado; si mientras tanto se alcanza la hora de ejecución de algún otro, aquel se iniciará en cuanto el que está en ejecución acabe.

4 Conclusiones

Este trabajo se inscribe en el interés de la informática hacia todo lo relacionado con el lenguaje. Tal inquietud ha producido y continúa produciendo técnicas y herramientas de ayuda importantes en diversas facetas del trabajo del lingüista. Así mismo, el campo de la informática relacionado con el desarrollo de técnicas de procesamiento del lenguaje natural se ha beneficiado y sigue beneficiándose de la atracción que estas herramientas ejercen en muchos filólogos y de la consiguiente mejora en el conocimiento de las lenguas que semejantes actividades implican.

La disponibilidad de herramientas adecuadas constituye el camino necesario para obtener el debido rendimiento de la metarred como fuente lingüística de caudal nunca antes imaginado. Existe el precedente de la propia expansión de Internet: aunque los elementos físicos estaban disponibles, no empezó a convertirse en un fenómeno de masas hasta que se diseñó el World Wide Web y aparecieron los navegadores adecuados para que los usuarios sin una cualificación informática o tecnológica específica pudieran acceder con facilidad a la información. Ahora ya no se trata de elaborar herramientas básicas, sino unas de carácter más complejo y especializado que permitan descubrir espacios de utilidad en el aprovechamiento de la información que abrieron las primeras aproximaciones. Precisamente, las herramientas relacionadas con el lenguaje auguran la mayor proyección de todas las posibles, ya que trabajan en la base de comunicación de cualquier clase de información.

En aras de conseguir mejores resultados con arquitecturas más flexibles, DAWeb tenderá a diversificarse en un conjunto de agentes independientes —programas especializados—, pero con capacidad de cooperación mutua que se activarían conjuntamente sobre el flujo de datos entrante en función de las tareas que se van a realizar —unos se ocuparán del análisis, otros de la búsqueda de coocurrencias, otros de la búsqueda de neologismos, etc.—, de forma

que cada uno se encargaría de una tarea diferente y simple —aunque puedan haber varios trabajando en un mismo tipo de tarea. Por ejemplo, en el caso de sólo buscar neologismos se dedicarían 10 agentes a esa labor, mientras que si interesaran además coocurrencias de categorías gramaticales se podrían dedicar 6 y 4 o 3 y 7 agentes o cualquier configuración que aumentase la eficiencia por dedicar más recursos a la tarea que más los necesite —la configuración podría alterarse dinámicamente a medida que evolucionara la situación.

Bibliografía

Tomas H. Cormen; charles E. Leiserson; Ronald L. Rivest. Introduction to Algorithms. The MIT Press, 1990.

Antonio Zampolli. *Hacia bases* multifuncionales *de datos léxicos*. Las industrias de la lengua. Ed.: Fundación Germán Sánchez Ruipérez, 1 991. 185/202.

Ballester Monzón, A.; Díaz Roca, M.; Santana Pérez, F.; Santana, O. *Recuperación de Información en Diccionarios*. Boletín de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN). Febrero, 1 993.Nº 13, 423/430.

Santana, O.; Rodríguez del Pino, J. C.; González Domínguez, J. D. *Frectext: Una Aplicación de Ayuda a la Elaboración de Documentos*. Boletín de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN). Febrero, 1 993. Nº 13, 451/462.

Santana, O.; Hernández, Z. J.; Rodríguez, G. *Conjugaciones Verbales*. Boletín de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN). Febrero, 1 993. Nº 13, 443/450.

Díaz, M.; Pérez, J.; Santana, O. *Distancia Dependiente de la Subsecuencia Común Más Larga entre Cadenas de Caracteres*. Anales de las II Jornadas de Ingeniería de Sistemas Informáticos y de Computación, Quito (Ecuador). Abril, 1 993. 117/123.

Rodríguez, G.; Hernández, Z.; Santana, O. *Agrupaciones de* Tiempos *Verbales en un Texto*. Anales de las II Jornadas de Ingeniería de Sistemas Informáticos y de Computación, Quito (Ecuador). Abril, 1 993. 132/137.

Santana, O.; Díaz, M.; Rodríguez, J. C.; González, D.; Rodríguez, G.; Hernández, Z.; Ballester, A. *Información Textual: Línea de Investigación y Proyectos de Desarrollo*. Español Actual. Ed.: Arco/Libros, S. L. Nº 59/1 993. 31/37.

Santana, O.; Hernández, Z.; Rodríguez, G.; Pérez, J.; Carreras, F.; Bogliani, S. *Reconocedor de conjugación en formas verbales que trata los pronombres enclíticos*. Lingüística Española Actual. Ed.: Arco/Libros, S. L. 1 994, Nº 16_1. 125/133.

Ronald J. Vetter; Chris Spell; Charles Ward. *Mosaic an the World-Wide Web*. Computer, Vol. 27 Nº 10, octubre 1 994. 49/57.

Santana, O.; Hernández, Z.; Rodríguez, G.; Rodríguez, J. C.; González, J. D. *Proyecto SOTA: Sistema de Organización de Texto Abierto*. Procesamiento de Lenguaje Natural, Revista nº 16. Ed.: SEPLN. Abril, 1 995. Nº 16, 92/94.

Santana, O.; Pérez, J.; Santos, S.; Rodríguez, G.; Hernández, Z. *Proyecto GEISA: GEstión Integrada de Sinónimos y Antónimos*. Procesamiento de Lenguaje Natural, Revista nº 16. Ed.: SEPLN. Abril, 1 995. Nº 16, 79/81.

José Ramón Alameda; Fernando Cuetos. *Diccionario de frecuencias de las unidades lingüísticas del castellano*. Servicio de publicaciones de la Universidad de Oviedo. 1 995.

Santana, O.; Hernández, Z.; Pérez, J.; Rodríguez, G.; Carreras, F. *Diccionarios en soportes informáticos*.. Cuadernos Cervantes de la Lengua Española, nº 11 Noviembre _ Diciembre, 1 996 68/77.

Santana, O.; Rodríguez, G.; Hernández, Z. *Herramienta para el manejo de diccionarios ideológicos*. Lingüística Española Actual XIX, 1, 1 997. Ed. Arco/Libros, S.L. 127/136.

Santana, O.; Pérez, J.; Carreras, F.; Santos, S.; Rodríguez, G.; Hernández, Z. *GEISA: Un* diccionario *de sinónimos en formato electrónico*. Revista de Lexicografía, Volumen III. Universidade da Coruña. 1 996_1 997. 111/134.

Santana, O.; Pérez, J.; Hernández, Z.; Carreras, F.; Rodríguez, G. *FLAVER: Flexionador y lematizador automático de formas verbales*. Lingüística Española Actual XIX, 2, 1997. Ed. Arco/Libros, S.L. 229/282.

Dan Gusfield. *Algorithms on strings, trees, and sequences. Computing Science and Computacional Biology*. Cambridge University Press, 1997.

Alvar Ezquerra, M. *La redacción lexicográfica* asistida *por ordenador: dificultades y deseos*. Diccionarios e informática, 1998. Publicaciones de la Universidad de Jaén. 3/22.

Concepción Maldonado González.. *Problemas reales en la elaboración de un diccionario: historia de los diccionarios SM*. Diccionarios e informática, 1998. Publicaciones de la Universidad de Jaén. 43/55.

Santana, O.; Pérez, J.; Carreras, F.; Duque, J.D.; Hernández, Z.; Rodríguez, G. *Reconocedor y generador automático de formas nominales*. Diccionarios e informática, 1998. Publicaciones de la Universidad de Jaén. 57/74.

Santana, O.; Pérez, J.; Carreras, F.; Hernández, Z.; Rodríguez, G.; Duque, J.D. *De un reconocedor y generador morfológico del español en Internet*. Publicado Mayo, 1999, Lexicon Planet Ltd.

Santana, O.; Pérez, J.; Carreras, F.; Duque, J.; Hernández, Z.; Rodríguez, G. *FLANOM: Flexionador y lematizador automático de formas nominales*. Lingüística Española Actual XXI, 2, 1999. Ed. Arco/Libros, S.L. 253/297.

J. Abbate. *Inventing the Web*. Proceedings of the IEEE, Vol. 87 Nº 11, noviembre 1999. 1999/2002.

Horacio Rodríguez Hontoria. *Técnicas estadísticas en el tratamiento del lenguaje natural*. Filología e Informática. Seminario de filología e informática de la Universidad Autónoma de Barcelona. 1999. 111/140.

José Antonio Millán. Estaciones *filológicas*. Filología e Informática. Seminario de filología e informática de la Universidad Autónoma de Barcelona. 1999. 143/164.

María Morrás. *Informática y crítica textual: realidades y* deseos. Filología e Informática. Seminario de filología e informática de la Universidad Autónoma de Barcelona. 1999. 143/164.

Santana, O.; Pérez, J.; Losada, L. Generación automática de respuestas en análisis morfológico. Estudios de lingüística. Universidad de Alicante, 14, 2000. Departamento de Filología Española, Lingüística General y Teoría de la Literatura. 245/257.