UN MÓDULO DE DESAMBIGUACIÓN MORFOSINTÁCTICA PARA EL CASTELLANO BASADO EN CONOCIMIENTO LINGÜÍSTICO

LOURDES AGUILAR

ANA-BELÉN AVILÉS

JORDI FONTSECA

CARME DE LA MOTA

YOLANDA RODRÍGUEZ

Departament de Filologia Espanyola (UAB)

PAOLA CAYMES

Departament d'Informàtica (UAB)

SERGIO BALARI

Departament de Filologia Catalana (UAB)

Dirección de Contacto: Sergio Balari Ravera, Departament de filologia Catalana, Facultat de Lletres, Edifici B, Universitat Autònoma de Barcelona, E-08193 Bellaterra (Barcelona). Sergi.Balari@uab.es

UN MÓDULO DE DESAMBIGUACIÓN MORFOSINTÁCTICA PARA EL CASTELLANO BASADO EN CONOCIMIENTO LINGÜÍSTICO

Resumen

En el presente artículo describimos una herramienta informática de desambiguación morfológica para el español, diseñada para ser integrada en un sistema de corrección gramatical avanzada para el castellano y el catalán basado en la combinación de dos tipos de herramientas, a saber: a) Un analizador morfosintáctico de bajo nivel y b) Un analizador sintáctico/semántico de alto nivel. Actualmente, el proyecto se halla en la fase de desarrollo de las herramientas de bajo nivel que, a medio plazo, deberían ser la base de un módulo de corrección gramatical capaz de capturar errores simples en texto irrestricto. El archivo de reglas contiene 743 reglas sobre ambigüedad morfológica, que en el corpus de desarrollo se aplican un total de 130.751 veces. El corpus de desarrollo recién etiquetado presenta un grado de ambigüedad del 64.78%, incluyendo aquí cualquier tipo de ambigüedad, tanto de categorías mayores, como de categorías menores. Después del proceso, el grado de ambigüedad se reduce a un 13.86%.

Palabras clave: Procesamiento del lenguaje natural, Desambiguación morfológica, Gramáticas de estados finitos.

Abstract

In this paper we describe a computational tool for morphological disambiguation for the Spanish language, designed to be eventually part of a larger grammar checking system for both Spanish and Catalan. This system is based on two different types of tools, namely, a) A low-level morphological parser, and b) A high-level syntactic-semantic parser. At present, we are developing all low-level tools, which, in the mid term, should constitute the basis for a grammar-checking module capable of capturing simple errors in unrestricted text. The rule file contains some 743 rules for morphological ambiguity, which are applied 130.751 times over our development corpus. This corpus, once performed the morphological tagging, has a degree of ambiguity of 64.78%, including both major and minor category ambiguities. After going through the disambiguation process, remaining ambiguities in the corpus amount to just a 13.86%.

Keywords: Natural language processing, Morphological disambiguation, Finite-State Grammars.

1 INTRODUCCIÓN

En el presente artículo describimos una herramienta informática de desambiguación morfológica para el español desarrollada en el marco del proyecto Preparación Automatizada de Documentos (PrADo). Dicho proyecto conjunto entre la Universitat Autònoma de Barcelona (UAB) y la Universitat Pompeu Fabra (UPF), tiene como objetivo final el

desarrollo de un sistema de corrección gramatical avanzada para el castellano y el catalán basado en la combinación de dos tipos de herramientas, a saber: a) Un analizador morfosintáctico de bajo nivel y b) Un analizador sintáctico/semántico de alto nivel. Actualmente, el proyecto se halla en la fase de desarrollo de las herramientas de bajo nivel que, a medio plazo, deberían ser la base de un módulo de corrección gramatical capaz de capturar errores simples en texto irrestricto.

Como se describe en el apartado siguiente, el sistema posee una arquitectura modular, organizada en diferentes susbsistemas que van operando sobre el texto de entrada hasta producir como salida un texto etiquetado morfosintácticamente con el mayor grado de precisión posible, es decir, con el menor número de ambigüedades. El resultado de este proceso sería la entrada para el sistema de corrección de bajo nivel, del cual existe ya un prototipo experimental para el catalán desarrollado por la UPF.

2 ARQUITECTURA DEL SISTEMA

La arquitectura del sistema DeMoNiO, de base lingüística, cuenta con una estructura de módulos interrelacionados para el tratamiento de texto libre en español: (1) preprocesado de texto, (2) etiquetador morfológico, (3) desambiguador morfológico, (4) etiquetador sintáctico, (5) desambiguador sintáctico; módulos que aportarán la necesaria y suficiente información lingüística para diagnosticar errores en textos escritos y proponer alternativas de sustitución.

2.2 Módulo de preproceso

El preprocesador está implementado en lenguaje *Java*, dado que, de cara a los requerimientos previstos para el futuro, es necesario que el módulo sea portable a diferentes sistemas operativos. Además es necesaria la utilización de un lenguaje compatible con el conjunto de caracteres Unicode. En combinación con Java, se utilizó una herramienta generadora de analizadores escritos en Java, denominada *JLex*. Esta herramienta (Berk 2003) permite generar un analizador léxico a partir de una especificación formal de los componentes léxicos del lenguaje en cuestión. Dicha especificación está compuesta por un conjunto de expresiones regulares que definen las entidades del lenguaje. Precisamente la utilización de una especificación formal compuesta de expresiones regulares para definir el lenguaje facilita la descripción conceptual del mismo y es lo que permite obtener un producto más robusto, correcto y completo, ya que se gana en tiempo, esfuerzo descriptivo y exhaustividad en la definición de entidades léxicas.

El módulo de preprocesado identifica fechas, cifras, nombres propios y abreviaturas, que, según las aplicaciones del sistema, podrán ser expandidas o no, y que incorporarán información flexiva. En el diseño del preprocesador se tuvieron en cuenta todos los aspectos relacionados con eficiencia, robustez, modularidad y facilidad de manejo de datos.

El preprocesador está compuesto por un motor de procesamiento, el analizador léxico, que conjuntamente con el listado de siglas, el de abreviaturas, el lexicón y el diccionario

permite procesar el corpus de entrada, a fin de darle el formato necesario (etiquetado especial para siglas, abreviaturas, nombres propios, etc.) para las siguientes fases del procesamiento.

La *Clase Diccionario* define un tipo abstracto que ofrece prestaciones similares a las de un diccionario de uso corriente, más algunas adicionales que permitan hacerlo más flexible; por ejemplo, *buscar(p)* permite buscar la palabra *p* en el diccionario. Más interesantes son las opciones *insertar(p)*, *eliminar(p)* y *corregir(p,q)* que permiten, respectivamente, agregar, quitar y cambiar palabras en el diccionario, de tal forma que éste puede estar en constante actualización, revisión y adaptación para las distintas aplicaciones.

El diccionario está implementado sobre un archivo persistente que contiene el listado de palabras del lenguaje en combinación con una tabla de Hash que se utiliza durante la ejecución del programa para que las operaciones se lleven a cabo más eficientemente que si se ejecutaran directamente sobre el archivo. Este archivo es lo que designamos con el término *lexicón*, y puede modificarse en diferentes ejecuciones del programa.

El lexicón es el listado de las palabras del lenguaje, el cual incluye el listado de las siglas y de las abreviaturas conocidas.

La *Clase Preprocesador* es la que maneja y lleva a cabo el procesamiento propiamente dicho del corpus de entrada, de acuerdo con el siguiente algoritmo:

1. Inicialización de listado de abreviaturas

2. Inicialización de listado de siglas

3. Creación y apertura del diccionario

4. Procesamiento de la entrada

4.1. Obtención de la siguiente palabra

4.2. Clasificación de la palabra

4.3. Verticalización de la palabra

5. Cierre del diccionario

2.3 Etiquetado morfológico

La proyección morfológica se realiza sin tener en cuenta el contexto, a partir de la información que ofrece el sistema *maco+* (Márquez 2001). No obstante, debido a que el proyecto PrADo se plantea como objetivos la corrección gramatical en castellano y en catalán, es preciso traducir las etiquetas morfológicas a las definidas por el Grupo de Lingüística Computacional de la UPF, participante del mismo proyecto.

Veamos en (1) un ejemplo a partir de la oración *Yo bajo con el hombre bajo a tocar el bajo bajo la escalera*, que presenta un grado de ambigüedad del 57.14%. El formato de salida de *maco+* consiste en la verticalización del texto y, junto a cada una de las palabras, los lemas posibles y la etiqueta de cada uno de los lemas; las etiquetas se corresponden con el estándar europeo EAGLES adaptado para el español.

(1)

Yo yo PP1CSN00

bajo bajar VMIP1S0 bajo AQ0MS00 bajo NCMS000 bajo SPS00

con con SPS00

el el TDMS0

hombre hombre NCMS000

bajo bajar VMIP1S0 bajo AQ0MS00 bajo NCMS000 bajo SPS00

a a NCFS000 a SPS00

tocar tocar VMN0000

el el TDMS0

bajo bajar VMIP1S0 bajo AQ0MS00 bajo NCMS000 bajo SPS00

bajo bajar VMIP1S0 bajo AQ0MS00 bajo NCMS000 bajo SPS00

la la NCMS000 la TDFS0 él PP3FSA00

escalera escalera NCFS000

. . Fp

En esta oración, el principal problema morfológico es el lema *bajo*, con cuatro posibles lecturas, a saber, verbo, adjetivo, nombre y preposición. Además de categorías mayores, entendidas en el sentido tradicional del término, las ambigüedades también pueden ser, obviamente de categorías menores: género, número, modo verbal, tiempo, persona, etc.

El formato de *maco+* se transforma a uno identificable por el formalismo de la *Constraint Grammar*, según se muestra en (2):

(2)

"<s id="1">

"<Yo>"

"yo" Pron person fort 1pers masc-fem sg nom - - PP1CSN00

"yo" Pron person fort 1pers neut sg nom - - PP1CSN00

"<bajo>"

"bajar" Verb MInd Pres 1pers sg Prin - VMIP1S0

"bajo" Adj qual masc sg - - AQ0MS0

"bajo" Nom com masc sg - NCMS000

"bajo" Prep SPS00

"<con>"

"con" Prep SPS00

"<el>"

"el" Esp art masc sg DA3MS0

"<hombre>"

"hombre" Interj I

"hombre" Nom com masc sg - NCMS000

"<bajo>"

"bajar" Verb MInd Pres 1pers sg Prin - VMIP1S0

"bajo" Adj qual masc sg - - AQ0MS0

"bajo" Nom com masc sg - NCMS000

"bajo" Prep SPS00

"<a>"

"a" Nom com fem sg - NCFS000

"a" Prep SPS00

"<tocar>"

"tocar" Verb Inf Prin VMN0000

"<el>"

"el" Esp art masc sg DA3MS0

"<bajo>"

"bajar" Verb MInd Pres 1pers sg Prin - VMIP1S0

"bajo" Adj qual masc sg - - AQ0MS0

"bajo" Nom com masc sg - NCMS000

"bajo" Prep SPS00

"<bajo>"

"bajar" Verb MInd Pres 1pers sg Prin - VMIP1S0

"bajo" Adj qual masc sg - - AQ0MS0

"bajo" Nom com masc sg - NCMS000

"bajo" Prep SPS00

"<la>"

"el" Esp art fem sg DA3FS0

"la" Nom com masc sg - NCMS000

"él" Pron person febl 3pers fem sg acus - - PP3FSA00

"<escalera>"

"escalera" Nom com fem sg - NCFS000

"<$.>"

</s>

Además del cambio de formato, exigido por el formalismo de la Constraint Grammar, se han adoptado algunas decisiones que afectan al tipo de categorías morfológicas utilizadas. A título de ejemplo, las etiquetas *determinante* y *pronombre*, se han fusionado bajo una sola categoría, la de *especificador*, siempre que el mismo lema permita la pronominalización

(Quixal 2003). Esto explica que en el caso de *ambas***,** donde tendríamos una triple ambigüedad, sólo encontremos dos posibles lecturas:

(3)

"<ambas>"

"amba" Nom com fem pl - NCFP000

"ambos" Esp card fem pl DC3FP0

En el caso de *la***,** en cambio, se mantiene la distinción entre *pronombre* y *especificador*, ya que se trata de lemas distintos:

(4)

"<la>"

"el" Esp art fem sg DA3FS0

"la" Nom com masc sg - NCMS000

"él" Pron person febl 3pers fem sg acus - - PP3FSA00

2.4 Desambiguación morfológica

El núcleo del sistema lo forman las gramáticas regulares escritas en el formalismo de la *Constraint Grammar* (Tapanainen 1996), modelo gramatical basado en restricciones. La estrategia esencial de esta aproximación consiste en elaborar un análisis morfosintáctico

parcial a partir de la información contextual proporcionada en cada oración. Los motores de las gramáticas han sido cedidos para su uso en investigación para la UAB. Es tarea nuestra proponer las reglas que den cuenta del análisis.

Entre los principios que han guiado las decisiones en la construcción del módulo de desambiguación morfológica, cabe destacar la prioridad de la precisión y el ajuste a la realidad descriptiva sobre la eficacia o robustez. Previo a la formulación de las reglas, se lleva a cabo un estudio lingüístico y, antes de su implementación definitiva, el funcionamiento de las reglas se documenta y verifica sobre textos, con el fin de proponer solo reglas muy fiables.

Por tanto, se necesitaba un corpus que se correspondiera con el perfil de usuario del proyecto: bilingüe catalán-castellano (en ningún caso no peninsular) de nivel cultural y de redacción medio-alto.

El corpus, que actualmente consta de 238.766 palabras, está organizado jerárquicamente siguiendo una estructura temática. Así disponemos de textos tanto de lenguaje no especializado (correo electrónico, web, literatura y prensa) como especializado (derecho y lingüística, aunque está previsto ampliarlo a otras ramas del saber). Los textos que integran el corpus son actuales (en ningún caso anteriores a 01-01-2000)[1].

El analizador que usamos actualmente es una versión conocida como CG-2 de un motor anterior. Ha sido desarrollada por el Departamento de Lingüística General de la Universidad de Helsinki y la empresa Connexor e implementado en C. La primera lengua natural para la

que se desarrolló una *Constraint Grammar* fue el inglés, y su proceso de desarrollo tuvo lugar entre 1992 y 1995 en la propia Universidad de Helsinki[2].

La información aportada por el texto analizado morfológicamente con todas las lecturas posibles, se modifica a partir de reglas restrictivas basadas en información lingüística contextual cuya ventana máxima es la oración. El objetivo del proceso de desambiguación morfológica es, precisamente, eliminar o reducir al máximo estas ambigüedades a partir de la implementación de reglas basadas en dos operadores, SELECT y REMOVE, que seleccionan o eliminan, respectivamente, aquellas lecturas que no proceden en un contexto determinado.

El primer bloque de reglas consiste en una serie de reglas particulares que eliminan aquellas ambigüedades que son producto de un etiquetaje demasiado amplio y que no está en consonancia con nuestro perfil de usuario por tratarse de anacronismos, americanismos, etc.

(5) Ejemplos de reglas particulares:

LIST FORMAS_RARAS: “vale”, “vasos”, “bastante”... (V.Fin + Cl)

LIST LETRAS_AMBIGUAS: “a”, “y”...

AMERICANISMOS: “bancar”, “desdar”, “cochar” (banco, desde, coche) ...

El resto de reglas se diseñó a partir de los ejemplos reales del corpus. Los principales criterios y decisiones lingüísticas que se han seguido fueron tomados siguiendo el modelo de la CATCG, dotando así de unidad a ambos sistemas según estaba previsto por el proyecto PrADo.

En lo que respecta a las decisiones lingüísticas, por ejemplo, se han adoptado la ya comentada convergencia de las categorías *Determinante* y *Pronombre* en la de *Especificador* o la resolución de la ambigüedad *Verbo Participio* y *Adjetivo calificativo* en favor de la lectura verbal, dado que la frontera entre ambas categorías es difusa y difícil de sistematizar (Boleda 2003).

En cuanto a los principios que guían la construcción del módulo de reglas, se ha priorizado la corrección por encima de la eficacia. A pesar de que en algunos casos era posible aplicar reglas que aseguraran una gran eficacia y un margen relativamente pequeño de error, se ha optado siempre por reglas cuyos fundamentos lingüísticos minimizaran la posibilidad de cometer errores. Es el caso de la ambigüedad entre *Conjunción* y *Pronombre Relativo*, ambigüedad de la que aún podemos encontrar 2687 ejemplos en el corpus de desarrollo.

Cabe destacar que la gramática de restricciones no tiene en cuenta consideraciones de orden de aplicación. No obstante, dada la complejidad en la elaboración de un fichero de reglas de desambiguación morfológica, dichas reglas están ordenadas según categorías mayores y las ambigüedades que con ellas confluyen, de modo que pueden irse escribiendo y modificando en función de los distintos problemas que se detectan en los textos etiquetados[3].

Siguiendo con el ejemplo de (1), después de la desambiguación morfológica, que en este caso deja un porcentaje de ambigüedad del 0%, la salida del proceso es la que se muestra en (6):

(6)

"<s id="1">

"<yo>" S:831/2

"yo" Pron person fort 1pers masc-fem sg nom - - PP1CSN00

"<bajo>" S:926/2, 2894/2, 3182/2

"bajar" Verb MInd Pres 1pers sg Prin - VMIP1S0

"<con>"

"con" Prep SPS00

"<el>"

"el" Esp art masc sg DA3MS0

"<hombre>" S:433/2

"hombre" Nom com masc sg - NCMS000

"<bajo>" S:926/2, 1178/2, 3053/2

"bajo" Adj qual masc sg - - AQ0MS0

"<a>" S:454/2

"a" Prep SPS00

"<tocar>"

"tocar" Verb Inf Prin VMN0000

"<el>"

"el" Esp art masc sg DA3MS0

"<bajo>" S:1178/2, 974/2, 3209/2

"bajo" Nom com masc sg - NCMS000

”<bajo>" S:1106/2, 1178/2, 939/2

"bajo" Prep SPS00

"<la>" S:623/2, 497/2

"el" Esp art fem sg DA3FS0

"<escalera>"

"escalera" Nom com fem sg - NCFS000

</s>

El esquema básico de una regla es, pues, el siguiente:

(7)

REMOVE/SELECT (tag) IF (0 xxx) (-1 yyy) (1 zzz) ;

En el caso de *bajo*, por ejemplo, las reglas que se han aplicado son:

(8)

REMOVE (Prep) IF (0 PREP_NO_PREP) (1C PREP) ;

REMOVE (Prep) IF (0 PREP) (-1C DET_MASC) ;

REMOVE (Nom) IF (0 PREP_NOM) (1C ESP OR NOM OR PRON OR VINF) ;

REMOVE (Adj) IF (0 <"bajo"> OR <"Bajo">) (-1C ART) ;

REMOVE (Verb) IF (0 PREP_VERB) (*-1 VFIN BARRIER CONJ OR PRONREL OR PUNT OR COMA) ;

REMOVE (Nom) IF (0 V1P) (-1C PPERS_FUERTE + 1P) ;

REMOVE (Adj) IF (0 NOM) (-1C ESP) (NOT -1 LO) (1 PREP) ;

Vemos que todas las reglas se decantan por eliminar una de las lecturas posibles; esto ocurre porque siempre se considera preferible borrar una lectura que seleccionarla, ya que la

segunda es una operación mucho más arriesgada y el grado de error puede aumentar si se utiliza sin el rigor necesario. Se han delimitado algunos contextos que definen algunas combinaciones como imposibles y, por tanto, se puede borrar alguna de las lecturas, por ejemplo en la primera reglas se elimina la lectura prepositiva si la palabra siguiente es, con toda seguridad (de ahí la C “carefully”), una preposición: *Yo* bajo con *el hombre bajo*... Para cada una de las reglas el proceso es parecido, aunque se pueden añadir más especificaciones, como por ejemplo el operador BARRIER, que permite poner límites de aplicación dentro de la oración.

3 DESARROLLOS FUTUROS

Actualmente, el desarrollo del módulo de desambiguación se considera terminado. El archivo de reglas contiene 743 reglas sobre ambigüedad morfológica, que en el corpus de desarrollo se aplican un total de 130.751 veces. El corpus de desarrollo recién etiquetado presenta un grado de ambigüedad del 64.78%, incluyendo aquí cualquier tipo de ambigüedad, tanto de categorías mayores, como en categorías menores. Después del proceso, el grado de ambigüedad se reduce a un 13.86%.

Se han desarrollado también unas herramientas, implementadas en Perl, que permiten hacer consultas en un corpus etiquetado sobre ambigüedad morfológica. Se pueden buscar palabras que contengan entre sus lecturas hasta tres categorías distintas. Otra herramienta

permite la búsqueda de concordancias morfológicas en el corpus ya desambiguado mediante la *Constraint Grammar*, con una opción avanzada que incluso permite especificar uno de los lemas de la búsqueda y recupera la fuente, es decir, la oración completa[4].

En este momento se está llevando a cabo el proceso de evaluación del módulo de desambiguación morfológica teniendo en cuenta únicamente categorías mayores. El proceso consiste en la alineación de un mismo texto desambiguado respectivamente por DeMoNiO y por una herramienta de tipo estadístico. Mediante un proceso automático se comparan ambos archivos y se considera que, en caso de coincidencia, ambas formas dan el análisis correcto. En caso de observarse una divergencia o de que DeMoNiO omita alguna ambigüedad, se considera que se ha producido un error. Aquí entra en juego el papel del evaluador humano, que decide qué herramienta ha efectuado el análisis correcto y qué lectura debería haber sido la elegida en el caso de la ambigüedad renuante.

Antes de dar por definitivamente cerrado el desarrollo del módulo de desambiguación morfológica se deberían tener en cuenta algunas consideraciones. La principal es el proceso de revisión y ampliación de algunas de las secciones de reglas, sobre todo las que se refieren a aquellas categorías antes mencionadas que aún presentan un grado relativamente alto de ambigüedad persistente. También debería adaptarse y explotar la información léxico-semántica que se ha utilizado para el desarrollo de la CG del catalán; quizás no parezca muy necesaria en lo que a desambiguación morfológica se refiere (aunque es obvio que ayudaría, sobre todo en los casos de régimen y subcategorización) pero será básica de cara al desarrollo de los módulos de etiquetado y desambiguación sintáctica, con los que ya se está empezando

a trabajar. Cabría proponer, además, un nuevo proceso de evaluación que incluya la revisión de todos los rasgos morfológicos, no sólo la categoría y que, a poder ser, tome para lo comparación un texto etiquetado manualmente.

Como ejemplo del funcionamiento del proceso de evaluación, en la oración *Es cierto* que *actualmente se están dando estas diferencias de opinión*, DeMoNiO se decanta, en el caso de *que* por la lectura de *conj subord* mientras el sistema estadístico se decanta por el pronombre relativo. Ocurre algo parecido en la oración *Las mujeres, por ejemplo, que sufren entre dos y tres veces más depresiones* que *los hombres*; el sistema estadístico selecciona la lectura de *pronombre relativo* porque la palabra va precedida de un nombre, DeMoNiO, en cambio, selecciona la *conjunción subordinante* porque hay reglas que hacen referencia a las estructuras comparativas. Otro ejemplo de divergencias lo encontramos en la oración *El* norteamericano medio *entre 25 y 30 años ha engordado 5 kilos*; el sistema estadístico presupone que es mucho más común *medio* como nombre que *medio* como adjetivo, sin embargo en este ejemplo no ocurre así y, mientras DeMoNiO permite que permanezca la ambigüedad, el sistema estadístico yerra.

Un último ejemplo: en el caso de *mínimos* en la oración *Además sus efectos secundarios son* mínimos debería permanecer, como en el análisis de DeMoNiO, la ambigüedad entre *nombre* y *adjetivo*, ya que sin información léxico semántica o, por decirlo de otra forma, únicamente con información morfológica, no es posible distinguir si aquí se está hablando de el “límite inferior, o extremo a que se reduce algo” (nombre) o de algo “tan pequeño en su especie, que no lo hay menor ni igual” (adjetivo).

NOTAS

1. En una fase posterior, al hacer el paso de analizador a corrector gramatical y, sobre todo, de estilo, el corpus deberá verse ampliado, siguiendo la estructura jerárquica actual, hasta abarcar también traducciones (básicamente del inglés). Con esta ampliación se podría ya hacer una tipología de errores que permitiera desarrollar convenientemente las herramientas más avanzadas.

2. En el marco del proyecto PrADo, además de la del castellano, se ha desarrollado una CG para el catalán, de la que se puede encontrar una demo en línea y que ha sido reutilizada en otros proyectos del Grup de Lingüística Computacional (GLiCom) de la UPF como BancTrad, ALLES o eTitle.

3. La tarea de creación de las reglas se repartió entre la UPF y la UAB, de modo que las categorías 'Nombre', 'Adjetivo', 'Especificador' y 'Pronombre' se trataron en la UPF, mientras que en la UAB se trabajó con las categorías 'Conjunción', 'Preposición', 'Verbo' y 'Adverbio'.

4. Las herramientas se encuentran disponibles para realizar consultas libres en la página del proyecto (http://prado.uab.es) y son las siguientes:

- Concordancias. Permite buscar cadenas de texto como un programa básico de concordancias. Como curiosidad, en el corpus de desarrollo aparece más veces la palabra *hobbit* (5) que la palabra *mariposa* (2).
- Ambigüedad morfológica. Ejemplo: Nombre-preposición-adverbio; resultado: aparte
- Concordancias morfológicas. Muy útil para buscar colocaciones, régimen, etc. Ejemplo: "viaje" seguido por una preposición: 10 casos, 6 *viaje a*, 3 *viaje de* y 1 *viaje por*.

BIBLIOGRAFÍA

Berk, E. 2003. “A lexical analyzer generator for Java”. Princeton University: Department of Computer Science. [Documento de Internet disponible en http://www.cs.princeton.edu/~appel/modern/java/JLex].

Boleda, G. 2003. “Adquisició de classes adjectivals”. Universitat Pompeu Fabra: Treball de Recerca. Doctorat de Ciència Cognitiva i Llenguatge. Barcelona.

Márquez, L. 2001. “Identificació i classificació de noms propis utilitzant ‘Margin-based Learning Algorithms’”. Universitat Politècnica de Catalunya: Grupo de Procesado del Lenguaje Natural del Dpto. de Lenguajes y Sistemas Informáticos. Barcelona. [Documento de Internet disponible en http://www.lsi.upc.es/~webia/doctia/lista/19105702142001.html].

Quixal, M. 2003. “Theoretical basis and implementation of a linguistic-based morphosyntactic tagger for Catalan”. Universitat Pompeu Fabra: Treball de Recerca. Doctorat de Ciència Cognitiva i Llenguatge. Barcelona.

Tapanainen, P. 1996. *The Constraint Grammar Parser CG-2*. Helsinki: Publications of the University of Helsinki, Nº 27.

AGRADECIMIENTOS

La investigación que aquí se presenta no habría sido posible sin la financiación concedida por el Ministerio de Ciencia y Tecnología para el proyecto PrADo con referencia TIC2000-1681-C02-02. Nuestro más sincero agradecimiento a nuestros colegas y coparticipantes en el proyecto del Grup de Lingüística Computacional de la UPF cuya colaboración y experiencia han sido fundamentales para llevar a buen puerto este trabajo. Gracias también al Ministerio de Asuntos Exteriores y a su Programa de Cooperación Interuniversitaria AL.E. 2003 que permitió que Paola Caymes colaborara en el proyecto como becaria.