

Un Analizador Morfológico para el castellano basado en Chart*

Angel Luis González Collar (algc@mat.upm.es)
José Miguel Goñi Menoyo (jmg@mat.upm.es)
José Carlos González Cristóbal (jcg@gsi.dit.upm.es)

*E.T.S.I. Telecomunicación, Universidad Politécnica de Madrid
Ciudad Universitaria, 28040 Madrid
Phone: +34 1 336.72.87
Fax: +34 1 336.72.89*

Palabras clave: Lenguaje natural, análisis morfológico, *chart*.

Resumen

Este artículo describe un sistema capaz de analizar morfológicamente palabras en castellano. Se describe un modelo computacional para el tratamiento de la morfología flexiva del castellano basado en la concatenación de morfemas y en la unificación de rasgos. La implementación del analizador emplea la técnica de análisis sintáctico que se apoya en la estructura de datos denominada *chart*, que permite registrar eficientemente los resultados finales e intermedios del análisis y evitar búsquedas redundantes.

Abstract

In this article we describe a system capable of doing morphological analysis of spanish words. This system uses a computational model of spanish inflectional morphology based on morpheme concatenation and feature unification. The analyser is a chart parser that allows to keep partial and final results efficiently and avoids redundant search.

1 Introducción

La Morfología se considera la parte de la Gramática que trata dos áreas fundamentales: el estudio de las unidades existentes dentro de la palabra y sus relaciones entre sí; y la teoría de las categorías léxicas o caracterización de unidades como sustantivo, verbo, etc.

Hockett [Moreno, 92] distingue tres enfoques sobre la estructura de la palabra:

Palabras y Paradigma (*Word and Paradigm*): En este modelo las palabras se clasifican por categorías y por paradigmas (grupo de palabras que sirven de modelo o clase, con las mismas particularidades morfológicas), y se establecen clases paradigmáticas sobre lemas que hacen de modelo (como por ejemplo, *amar* para la primera conjugación, etc.). Los formantes de la palabra en este caso son las raíces o lexemas y las terminaciones o accidentes. Las implementaciones de procesadores de este modelo son

*Este trabajo ha sido financiado parcialmente por el Plan Nacional de I+D, bajo el proyecto titulado: *Una Arquitectura para Interfaces en Lenguaje Natural con Modelado de Usuario* (TIC91-0217C02-01).

bastante flexibles y computacionalmente eficientes. Ejemplos de sistemas implementados son el analizador morfosintáctico del Instituto de Lingüística Computacional de Pisa para el español [Ratti et. al., 83], el procesador morfológico de IBM para el español [Rodríguez et. al., 90], y MORFOGEN [Pentheroudakis, 91].

Elementos y Proceso (*Item and Process*): El modelo consta de elementos (los morfemas léxicos), sobre los cuales actúan un menor número de morfemas (flexivos, derivativos) que modifican a los primeros. A estos cambios que afectan a la raíz léxica dando lugar a la llamada forma superficial, se les denomina procesos morfológicos.

Los procesadores que siguen este enfoque suelen utilizar autómatas de estados finitos para implementar reglas. Suponen dos niveles de representación: un nivel de representación léxica (usada en el léxico del procesador), y un nivel de representación de cadenas superficiales (palabras ya flexionadas). Entre las cadenas de caracteres de cada uno de estos niveles se establece una correspondencia carácter a carácter que permite transformar una cadena en otra. La descripción de las lenguas que tratan se basa en reglas fonológicas y ortográficas. Como ejemplos citaremos el procesador morfológico para el español de [Tzoukermann and Liberman, 90] y el Procesador Morfológico de dos niveles PC-KIMMO [Antworth, 90].

Elementos y Colocación (*Item and Arrangement*): Este es el modelo utilizado por nuestro analizador morfológico. En él se considera que las palabras se pueden segmentar en unidades abstractas, que son los morfemas, relacionados entre sí por mera sucesión (un morfema tras otro). Según [Moreno, 92], en este modelo el análisis morfológico se reduciría a:

1. Una especificación del inventario de morfemas.
2. Una especificación de las secuencias en las que estos morfemas pueden aparecer.
3. La especificación del morfo o alomorfos (variantes de un morfema) que pueden realizar cada morfema¹.

La tarea fundamental de los procesadores que siguen este modelo es concatenar correctamente cada alomorfo con el correspondiente que le sigue o precede. Ejemplos destacados son GRAMPAL, una implementación en Prolog del modelo morfológico que adoptamos [Moreno, 92] y [Moreno and Goñi, 95], LA-MORPH [Hausser, 90], y el procesador objeto de este artículo.

Dos de los procesos más significativos en la formación de palabras son la flexión y la derivación. La flexión es el proceso en el que se unen morfemas flexivos a palabras o a raíces que no constituyen palabras por sí mismas. Distinguiremos dos tipos de flexión: verbal (o conjugación) si se realiza con verbos, y nominal (o declinación) si se hace con formas que admiten género y/o número, como nombres y adjetivos.

La derivación puede definirse como la formación de palabras a partir de lexemas adjuntando morfemas derivativos [Yllera et. al., 83]. Existen diversas causas por las que los procesos morfológicos derivativos son de una complejidad superior a los flexivos, por lo que el procesamiento computacional que aquí se describe no incluirá la derivación, considerando exclusivamente la flexión, si bien tanto el modelo computacional como el procesador morfológico podrían considerar la derivación si dispusiesen de un modelo lingüístico formalizado.

¹ Los alomorfos son las variantes de un morfema en determinados entornos. Por ejemplo, los alomorfos del morfema negativo *in-* son *i-*, que aparece en *irrespirable* o *ilegal* (en vez de *in-*, pues de lo contrario tendríamos **inrespirable* o **inlegal*), e *im-*, que aparece en *improbable* o *imbatido*.

2 Modelo computacional para el castellano

Para describir la morfología flexiva del castellano se ha utilizado el modelo computacional descrito en [Moreno, 92] y en [Moreno and Goñi, 95], que puede encuadrarse en el enfoque Elementos y Colocación. Este modelo supone un léxico de formantes y sus alomorfos, y una gramática basada en la unificación de rasgos para la concatenación de los morfemas,

2.1 El léxico

Como marco de representación léxica hemos diseñado e implementado una plataforma específica que se adapta al modelo computacional que hemos elegido. Esta plataforma se describe con detalle en [Goñi and González, 95] y [Goñi et. al., 95].

Cada una de sus entradas estará compuesta por una cadena superficial a la que se asocia una serie de rasgos, cada uno de los cuales tendrá asignado un valor. Por ejemplo, la entrada correspondiente al morfema *-ábamos* sería:

```
'abamos
agr pers = 1
agr num = plu
vinfo tense = impf
vinfo mood = ind
conj = 1
stt = 24
sut = reg
concat = vm
```

Los rasgos son los elementos que aportan toda la información léxica y/o gramatical asociada a la entrada correspondiente, como la categoría o la información necesaria para la concatenación con otros morfemas. Las entradas que forman el léxico pueden ser de tres tipos distintos:

Raíces: (verbales o nominales) y sus alomorfos. Elementos con insuficiente información gramatical, que necesitarán concatenarse con elementos flexivos.

Morfemas flexivos: y sus alomorfos. Elementos con información gramatical que necesitan concatenarse con elementos léxicos.

Formas lexicalizadas: Elementos léxicos con información gramatical suficiente que no se analizan por considerarse palabras, como sucede con formas verbales muy irregulares, formas de género inherente, formas invariables, etcétera.

Nuestro modelo considera que los alomorfos aparecen cuando un morfema puede tener diferentes realizaciones "gráficas", no meramente fonológicas, como ocurre en los tratados gramaticales tradicionales [Alarcos, 94].

2.1.1 La flexión verbal

Se considera que el verbo queda descompuesto únicamente en dos formantes: la raíz verbal y el morfema flexivo verbal (que incluye toda la información referida al tiempo, modo, aspecto, etc.). La codificación usada en el léxico permite ligar cada raíz con su morfema

flexivo correspondiente para cada forma de la conjugación de un verbo. Esta codificación sirve tanto para verbos regulares como irregulares. Según esta codificación, se asocia cada cadena superficial (representación o forma escrita de cada término léxico) con dos rasgos que indicarán con qué otras cadenas superficiales se puede combinar. Estos rasgos son:

stt (tipo de raíz): identifica la raíz verbal y la terminación correspondiente a cada forma verbal, mediante un código numérico.

sut (tipo de desinencia): distingue entre varios alomorfos del morfema flexivo verbal gracias a un conjunto de valores distintos.

Para referirse a cualquier forma verbal y poder indicar qué alomorfos de la raíz y de la terminación deben concatenarse se utiliza la matriz de las formas verbales (tabla 1). Cada forma verbal tiene una entrada, que se corresponde con los valores de cuatro rasgos: en vertical aparecen los valores de los rasgos "tiempo" y "modo", y en horizontal los de los rasgos "persona" y "número".

	sing_1	sing_2	sing_3	plu_1	plu_2	plu_3	—
pres_ind	11	12	13	14	15	16	
impf_ind	21	22	23	24	25	26	
indf_ind	31	32	33	34	35	36	
fut_ind	41	42	43	44	45	46	
pres_subj	51	52	53	54	55	56	
impf_subj	61	62	63	64	65	66	
cond	71	72	73	74	75	76	
imper		82	83		85	86	
inf							00
ger							90
part							99

Tabla 1: Matriz de conjugación verbal.

Los códigos numéricos de la matriz de conjugación son los valores del rasgo **stt**. El código de este rasgo debe incorporarlo tanto el alomorfo de la raíz de una forma verbal determinada, como la desinencia que le corresponda.

El código 100, que no aparece en la tabla 1, se reserva para las raíces únicas (las de los verbos regulares), para no tener que escribir todos los códigos de la conjugación completa. Los códigos 83 y 86 corresponden a las formas del llamado imperativo de cortesía, del singular y del plural respectivamente (como diga -usted-, digan -ustedes-).

La codificación que se utiliza para distinguir los distintos alomorfos de las desinencias se realiza representando los valores del rasgosut mediante cadenas alfanuméricas. Éstos valores son: **reg**, **pres**, **pret1**, **pret2**, **fut_cond**, **imp_subj**, **imper**, **infin**, **ger**, **part1** y **part2** (desinencias regulares, o variantes alomórficas para los morfemas de presente, pretérito o futuro de de indicativo, condicional, imperfecto del subjuntivo, imperativo, infinitivo, gerundio o participio, respectivamente).

Los rasgos **stt** y **sut** deben aparecer tanto en la raíz como en la terminación, ya que es necesario que ambas compartan el mismo valor para que puedan concatenarse, salvo el valor 100 para el rasgo **stt** de las raíces, que es compatible con cualquier otro.

2.1.2 La flexión nominal

Las formas nominales, a diferencia de las verbales, permiten segmentar sus formantes más claramente. Así, las formas nominales segmentables se podrán descomponer hasta en raíz, y morfemas de género y de número (en el caso de algunos plurales).

Para distinguir qué alomorfo del morfema de género de los tres existentes (dos para el masculino, -o y -e, y uno para el femenino, -a), debe concatenarse a una raíz nominal, se utiliza el rasgo *get* (tipo de género). Sus valores, (*mas1*, *mas2*, *fem* y *no*), se asignan a las raíces que admiten los morfemas de género citados. Al igual que el rasgo *sut* en el caso verbal, el rasgo *get* debe aparecer tanto en la raíz como en el morfema flexivo. Y para que se puedan concatenar dichos morfemas, los valores de este rasgo deben ser iguales en ambos.

Otro rasgo, llamado *nut*, será el utilizado para concatenar correctamente una palabra o raíz con uno de los dos alomorfos del morfema de plural (-s y -es). Deberán incorporarlo las palabras y raíces que se concatenen con alguno de dichos alomorfos (que también lo incluirán). Sin embargo, no lo llevarán las palabras que no admitan morfema de plural, como las invariantes en número (*virus*, *lunes*, etc.) y los *singularia y pluralia tantum*. Esto también se aplica a palabras (como *león* y *pez*) cuyos plurales no se forman a partir de las cadenas del singular debido al cambio gráfico: *leon-es*, *pec-es*. Por esta razón se deben incluir entradas adicionales en el léxico (los alomorfos de la raíz para formar los plurales).

En nuestro modelo se distinguen fundamentalmente dos tipos de rasgos según la información asociada a los mismos: los rasgos con información gramatical, que se utilizan en el proceso de combinación de información (que no es unificación estricta, sino pseudounificación), y rasgos con información contextual, a los que se les puede asignar una serie de valores en disyunción y que se utilizan en el proceso de concatenación morfológica. Ejemplos de los primeros son *agr gen* o *vinfo tense* y de los segundos *stt*, *sut*, *get* o *nut*.

2.2 La gramática

Para describir la morfología del castellano hemos adoptado un formalismo computacional basado en unificación de rasgos ampliamente conocido: PATR-II [Shieber, 86]. Las reglas que permiten la concatenación de morfemas son de tipo sintagmático (PSG) del tipo de Contexto Libre, aumentadas con un conjunto de ecuaciones entre las especificaciones de rasgos, que deben satisfacerse para que la aplicación de la regla tenga éxito.

% 1. Verbos regulares		% 2. Verbos irregulares	
w -> vl vm		w -> vl vm	
<x1 conj>	= <x2 conj>	<x1 conj>	= <x2 conj>
<x1 stt>	= 100	<x1 stt>	= <x2 stt>
<x1 sut>	= <x2 sut>	<x1 sut>	= <x2 sut>
<x0 cat>	= <x1 cat>	<x0 cat>	= <x1 cat>
<x0 agr>	= <x2 agr>	<x0 agr>	= <x2 agr>
<x0 vinfo>	= <x2 vinfo>	<x0 vinfo>	= <x2 vinfo>
<x0 lex>	= <x1 lex>	<x0 lex>	= <x1 lex>

Figura 1: Reglas de morfología verbal.

Las reglas para la flexión verbal son sólo las dos que se muestran en la figura 1. La primera de éstas admite las formas verbales regulares, y la segunda admite las formas ver-

bales irregulares. Las reglas, en general, estarán formadas por una expresión (por ejemplo: $w \rightarrow vl vm$) que indica que el constructo superior (la palabra) se compone de una serie de constructos inferiores (los formantes), seguida por una serie de ecuaciones. Los símbolos w (palabra), vl (lexema verbal), vm (morfema verbal), etc. son los valores del rasgo *concat* con el que se identifica cada formante del léxico.

Para las formas nominales existen cuatro reglas. La primera de éstas admite palabras en singular con morfema de género (*bonit-o*); la segunda admite plurales sin morfema de género, formados a partir de raíces y siempre usando el morfema de plural *-es* (*pec-es*); la tercera admite plurales formados a partir de palabras (*bonito-s*), y la cuarta regla valida palabras en singular. Las dos primeras se muestran en la figura 2.

<p>% 1. Raíz + morfema de género $wl \rightarrow nl ng$ <x1 get> = <x2 get> <x0 nut> = plu1 <x0 agr gen> = <x2 agr gen> <x0 agr num> = <x2 agr num> <x0 lex> = <x1 lex> <x0 ninfo> = <x1 ninfo> <x0 cat> = <x1 cat></p>	<p>% 2. Raíz + morfema de número $w \rightarrow nl nn$ <x1 nut> = <x2 nut> <x1 nut> = plu2 <x0 agr gen> = <x1 agr gen> <x0 agr num> = <x2 agr num> <x0 lex> = <x1 lex> <x0 ninfo> = <x1 ninfo> <x0 cat> = <x1 cat></p>
--	---

Figura 2: Reglas para morfología nominal.

3 Arquitectura del analizador

Nuestro sistema de análisis basado en *chart* se compone de dos módulos básicos: el sistema Núcleo [Casado, 93], y el analizador específico. Esta estructura aparece representada en la figura 3, junto un bloque de aplicación que hace uso del analizador.

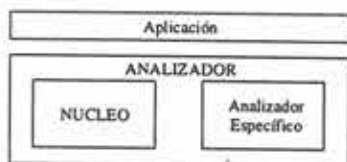


Figura 3: Arquitectura del analizador.

El sistema *Núcleo*, que está inspirado en la filosofía de diseño del sistema para la producción de analizadores sintácticos CMCHART [Thompson, 83], constituye el soporte sobre el que se construye el analizador. Proporciona funciones para mantener y manipular la estructura *chart* que crea el proceso de análisis, y para controlar dicho proceso. Este módulo es reutilizable, es decir, podrían construirse distintos analizadores gramaticales (sintácticos

o morfológicos) todos ellos con el mismo *Núcleo*, que se diferenciarían en su analizador específico. Cada analizador específico debería proporcionar las funciones que cada analizador particular necesitase. Estas funciones son utilizadas por *Núcleo*, tomándolas de una estructura denominada *tabla de funciones o señales*, en la que están referenciadas las funciones mencionadas.

3.1 Inicialización del analizador

Esta etapa comienza con la apertura de los ficheros de datos que requiere el analizador y con la inicialización de todas las estructuras de datos necesarias. Principalmente se crean las estructuras de datos internas que almacenarán el diccionario de datos, el léxico y la gramática. Cuando se carga la gramática, las ecuaciones de las reglas se ordenan según el algoritmo descrito en [Casado, 93] para que posteriormente se puedan verificar correctamente con dos pasadas en sentidos opuestos. Este algoritmo es la implementación que se ha realizado para la (pseudo)unificación de rasgos [Tomita, 88]. Si el analizador utilizase una unificación de rasgos real, los constructos compartirían rasgos en lugar de incorporar copias idénticas de ellos, lo que aseguraría la correcta transmisión de rasgos entre los constructos especificados en distintas identidades. En nuestro caso, el método citado ordena las ecuaciones de cada regla gramatical considerando el número de relaciones del haz o haces de rasgos de un constructo con los de otros constructos, establecidas mediante las ecuaciones, con el fin de evaluar en primer lugar las ecuaciones cuyos haces tengan el menor número posible de relaciones con otros haces en el conjunto de todas las ecuaciones de una regla gramatical. Con la ordenación realizada según este algoritmo, la transmisión de los rasgos quedará asegurada si además se evalúan las ecuaciones dos veces en sentidos opuestos.

En esta misma etapa se inicia también la tabla de señales de *Núcleo* que guiará su comportamiento. Esta estructura se crea mezclando las tablas de funciones del analizador específico y la que aporta el propio *Núcleo*, interconectando así este sistema con las funciones particulares del anterior. El sistema *Núcleo* dictará el flujo de acciones que deberán desarrollarse para realizar el análisis, pero cuando se necesite un funcionamiento particular según el procesamiento que se lleve a cabo, se recurrirá a las funciones del analizador específico. Posteriormente se iniciará la estructura *agenda*. Se trata de un almacén en el que se guardan por orden de prioridad las acciones que están pendientes de realizarse (que en nuestro caso son funciones encargadas de incorporar lazos al *chart*). *Núcleo* desarrolla su actividad siguiendo la secuencia que le va dictando *agenda*.

Por último, se iniciará una tabla en la que se mantendrán las estructuras *chart* que se vayan generando. El analizador específico puede adoptar esta posibilidad, proporcionada por *Núcleo*, o bien emplear su propia forma de almacenamiento de información.

3.2 Análisis gramatical de la cadena

En esta fase, se toma en primer lugar la palabra que se desea analizar y se procede a su segmentación, obteniéndose los formantes que integran la palabra. A continuación se crean los vértices del *chart* y los lazos léxicos de los formantes de la palabra, que pasan a *agenda*. Ello significa que por cada lazo léxico se introduce en *agenda* una función, que cuando sea invocada se encargará de añadir el lazo al *chart*.

Seguidamente se ejecuta el ciclo de extracción de lazos de *agenda*. En cada iteración de este ciclo se extrae un lazo, realizándose sobre él las siguientes operaciones:

1. Examen de redundancia del lazo extraído de *agenda*. Que un lazo sea redundante significa que ya hay en el *chart* un lazo con la misma información gramatical. Si el lazo es redundante no se añade al *chart* y se pasa al siguiente ciclo de extracción.
2. Si el lazo no es redundante entra en el *chart*. A continuación se intenta aplicar con el mismo la regla fundamental. De ser así se crean nuevos lazos pasarían a *agenda*.
3. Se aplica la invocación gramatical sobre el lazo añadido al *chart*. Por cada regla gramatical que sea una hipótesis admisible se creará un lazo, que pasará a *agenda*.

Los lazos que se originen por la aplicación de la regla fundamental y la búsqueda gramatical, al igual que los lazos léxicos, se introducirán en *agenda*. De esta manera se tiene un proceso en el que se estarán extrayendo continuamente lazos a los que se aplicarán las operaciones anteriores, y en el que a partir de un lazo se van generando otros. Cuando no queden lazos en *agenda* se habrá finalizado la etapa de análisis.

La prioridad con la que los lazos entran en *agenda* la determina el diseñador del analizador. Usualmente se da por más prioridad a los lazos fundamentales (generados por aplicación de la regla fundamental), que a los gramaticales (generados a partir de la invocación de reglas), y mayor prioridad a éstos que a los léxicos. De esta manera, en cada ciclo de *agenda* se extraerá el lazo de mayor prioridad, realizándose un recorrido descendente del espacio de búsqueda. Manipulando las prioridades de los lazos y la disciplina (como *FIFO* o *LIFO*) con la que entran los lazos en las distintas colas que posee *agenda*, se puede alterar la estrategia de recorrido del espacio de búsqueda.

Después de finalizar el ciclo de extracciones de lazos de *agenda* el *chart* estará construido. El sistema *Núcleo* devolverá un identificador del anterior a la aplicación. Si la palabra analizada tuviese varias segmentaciones posibles, se realizaría el análisis y la construcción del *chart* correspondiente para cada una de ellas.

4 Evaluación

El diccionario contiene actualmente unas 38.000 entradas (lemas), que cubren más de 450.000 formas flexivas. Se reparten aproximadamente como sigue: 20.000 nombres, 7.500 verbos, 10.000 adjetivos y 500 entradas para artículos, pronombres, preposiciones, conjunciones y adverbios. El número de morfemas flexivos incluidos supera los 500. Aparte de los fenómenos flexivos citados previamente, se tratan los pronombres clíticos en las formas verbales, la flexión de los participios verbales cuando funcionan como adjetivos y la formación de adverbios terminados en *-mente*.

Se ha realizado una evaluación de la cobertura de nuestros diccionarios y del analizador morfológico utilizando una colección de textos del suplemento cultural del diario ABC. Sobre dichos textos se realizó un filtrado previo para eliminar nombres propios, acrónimos, etc., resultando un corpus de unas 700.000 palabras, con 42.000 formas diferentes.

Se incorporó a los diccionarios de los tres sistemas empleados para la evaluación² un diccionario especializado de términos culturales (unos 400 lemas).

Los resultados de la evaluación se muestran en la tabla 2.

²Se emplearon el corrector ortográfico de WordPerfect 5.1 en Español, la versión 1.4 de los diccionarios de Español para ISPELL (desarrollados por J. Carretero y S. Rodríguez en la Facultad de Informática (UPM), y nuestro sistema ARIES.

Spell checker	Palabras distintas no reconocidas	% error (palabras distintas)	Palabras no reconocidas	% error (total palabras)
ARIES	3,167	7.5%	4,264	0.6%
ISPELL	5,630	13.3%	9,919	1.4%
WP 5.1	9,245	21.8%	18,622	2.6%

Tabla 2: Resultados de la evaluación

5 Conclusiones

El analizador morfológico desarrollado utiliza un modelo computacional basado en la concatenación de morfemas y en la unificación de rasgos, que permite tratar las formas flexivas del castellano. La sencillez y generalidad de las reglas morfológicas que incluye el modelo se debe en parte a que la complejidad está centrada en el léxico, puesto que concentra la información en los rasgos asociados a los formantes en vez de hacerlo en las reglas gramaticales. También destaca su declaratividad, en cuanto que realiza una descripción de las asociaciones permitidas entre los distintos alomorfos, mediante los rasgos utilizados específicamente para la concatenación. Esta característica permite que el modelo computacional sea apto tanto para análisis como para generación, como se demuestra con el prototipo GRAMPAL (Prolog) en [Moreno and Goñi, 95]. La notación simplificada de las reglas se debe también al empleo del formalismo computacional PATR-II.

En cuanto a la implementación, el sistema Núcleo tiene como principales características su flexibilidad, por servir como soporte o base común a distintos analizadores gramaticales, lo que simplifica el desarrollo de nuevos sistemas de análisis. La utilización de la estructura *chart* supone un intento de incrementar la eficiencia en tiempo del analizador, ya que esta estructura registra eficientemente los resultados parciales y finales del análisis, evitando la búsqueda redundante de los mismos. La implementación realizada en lenguaje ANSIC supone una mayor eficiencia con respecto a sistemas similares realizados con lenguajes declarativos (como PROLOG), además de facilitar su portabilidad a distintas máquinas y sistemas operativos.

Con respecto a otros procesadores encuadrados en los modelos de la tipología basada en los distintos enfoques de la estructura de la palabra, existen ciertas características diferenciadoras que permiten una mayor eficiencia computacional. Una de ellas es la incorporación de un inventario de alomorfos, que aunque incrementa el espacio consumido por el léxico al contar éste con un número mayor de entradas, supone un ahorro en tiempo de ejecución al no necesitar la generación de los mismos durante el procesamiento morfológico. El inconveniente de necesitar un mayor número de entradas en el diccionario se ha eliminado añadiendo las funcionalidades necesarias para posibilitar la generación de estos alomorfos automáticamente a partir de una forma base, mediante la aplicación de reglas alomórficas [Goñi and González, 95]. La segunda es el hecho de integrar la segmentación de la palabra a analizar con la búsqueda de los formantes en el léxico, por utilizarse una estructura *trie* como índice de acceso al léxico [Aoe and Morimoto, 92] y [Goñi and González, 95].

Actualmente estamos trabajando en la integración de este analizador morfológico con un analizador sintáctico también basado en el sistema Núcleo y en el desarrollo de un generador morfológico que esté basado en el mismo léxico y reglas.

Referencias

- [Alarcos, 94] Alarcos Llorach, E. Gramática de la Lengua Española. *Real Academia Española. Colección Nebrija y Bello. Espasa Calpe*, (1994).
- [Aoe and Morimoto, 92] Aoe, J. and Morimoto, K. An efficient implementation of trie structures. *Software-Practice and Experience*, vol. 22, n. 9, pp. 695-721, (1992).
- [Antworth, 90] Antworth, E. PC-KIMMO: A Two-level Processor for Morphological Analysis. *Academic Computing Department, Summer Institute of Linguistics, Dallas*, (1990).
- [Casado, 93] Casado, F. Realización de un núcleo para el desarrollo de sistemas de análisis sintáctico basados en chart. *Proyecto Fin de Carrera. ETSIT-UPM*, (1993).
- [Goñi and González, 95] Goñi, J.M. and González, J.C. A framework for lexical representation. *Proceedings of AI'95 (Language Engineering '95)*, pp. 243-252. Montpellier, (1995).
- [Goñi et. al., 95] Goñi, J.M.; González, J.C. and Moreno, A. A lexical platform for Spanish. *Proceedings of The Computational Lexicon (ESSLLI'95)*, pp. 61-65, (1995).
- [Hausser, 90] Hausser, R. Principles of Computational Morphology. *Laboratory for Computational Linguistics, Carnegie Mellon University*, (1990).
- [Moreno, 92] Moreno, A. Un modelo computacional basado en la unificación para el análisis y la generación de la morfología del español. *Tesis Doctoral. U.A.M.*, (1992).
- [Moreno and Goñi, 95] Moreno, A. and Goñi, J.M. GRAMPAL: A morphological model and processor for Spanish implemented in Prolog. *Proceedings of the 1995 Joint Conference on Declarative Programming (GULP-PRODE'95)*, pp. 321-331, (1995).
- [Pentheroudakis, 91] Pentheroudakis, J. E. MORFOGEN: A morphological analysis and dictionary interface tool. *Executive Communication Systems, Inc., Utah*, (1995).
- [Ratti et. al., 83] Ratti, D.; Saba, A.; Catarsi, M.N. and Cappelli, G. Analizador morfosintáctico de textos en lengua española. *Giardini, Pisa*, (1983).
- [Rodríguez et. al., 90] Rodríguez, A.; Sopena, L.; Valladares, C. y Villar, C. Clasificación morfológica del léxico castellano para un analizador en ordenador. *En Actas del VII Congreso Nacional de Lingüística Aplicada*, pp. 491-503, (1990).
- [Shieber, 86] Shieber, S.M. An Introduction to Unification-Based Approaches to Grammar. *CSLI Lecture Notes. Stanford University*, (1986).
- [Thompson, 83] Thompson, H.S. MCHART: A Flexible Modular Chart Parsing System. *In Proceedings of the National Conference on AI (AAAI'83)*, pp. 408-410, (1983).
- [Tomita, 88] Tomita, M. The Generalized LR Parser/Compiler. User's Guide. *Center for Machine Translation. Carnegie Mellon University*, (1988).
- [Tzoukermann and Liberman, 90] Tzoukermann, E. and Liberman, M.Y. A Finite-State Morphological Processor for Spanish. *In Proceedings of the 13th International Conference on Computational Linguistics (COLING'90)*, pp. 277-281, (1990).
- [Yllera et. al., 83] Yllera, A. et. al. Introducción a la Lingüística. *Ed. Alhambra U.*, (1983).