ANALISIS COMPUTACIONAL DE LA MORFOLOGIA DEL ESPAÑOL

M. Angel de Pineda Pérez
Mª de Carmen Piña Duarte
Pedro José Vázquez López

We intend to show in this paper an application made in the area of the Computational Linguistics: an automatic morphological processor of the spanish. We have explained the process of elaboration, since the building of the dictionaries and tables, which store lemmas and morphemes with their information, until the running of the computer formalism programmed in Turbo-Prolog, which makes the automatic analysis of the words.

Introducción

El área de investigación en el que se enmarca este trabajo ha recibido diversas denominaciones. Tal vez la más conocida sea la de *Lingüística Computacional,* traducción de la expresión inglesa «Computational Linguistics». La Lingüística Computacional [LC] es hoy una de las ramas más novedosas e interesantes de la investigación lingüística. Para encontrar el origen de esta disciplina hemos de remontarnos a los trabajos de traducción automática que realizaron A. D. Booth y W. Weaver en los años 1946-1949. Pero la LC ha experimentado una evidente evolución desde esos años hasta nuestros días; ha dejado de ser una disciplina auxiliar cuyos trabajos se limitaban a la elaboración de índices, concordancias, listas de palabras... etc., para convertirse en una ciencia autónoma que precisa de la creación de formalismos adecuados para llevar a cabo lo que se ha denominado *Procesamiento del Lenguaje Natural* [PLN].

La LC pretende explicar el comportamiento lingüístico mediante mecanismos simuladores de las actuaciones del hablante/oyente. Estos mecanismos se basan en ordenadores y algoritmos y pretenden reproducir los procesos onomasiológico y semasiológico de la comunicación lingüística. Algunas de la aplicaciones más desarrolladas dentro de esta

disciplina son la traducción automática, los sistemas de reconocimiento y síntesis de la voz humana y la elaboración de sistemas de acceso a banco de datos mediante el lenguaje natural. Cualquiera de estas tareas precisa de una herramienta fundamental que manipule la estructura sintáctica de los enunciados de forma automática; el término inglés con el que se denomina este análisis sintáctico automático es *parser* (derivado del latín «pars orationis»). Recogemos la definición del término dada por Kartunnen[1]:

«Parsing es una operación realizada por un ordenador con oraciones de una lengua natural que realiza agrupamientos parciales sucesivos de símbolos en unidades superiores e interpreta estos grupos como cambios en los estados de la máquina; esta operación se realiza mediante un algoritmo y/o con estrategias, heurística».

Las labores de investigación que venimos realizando en el Centro de Cálculo de la Facultad de Filología tienen como finalidad la construcción de un parser. La primera necesidad básica con la que tuvimos que enfrentarnos a la hora de llevar a cabo nuestros planteamientos fue la de disponer de un procesador que proporcionara la información morfológica de los elementos terminales implicados en el análisis. El objetivo de este trabajo es exponer cómo construimos nuestro analizador morfológico y cuál ha sido el resultado.

Análisis morfológico automático

El principal objetivo de cualquier procesador morfológico es el análisis y/o síntesis de las palabras de una lengua. En morfología computacional se entiende por palabra una cadena de caracteres que se encuentra entre dos espacios en blanco; a su vez, para la clasificación de los alomorfos se sigue un criterio estrictamente gráfico considerándose variante alomórfica toda cadena que presente alguna variación en sus caracteres.

En el procesamiento morfológico se suelen distinguir dos partes:

1. La lematización. Consiste en la segmentación de la palabra en morfemas, identificando las raíces y terminaciones en el diccionario.
2. Recuperación de la información léxica a partir de los morfemas segmentados.

Los procesadores morfológicos construidos hasta el momento se adscriben en cierto modo a alguno de los tres modelos morfológicos señalados por Hockett[2]:

1. Palabra y paradigma. Se basa en la analogía y el concepto de paradigma. En el diccionario se encuentran recogidas las distintas entradas a las que se les asigna un código que representa el paradigma al que pertenece. A través de reglas analógicas se obtienen las distintas formas de los distintos paradigmas. Pertenecen a este modelo el

---

[1] Dowty, D., Kartunnen, L., Zwicky, A.: *Natural Languaje Parsing*, Cambridge Univ. Press, 1985, p. 6.

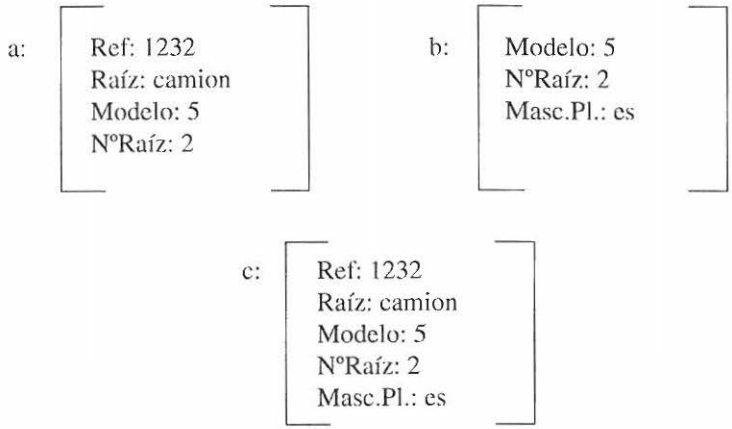
[2] Hockett, C.F.: *A Course in Modern Linguistics*. Toronto, Macmillan, 1958. Trad. esp.: *Curso de Lingüística Moderna*, Buenos Aires, Eudeba, 1971.

Analizador morfosintáctico de textos en español desarrollado en el Instituto de Lingüística Computacional de Pisa y el Procesador Morfológico de IBM para el español.

2. Elemento y proceso. Trata de explicar la relación existente entre palabras de la misma familia morfológica como resultantes de un proceso lingüístico. Parte de la gramática morfografémica de Kaplan y Kay y obtiene su mayor desarrollo en la morfología en dos niveles de Koskenniemi. Se basan en este modelo el sistema MARS de la empresa SIEMENS y el analizador del euskera de la Facultad de Informática de San Sebastián.
3. Elemento y colocación. Se basa en el morfema, representándose en el diccionario cada variante alomórfica por medio de una entrada. La labor fundamental consiste en asignar correctamente cada alomorfo al correspondiente que le sigue o precede. Se la suele conocer como gramática morfosintáctica de rasgos. El analizador morfológico del proyecto EUROTRA constituye una aproximación a este tipo de gramática.

Nuestro procesador sigue básicamente las premisas del primer modelo y se enmarca dentro de las llamadas *Gramáticas de Unificación*. Las Gramáticas de Unificación son formalismos que comparten dos importantes características: la descripción de los objetos lingüísticos mediante rasgos, y la unificación como técnica básica para construir objetos amplios a partir de otros más simples.

La unificación es una operación que combina dos estructuras de rasgos en una sola. Dos rasgos o atributos serán compatibles (unificarán) si tienen los mismos valores. Si el valor de un rasgo es una variable ésta unificará con cualquier otro valor. Veamos el siguiente ejemplo en el que «c» es el resultado de la unificación de «a» y «b»:



La unificación es una operación extraordinariamente potente a la hora de resolver muchos de los problemas que plantea el tratamiento del lenguaje. Especialmente apto para el tratamiento de la lengua natural resulta también PROLOG, lenguaje de programación cuyo principal mecanismo de resolución es precisamente la unificación de términos.

PROLOG[3] es un lenguaje de programación basado en la lógica de primer orden. Fue creado a finales de los años 60 por Alain Colmerauer en Marsella y tenía como principal fin poder llevar a cabo un tratamiento automático del lenguaje natural. Aunque no es nuestro objetivo explicar el funcionamiento de PROLOG creemos necesario, para una mejor comprensión de lo que sigue, señalar algunas de sus características fundamentales.

Un programa en Prolog es una secuencia de reglas o cláusulas de la forma siguiente:

P:- Q1, Q2, ...,Qn.

que se leen: «P es verdadero si Q1, Q2, ...,Qn son verdaderos».

P es la *cabeza de la regla*, y *Q1, Q2, ..., Qn* son el *cuerpo* de la misma. El signo :- equivale a *si*. Cuando una regla tiene la forma:

P.

ésta se denomina *hecho*, puesto que no hay ninguna condición para que sea verdad. Cada uno de los elementos (P,Q...) se denominan *predicados* y se componen de un *funtor* –que suele dar nombre al predicado– y de *argumentos* (opcionales) que van entre paréntesis y separados por comas. Por ejemplo:

determinante(la,femenino,singular)
adverbio(X)

En el primer caso los argumentos son *constantes* (cadenas de caracteres que comienzan por una letra minúscula o secuencias de letras y otros caracteres encerradas entre comillas); en el segundo hay un solo argumento y éste es una *variable* (cadena de caracteres que comienza por una letra mayúscula o por un signo de subrayado y que puede adquirir un valor determinado en cada momento).

El argumento de un predicado puede ser a su vez otro predicado:

o(sn(determinante,sustantivo),sv(verbo)).

Un programa Prolog, por lo tanto, está compuesto de una secuencia de predicados y de relaciones entre éstos según la sintaxis explicada.

Bases de datos léxicas y morfológicas

Normalmente un sistema para el procesamiento del lenguaje natural consta básicamente de dos partes: un diccionario y una gramática. Nos centraremos, en primer lugar, en el componente léxico del sistema.

La estructura de nuestro diccionario se puede dividir en tres conjuntos separados en función de sus características morfológicas:

---

[3] Para mayor información sobre este lenguaje véase: Clocksin, W., y Mellish, C.: *Programación en Prolog*, Barcelona, Ed. Gustavo Gili, 1987.

1. Entradas correspondientes a las categorías sustantivo y adjetivo.
2. Entradas correspondientes a la categoría verbo.
3. Entradas correspondientes al resto de las categorías morfosintácticas.

La base de datos léxica correspondiente al diccionario posee una estructura flexible que permite tanto la síntesis como el análisis morfológico. Dicha base es independiente del lenguaje de programación utilizado en la redacción de los algoritmos encargados de formalizar el proceso lingüístico, así como de la teoría morfológica utilizada.

Sustantivos y adjetivos

Hemos agrupado estas dos categorías por la similitud en sus estructuras morfológicas. Esta base de datos está constituida por 45.000 entradas o lemas, aproximadamente, y sus correspondientes raíces procedentes de diccionarios de lengua y de textos escritos y orales.

La descripción morfológica se realiza mediante una codificación numérica que representa el modelo de flexión de la unidad respecto al género y al número. Cada código de modelo se corresponde, a su vez, con otros códigos que explican la forma en que la entrada léxica flexiona respecto al género y al número. El resultado es una tabla que tiene la siguiente apariencia:

Modelo	*Género*	*Número*	*Ejemplo*
1	1 1	1 1	niño
2	1 2	1 2	domador
3	1 3	1 1	jefe
4	1 5	1 1	duque
5	1 6	1 1	actor

El primer código de la columna de género describe si existe (1) o no (2) flexión de género. El segundo corresponde a la forma de flexión (hasta 8 variantes: 1: o/a, 2: /a, 3: e/a, 4: variante léxica, 5: /sa, 6: or/riz, 7: -x/isa, 8: -x/ina), o al género de la entrada (1: masculino, 2: femenino, 3: masculino y femenino). Por lo tanto, el significado del segundo código depende del significado del primero. La misma lógica sigue la descripción del número: el primer dígito nos dice si existe (1) o no (2) flexión de número. El segundo remite a la forma de flexión (1: /s, 2: /es, 3: z/ces), o al número obligado de la entrada (1: singular, 2: plural, 3: singular y plural).

Así, por ejemplo, al modelo 1 corresponden los siguientes códigos:

Género:

1: la entrada posee flexión de género

1: la formación del género es del tipo -o/-a

Número:

1: la entrada posee flexión de número

1: la flexión de número es del tipo /-s

Hay modelos que poseen el mismo tipo de formación de género y número. La diferencia se debe en estos casos al número de raíces que posee la entrada léxica. Por ejemplo, «león» y «domador» pertenecen al mismo tipo de formación de género y número (12 12), pero «león» posee dos raíces (diferenciadas por el acento), mientras que «domador» tiene una sola.

La base de datos correspondiente a este conjunto posee dos relaciones: la primera contiene tres campos: código de referencia, lema y categoría morfosintáctica. La segunda, unida a la primera por el código de referencia, posee, además de este campo, los de raíz, número de raíz y modelo paradigmático.

Por ejemplo, la entrada «león» se representa en la primera relación como:

_Ref león _Categoría

y en la segunda relación como:

_Ref león 1 _Modelo
_Ref leon 2 _Modelo

Verbos

En la descripción de la morfología verbal se siguen los mismos principios metodológicos. La base de datos cuenta con 15.000 entradas procedentes de diccionarios y tratados sobre la morfología verbal, así como de textos escritos y orales.

Para la descripción morfológica del verbo hemos aceptado la clasificación tradicional que distingue:

1. *Verbos regulares*, que, en cualquier tiempo, persona o modo no alteran la raíz o las desinencias propias del modelo (conjugación) al que pertenecen.
2. *Verbos irregulares*, cuyas formas flexionadas presentan alteraciones en su desinencia, raíz o en ambas, respecto del modelo conjugacional al que pertenecen.

Las gramáticas suelen señalar otros subtipos de verbos irregulares:

a) *Verbos defectivos,* que presentan un cuadro flexivo incompleto, es decir, que no se emplean en todas las formas conjugadas.

b) *Verbos unipersonales,* que sólo se usan en infinitivo y en la tercera persona del singular de todos los tiempos.

Por nuestra parte, hemos establecido, mediante un código numérico, un único tipo de modelo flexivo. Los verbos defectivos y unipersonales no contituyen modelos específi-

1. Entradas correspondientes a las categorías sustantivo y adjetivo.
2. Entradas correspondientes a la categoría verbo.
3. Entradas correspondientes al resto de las categorías morfosintácticas.

La base de datos léxica correspondiente al diccionario posee una estructura flexible que permite tanto la síntesis como el análisis morfológico. Dicha base es independiente del lenguaje de programación utilizado en la redacción de los algoritmos encargados de formalizar el proceso lingüístico, así como de la teoría morfológica utilizada.

Sustantivos y adjetivos

Hemos agrupado estas dos categorías por la similitud en sus estructuras morfológicas. Esta base de datos está constituida por 45.000 entradas o lemas, aproximadamente, y sus correspondientes raíces procedentes de diccionarios de lengua y de textos escritos y orales.

La descripción morfológica se realiza mediante una codificación numérica que representa el modelo de flexión de la unidad respecto al género y al número. Cada código de modelo se corresponde, a su vez, con otros códigos que explican la forma en que la entrada léxica flexiona respecto al género y al número. El resultado es una tabla que tiene la siguiente apariencia:

Modelo	*Género*	*Número*	*Ejemplo*
1	1 1	1 1	niño
2	1 2	1 2	domador
3	1 3	1 1	jefe
4	1 5	1 1	duque
5	1 6	1 1	actor

El primer código de la columna de género describe si existe (1) o no (2) flexión de género. El segundo corresponde a la forma de flexión (hasta 8 variantes: 1: o/a, 2: /a, 3: e/a, 4: variante léxica, 5: /sa, 6: or/riz, 7: -x/isa, 8: -x/ina), o al género de la entrada (1: masculino, 2: femenino, 3: masculino y femenino). Por lo tanto, el significado del segundo código depende del significado del primero. La misma lógica sigue la descripción del número: el primer dígito nos dice si existe (1) o no (2) flexión de número. El segundo remite a la forma de flexión (1: /s, 2: /es, 3: z/ces), o al número obligado de la entrada (1: singular, 2: plural, 3: singular y plural).

Así, por ejemplo, al modelo 1 corresponden los siguientes códigos:

Género:

1: la entrada posee flexión de género

1: la formación del género es del tipo -o/-a

Número:

1: la entrada posee flexión de número

1: la flexión de número es del tipo /-s

Hay modelos que poseen el mismo tipo de formación de género y número. La diferencia se debe en estos casos al número de raíces que posee la entrada léxica. Por ejemplo, «león» y «domador» pertenecen al mismo tipo de formación de género y número (12 12), pero «león» posee dos raíces (diferenciadas por el acento), mientras que «domador» tiene una sola.

La base de datos correspondiente a este conjunto posee dos relaciones: la primera contiene tres campos: código de referencia, lema y categoría morfosintáctica. La segunda, unida a la primera por el código de referencia, posee, además de este campo, los de raíz, número de raíz y modelo paradigmático.

Por ejemplo, la entrada «león» se representa en la primera relación como:

_Ref león _Categoría

y en la segunda relación como:

_Ref león 1 _Modelo
_Ref leon 2 _Modelo

Verbos

En la descripción de la morfología verbal se siguen los mismos principios metodológicos. La base de datos cuenta con 15.000 entradas procedentes de diccionarios y tratados sobre la morfología verbal, así como de textos escritos y orales.

Para la descripción morfológica del verbo hemos aceptado la clasificación tradicional que distingue:

1. *Verbos regulares*, que, en cualquier tiempo, persona o modo no alteran la raíz o las desinencias propias del modelo (conjugación) al que pertenecen.
2. *Verbos irregulares*, cuyas formas flexionadas presentan alteraciones en su desinencia, raíz o en ambas, respecto del modelo conjugacional al que pertenecen.

Las gramáticas suelen señalar otros subtipos de verbos irregulares:

a) *Verbos defectivos*, que presentan un cuadro flexivo incompleto, es decir, que no se emplean en todas las formas conjugadas.

b) *Verbos unipersonales*, que sólo se usan en infinitivo y en la tercera persona del singular de todos los tiempos.

Por nuestra parte, hemos establecido, mediante un código numérico, un único tipo de modelo flexivo. Los verbos defectivos y unipersonales no contituyen modelos específi-

cos, y quedan descritos mediante un código alfabético (**-D** para los defectivos y una **-U** para los unipersonales). De esta forma nuestra descripción comprende 63 modelos paradigmáticos.

Tomamos como base la catalogación propuesta por Ramón y Fernando García-Pelayo y Gross y Micheline Durand en el libro de la Editorial Larousse Conjugación. Consiste en una clasificación de 90 modelos distribuidos del modo siguiente:

modelos 1 y 2, verbos auxiliares (haber, ser).

modelos 3, 4 y 5, verbos regulares (1ª, 2ª y 3ª conjugación).

modelos 6 al 70, verbos irregulares

modelos 71 al 90, verbos con modificaciones ortográficas o prosódicas.

La mayor novedad de esta taxonomía se encuentra en estos últimos modelos correspondientes a verbos que, como señala la Real Academia, por tener leves mutaciones de tipo ortográfico no dejan de ser regulares. No obstante hemos optado por encuadrarlos en modelos diferentes a los de las tres conjugaciones regulares en función de una mayor homogeneidad en el tratamiento computacional.

Nuestra clasificación restringe el número de modelos por motivos de eficiencia computacional, ya que en cada grupo se encuandran aquellos verbos que tienen el mismo número de raíces y cada una de éstas toma las mismas desinencias.

La base de datos relacional que contiene la información referente a la morfología verbal posee una estructura semejante a la descrita anteriormente para sustantivos y adjetivos. La primera relación posee tres campos: referencia, lema (infinitivo) y categoría morfosintáctica (en esta relación siempre es verbo). La segunda relación, enlazada por el campo referencia con la anterior, incluye el campo de la raíz, número de la raíz, modelo, tipo (defectivo o unipersonal), y código del conjunto de desinencias que se unen a la raíz en cuestión para dar lugar a las distintas formas flexivas del verbo.

Así, podremos tener, en la primera relación:

_Ref acertar_Categoría

Y en la segunda relación:

_Ref acert _1ª_raíz _Modelo _Tipo _Desinencias
_Ref aciert _2ª_raíz _Modelo _Tipo _Desinencias

Proceso de analisis

Una primera etapa del análisis trata de localizar la forma de entrada en la base de datos de formas no flexivas o flexivas gramaticales, cuyos registros tienen la forma:

d_nflex(«a»,»6")
d_nflex(«ante»,»6")
d_nflex(«bajo»,»6")

```
d_nflex(«con»,»6")
d_nflex(«contra»,»6")
d_nflex(«de»,»6")
d_nflex(«desde»,»6")
d_nflex(«en»,»6")
d_nflex(«entre»,»6")
d_nflex(«hacia»,»6")
d_nflex(«hasta»,»6")
.....
d_flex(«yo»,»0",»1",»yo»,»2",»1",»1")
d_flex(«nosotros»,»1",»2",»yo»,»2",»1",»1")
d_flex(«nosotras»,»2",»2",»yo»,»2",»1",»1")
d_flex(«me»,»0",»1",»yo»,»2",»1",»1")
d_flex(«nos»,»0",»2",»yo»,»2",»1",»1")
d_flex(«mí»,»0",»1",»yo»,»2",»1",»1")
d_flex(«tú»,»0",»1",»tú»,»2",»1",»1")
d_flex(«vosotros»,»1",»2",»tú»,»2",»1",»1")
d_flex(«vosotras»,»2",»2",»tú»,»2",»1",»1")
d_flex(«te»,»0",»1",»tú»,»2",»1",»1")
```

Los distintos argumentos de estos predicados describen la categoría y subcategoría morfológicas de la forma, así como su adscripción a un lema determinado.

Si la forma no es localizada –o, en ciertos casos de posible homografía, incluso si lo es– en esta base de datos se inicia el proceso de análisis. Este se basa en la comprobación de hipótesis sobre la composición en raíz y desinencia de la palabra a analizar. Tal comprobación se realiza según el método de emparejamiento de formas («pattern matching»), mediante la unificación de los distintos atributos que constituyen el sistema morfológico.

Estos atributos están codificados mediante tablas de correspondencias.

Tablas para la morfología nominal

La primera de éstas establece las relaciones entre el modelo paradigmático, el número de la raíz y el conjunto de desinencias aceptadas por la raíz.

La segunda codifica cada una de las desinencias según un código numérico en correspondencia con la desinencia concreta y su descripción morfológica.

Estas tablas tienen la forma de predicados de prolog. Así el predicado «mod_s» corresponde a la primera tabla nominal, y el predicado «d_s», a la segunda.

```
mod_s(1,1,5)
.....
mod_s(10,1,14)
```

```
mod_s(10,1,15)
mod_s(11,1,2)
mod_s(12,1,3)
mod_s(13,1,4)
mod_s(14,1,1)
mod_s(14,1,3)
mod_s(15,1,2)
mod_s(15,1,4)
mod_s(16,1,1)
mod_s(16,1,17)
mod_s(17,1,1)
mod_s(17,1,16)
mod_s(18,1,2)
mod_s(18,1,25)
.....
mod_s(41,2,24)
```

El primer predicado puede leerse de la siguiente forma: «para el modelo flexional 01 y la raíz 1 el código de desinencias es el 5».

```
d_s(1,»»,masculino,singular)
d_s(2,»»,femenino,singular)
d_s(3,»»,masculino,plural)
d_s(4,»»,femenino,plural)
d_s(5,o,masculino,singular)
d_s(6,a,femenino,singular)
d_s(7,e,masculino,singular)
d_s(8,sa,femenino,singular)
d_s(9,or,masculino,singular)
d_s(10,riz,femenino,singular)
d_s(11,esa,femenino,singular)
d_s(12,isa,femenino,singular)
d_s(13,ina,femenino,singular)
d_s(14,os,masculino,plural)
d_s(15,as,femenino,plural)
d_s(16,es,masculino,plural)
d_s(17,s,masculino,plural)
d_s(18,sas,femenino,plural)
d_s(19,ores,masculino,plural)
d_s(20,rices,femenino,plural)
d_s(21,esas,femenino,plural)
d_s(22,isas,femenino,plural)
d_s(23,inas,femenino,plural)
d_s(24,es,femenino,plural)
d_s(25,s,femenino,plural)
```

En esta segunda tabla puede interpretarse el primer predicado como «la desinencia codificada como 1 es un elemento vacío («») cuya descripción es masculino y singular».

Proceso de análisis de la morfología nominal

Dada una forma concreta, por ejemplo «campo», el proceso comienza disponiendo un predicado como el siguiente:

f(campo, Ref, L, C, R, N, M, Des, Gen, Num)

Los argumentos tienen el siguiente significado: Ref: referencia, L: lema, C: categoría, R: raíz, N: número de raíz, M: modelo, Des: desinencia, Gen: género, Num: número.

La primera hipótesis de descripción supone que la forma completa coincide con una raíz y que, por tanto la desinencia es un conjunto vacío («»); con lo cual las variables Raíz y Desinencia del predicado anterior quedan instanciadas como sigue:

f(campo, Ref, L, C, campo, N, M, «», Gen, Num)

Para confirmar la hipótesis se procede a localizar la raíz «campo» en la base de datos correspondiente, donde se encuentra un registro como el siguiente:

Ref:	Raíz:	Nº raíz:	Modelo:
920	campo	1	16

En este momento la variable número de raíz queda instanciada con «1» y la variable modelo con el valor «16».

f(campo, 920, L, C, campo, 1,16,»», Gen, Num)

A continuación se busca un predicado «mod_s» cuyo primer argumento (modelo) unifique con «16», y cuyo segundo argumento (número de raíz) unifique con «1». Existen dos predicados que cumplen esta condición:

mod_s(16,1,1)
mod_s(16,1,17)

Para comprobar cuál de ellos corrobora la hipótesis de descripción se procede a comprobar que el segundo argumento del predicado «mod_s» (número de raíz) unifique con el mismo argumento del predicado «f». Esta unificación tiene éxito para los dos casos posibles. Por tanto es necesario comprobar el tercer argumento del predicado «mod_s» (código de desinencia). Para ello se busca un predicado «d_s» cuyo argumento desinencia sea «». Este es:

d_s(1,»»,masculino,singular)

La alternativa para el predicado «mod_s» (mod_s(16,1,17)), quedará rechazada puesto que en el predicado d_s(17,»s»,masculino,plural), el argumento desinencia es «s», que no unifica con el mismo argumento en el predicado «f».

Por tanto, los argumentos del predicado «f» quedan instanciados así:

f(campo, 920, L, C, campo, 1, 16, «», masculino, singular)

Por último, mediante el código de referencia se puede comprobar en la base de datos de lemas que el correspondiente a la refencia «920» es «campo», cuya categoría es «sustantivo».

f(campo,920,campo,sustantivo,campo,1,16,»»,masculino,singular)

Si se tratara de analizar la forma «campos», la primera hipótesis de descripción fallaría (no existe en la base de datos una raíz «campos»). Y se recurre a un proceso de segmentación, por la derecha, de una posible desinencia. La segunda hipótesis de descripción, sería pues:

f(campos, Ref, L, campo, N, M, s, Gen, Num)

Las instanciaciones subsiguientes:

f(campo, 920, L, campo, 1, 16, s, Gen, Num)
mod_s(16,1,1)
mod_s(16,1,17)
d_s(1,masculino,singular)
d_s(17,s,masculino,plural)

dan como resultado la posible descripción

f(campos,920, campo, sustantivo, campo,1,16,s,masculino,plural)

Tablas para la morfología verbal

El conjunto de tablas para la morfología verbal es más numeroso y complejo que el anteriormente descrito para el análisis de sustantivos y adjetivos.

En primer lugar existe una pequeña base de datos que recoge todas las formas y descripciones de los verbos especiales (ser, estar, ir).

especial(1090,era,23121)
especial(1090,era,21121)
especial(1090,erais,22221)
especial(1090,eran,23221)
especial(1090,eras,22121)
especial(1090,eres,22111)
especial(1090,es,23111)
especial(1035,fue,33131)
especial(1090,fue,23131)
especial(1090,fuera,23122)
especial(1035,fuera,33122)
especial(1035,fuera,31122)
especial(1090,fuera,21122)
especial(1035,fuerais,32222)
especial(1090,fuerais,22222)
especial(1090,fueran,23222)

especial(1035,fueran,33222)
.....

En este fragmento pueden comprobarse las descripciones alternativas de una misma forma (por ejemplo «fue» homógrafo para los verbos «ir» y «ser») que queda reflejada en el primer argumento (referencia del infinitivo: 1035: «ir», 1090: «ser»). El último argumento recoge la descripción morfémica de la forma, según se explica en el siguiente apartado.

Para los restantes verbos el conjunto de tablas se organiza de la siguiente forma:

1) Desinencias. La primera tabla hace corresponder a cada desinencia verbal un código numérico y una descripción morfológica. Debido a la homografía es posible tener para una misma secuencia de letras más de un código y más de una descripción. El predicado que contiene esta información es «d_v». Una pequeña muestra es la siguiente:

d_v(76,aríamos,11251,7)
d_v(82,eríamos,21251,7)
d_v(88,iríamos,31251,7)
d_v(118,iéramos,21222,7)
d_v(124,iéramos,31222,7)
d_v(136,iésemos,21222,7)
d_v(142,iésemos,31222,7)
d_v(154,iéremos,21242,7)
d_v(160,iéremos,31242,7)
d_v(251,iéramos,11222,7)
d_v(269,iésemos,11222,7)
d_v(287,iéremos,11242,7)
d_v(22,ábamos,11221,6)
d_v(41,asteis,12231,6)
d_v(47,isteis,22231,6)
d_v(53,isteis,32231,6)
d_v(58,aremos,11241,6)
d_v(64,eremos,21241,6)

El tercer argumento del predicado codifica numéricamente la información morfémica según los siguientes criterios:

1º dígito:
atributo: conjugación
valor:
1.- Primera.
2.- Segunda.
3.- Tercera.

2º dígito:
atributo: persona
valor:

0.- No persona.
1.- Primera.
2.- Segunda.
3.- Tercera.

3º dígito:
atributo: número
valor:
0.- No numero.
1.- Singular.
2.- Plural.

4º dígito.
atributo: tiempo
valor:
0.- No tiempo.
1.- Presente.
2.- Pret. Imperfecto.
3.- Pret. Indefinido.
4.- Futuro.
5.- Condicional.

5º dígito:
atributo: modo
valor:
1.- Indicativo.
2.- Subjuntivo.
3.- Imperativo.
4.- Infinitivo.
5.- Gerundio.
6.- Participio.

El último argumento es la longitud de la desinencia. Su función es simplemente algorítmica.

2) Modelos. En esta segunda tabla se hacen corresponder mediante los argumentos del predicado «mod_v», los atributos de modelo, tipo, número de raíz y conjunto de desinencias. De tal forma que cada modelo queda determinado por un tipo (defectivo, unipersonal o sin restricciones flexivas). A su vez para cada modelo y número de raíz de un verbo existe un conjunto de desinencias. Una muestra de esta tabla es:

Para el modelo «1», los cuatro primeros predicados especifican dos tipos («X»: sin restricciones flexivas, «U»: unipersonal). Para el primer tipo y la primera raíz del verbo existe el conjunto de desinencias «1», y para la segunda raíz el conjunto de desinencias «2». En cambio, si el verbo es unipersonal, para la primera raíz es posible el conjunto de desinencias «100» y para la segunda el conjunto de desinencias «101».

```
mod_v(1,X,1,1)
mod_v(1,X,2,2)
```

```
mod_v(1,U,1,100)
mod_v(1,U,2,101)
mod_v(2,X,1,1)
mod_v(2,X,2,2)
mod_v(3,X,1,3)
mod_v(3,X,2,4)
mod_v(4,X,1,5)
mod_v(4,X,2,6)
mod_v(4,D,1,109)
mod_v(4,D,2,110)
mod_v(5,X,1,1)
mod_v(5,X,2,2)
mod_v(5,U,1,100)
mod_v(5,U,2,101)
mod_v(6,X,1,1)
mod_v(6,X,2,2)
mod_v(7,X,1,3)
mod_v(7,X,2,4)
mod_v(7,U,1,102)
mod_v(7,U,2,103)
.....
```

3) Conjunto de desinencias. Esta tabla relaciona los conjuntos de desinencias con las desinencias que los constituyen. La utilidad de esta tabla es evitar una cantidad excesiva de predicados en la tabla anterior. Si prescindiéramos de ella sería necesario especificar cada una de la desinencias posibles en el predicado «mod_s». El primer argumento del predicado «con_d» es el código del conjunto de desinencias, y el segundo el código de desinencia, que es primer argumento en el predicado «d_v».

```
con_d(1,4)
con_d(1,5)
con_d(1,19)
con_d(1,20)
con_d(1,21)
con_d(1,22)
con_d(1,23)
con_d(1,24)
con_d(1,37)
con_d(1,38)
con_d(1,39)
con_d(1,40)
con_d(1,41)
con_d(1,42)
con_d(1,55)
.....
```

ANÁLISIS DE LA MORFOLOGÍA VERBAL

El procedimiento que se sigue es análogo al del análisis nominal.

Dada una forma concreta, por ejemplo «aciertan», el proceso comienza disponiendo un predicado como el siguiente:

f(aciertan, Ref, L, C, R, N, M, T, Des, Descr)

Los argumentos tienen el siguiente significado: Ref: referencia, L: lema, C: categoría, R: raíz, N: número de raíz, M: modelo, T: tipo, Des: desinencia, Descr: descripción.

Las hipótesis sucesivas sobre la segmentación de la forma verbal son las siguientes:

a) Raíz: «aciertan», Desinencia: «»
b) Raíz: «acierta», Desinencia: «n»
c) Raíz: «aciert», Desinencia: «an»

Las dos primeras hipótesis son rechazadas en los primeros pasos del proceso: no existen raíces iguales a las supuestas. En cambio, la comprobación de la tercera hipótesis tendrá éxito según los siguientes procesos de unificación realizados sobre el predicado inicial:

f(aciertan, Ref, L, C, aciert, N, M, T, an, Descr)

En la base de datos de raíces se encuentra un registro de la forma:

Ref:	Raíz:	Nº raíz:	Modelo:
012	aciert	2	1

En el predicado anterior quedarán instanciadas la variable número de raíz con «2» y la variable modelo con el valor «1».

f(aciertan, 012, L, C, aciert, 2, 1, T, an, Descr)

A continuación se busca un predicado «mod_v» cuyo primer argumento (modelo) unifique con «1», y cuyo segundo argumento (número de raíz) unifique con «2». Existe un sólo predicado que cumple esta condición:

mod_v(1,X,2,2)

con lo cual se puede instanciar el argumento «T» de predicado «f» con el valor «X».

La comprobación continúa con el intento de unificación del argumento Desinencia –instanciado ya al valor «an» en el predicado «f»– en la primera tabla («d_v»), en donde se localiza una cláusula de la forma:

d_v(6,an,13211)

En este paso se obtiene un nuevo argumento (código de desinencia) cuyo valor «6» tendrá que pertenecer al conjunto de desinencias «2» obtenido como último argumento del predicado «mod_v». Para comprobar este hecho se recurre a la tercera tabla («con_d») en la que se localiza una cláusula de la forma:

con_d(2,6)

Tras este paso el predicado «f», presenta los siguientes argumentos:

f(aciertan, 012, L, C, aciert, 2, 1, X, an, 13211)

Por último la referencia localizada en la base de datos de infinitivos hará posible la instanciación de las variables referentes al lema y la categoría:

f(aciertan, 012, acertar, verbo, aciert, 2, 1, X, an, 13211)

El sistema no descuida las posibles alternativas de descripción de homógrafos. Para ello, obtenida una descripción, se fuerza el fracaso de la hipótesis ya comprobada y se intentan otras, ya sean para la misma categoría o para otra.

Además, el procesador morfológico permite no sólo el análisis de las formas simples del verbo, sino que además detecta las formas compuestas. También puede segmentar contracciones y formas verbales con enclíticos. Así, el análisis de una forma como «cantándola», procede, en primer lugar, separando el pronombre enclítico, analizándolo y describiéndolo. Posteriormente reconstruirá la forma verbal primigenia «cantando» (sin acento), y procederá al correspondiente análisis.

Bibliografía

Aguirre, E. et al., «Aplicación de la morfología de dos niveles al euskera», *Procesamiento del Lenguaje Natural*, 7, (1989), 87-103.

Clocksin, W. y Mellish, C., *Programación en Prolog*, Barcelona, ed. Gustavo Gili, 1987.

García-Pelayo y Gross, *Larousse de la conjugación*, Barcelona, Larousse, 1989.

Gazdar, G. y Mellish, C., *Natural Language Processing in Prolog: An introduction to Computational Linguistics*, Wokingham, Addinson-Wesley, 1989.

Kartunnen, L., Dowty, D. y Zwicky, A., *Natural Language Parsing*, Cambridge, Univ. Press, 1985.

Kay, M., «Morphological and Syntactical Analisys», *Linguistic Structures Processing*, New York, 1977.

Koskenniemi, K., *Two-Level Morphology*, Univ. Helsinki, 1983.

Meya Llopart, M., y Huber, W., *Lingüística Computacional*, Barcelona, Teide, 1986.

Meya LLopart, M., «Gramática morfémica del español», *R.S.E.L.*, 1985.

Quince mil verbos españoles, Barcelona, Ramón Sopeña, 1980.

Real Academia Española, *Esbozo de una nueva gramática de la lengua española*, Madrid, Espasa Calpe, 1989.

Rodríguez Magro, C., et al., «Clasificación morfológica del léxico castellano para un analizador en ordenador», *Actas del séptimo Congreso Nacional de Lingüística Aplicada*, Univ. Sevilla, 1990, 491-503.