EL TRATAMIENTO DE LA MORFOLOGÍA FLEXIVA DEL CASTELLANO MEDIANTE REGLAS DE DOS NIVELES EN UNA GRAMÁTICA DE UNIFICACIÓN.

Marta Carulla, Auke Oosterhoff
Institut Universitari de Lingüística Aplicada
Universitat Pompeu Fabra
La Rambla 30, Barcelona 08002
carulla@upf.es, oosterho@upf.es

Abstract

This paper describes the strategy adopted to treat the (flexive) morphology of Spanish with the tools and algorithms (i.e. a text handling tool and two level morphology rules) provided by ALEP (Advanced Linguistic Engineering Platform). The morphological analyzer forms part of the Spanish grammar developed within the LSGRAM (Large Scale Grammars, LRE-61029) project, funded by the Comission of the European Union. Our approach has been to segment words into two morphemes, a root and a termination, at the same time regularizing allomorphic and orthographic variations. We explain briefly the underlying philosophy and give some examples to illustrate the fact that all phenomena can be adequately treated by rules that make use of either context information or lexical information (the latter in the form of feature structures).
Keywords: two level morphology

1 Introducción

En este artículo queremos presentar el analizador morfológico que hemos desarrollado en el marco del proyecto europeo LSGRAM (LRE-61029), cuyo objetivo ha sido desarrollar gramáticas de unificación para gran parte de las lenguas comunitarias utilizando la plataforma ALEP ofrecida por la misma UE.

En el marco de la gramática para el español nuestro centro se ocupó principalmente de la implementación de la morfología flexiva.

Para ello la plataforma ALEP ofrece en primer lugar un segmentador de palabras basado en la técnica de dos niveles integrado en el tratamiento de texto (text handling) y en

segundo lugar un analizador (parser) basado en unificación.

El tratamiento de la morfología flexiva, por consiguiente, se divide en dos etapas, la primera en que se segmentan las palabras en morfemas regularizando variaciones alomórficas y ortográficas y la segunda en que se construyen las palabras confiriéndoles la información morfo-sintáctica necesaria para el análisis sintáctico y semántico (cf. Alshawi et.al.).

La función de segmentador es doble: dar un resultado apto para un parsing cuya base son morfemas y permitir un diseño del léxico que evite redundancias debidas a diferentes realizaciones de superficie de los morfemas.

En este artículo nos centraremos sobre todo en las estrategias de segmentación seguidas para alcanzar tal objetivo y en la información léxica necesaria para tratar la segmentación del castellano.

2 El formalismo de dos niveles de la plataforma ALEP

El formalismo de dos niveles de la plataforma ALEP sigue en su concepción básica la propuesta de su creador Kimmo Koskenniemi (1984), con reglas que establecen una proyección entre una cadena de superficie y una cadena léxica (lematizada). La aplicación de la regla puede ser opcional (operador '⇒') u obligatoria (operador '⇔').

```
cadena de superficie    cadena léxica
        [ ] [u,e] [ ] ⇒ [ ] [o] [ ].
```

En este ejemplo se proyecta la cadena 'ue' de la palabra superficial (p.ej. *puedo*) hacia la cadena léxica (p.ej. 'pod').

En las reglas se puede especificar también el contexto, tanto a la derecha como a la izquierda, en que debe estar la cadena afectada y el contexto que se espera como resultado de la regla:

```
[p] [u,e] [d] ⇒ [p] [o] [d].
```

Existe la posibilidad de generalizar sobre los caracteres de la cadena mediante variables que se refieren a conjuntos de caracteres.

```
[A] [u,e] [B] ⇒ [ ] [o] [ ],
        A en {p,m,t,l},
        B en consonantes.
```

Incorporando las propuestas formuladas por Bear (1988) y Trost (1990) entre otros, las reglas de dos niveles de ALEP pueden hacer uso de la información léxica. Las reglas pueden ser aumentadas con una estructura de rasgos que debe ser unificable con la del morfema al cual se aplica la regla:

```
[ ] [u,e] [ ] ⇒ [ ] [o] [ ],
        traíz_verbal:{diptongo = sí}.
```

En este caso, el morfema léxico resultado de la proyección de dos niveles debe ser del tipo 'traíz_verbal' y contener el rasgo 'diptongo = sí'.

El formalismo ofrece también la posibilidad de usar diacríticos (caracteres abstractos) en el léxico. Así, por ejemplo, podemos codificar la vocal temática que esté sujeta a variaciones alomórficas mediante un diacrítico, al cual las reglas de dos niveles proyectan todas las posibles realizaciones superficiales (p.ej. la raíz del verbo *poder* como 'pOd').

3 Estructura del signo lingüístico y codificación léxica

El signo lingüístico en la gramática española está estructurado básicamente siguiendo la propuesta de Pollard y Sag (1992) y es la siguiente:

$$\text{sign}\begin{bmatrix}\text{specs}\\ \text{ortho}\\ \text{tlmws} & \begin{bmatrix}\text{tlm}\\ \text{ws}\end{bmatrix}\\ \text{syn}\\ \text{sem}\end{bmatrix}$$

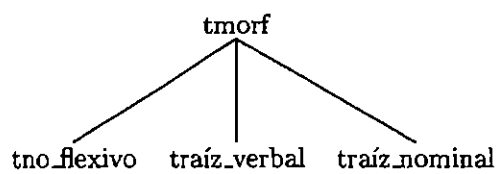
La codificación de los diferentes signos está basada en estructuras de rasgos tipificadas (typed feature structures). La información morfológica se encuentra bajo el rasgo tlmws que a su vez se divide en dos subtipos: 'tlm' que contiene la información necesaria para las reglas de dos niveles y 'ws' donde está codificada la información morfo-sintáctica necesaria para las reglas de estructura de palabra. La información léxica del tipo 'tlm' con sus subtipos y rasgos refleja nuestra concepción general de la segmentación, cuyas características principales son las siguientes:

a) Distinguimos entre flexión nominal, a la cual pertenecen las categorías nominales (nombres, determinantes y adjetivos) y flexión verbal.

b) Seguimos una estrategia de segmentación en dos unidades: raíz y terminación, es decir, consideramos a la terminación como una sola unidad sin hacer distinción entre posibles morfemas de número, persona, tiempo etc.

c) No distinguimos entre terminaciones verbales y nominales para la segmentación, puesto que no se diferencian en la cadena de caracteres sino en la información morfo-sintáctica.

De esta estrategia de segmentación resulta una jerarquía de tipos con un supertipo 'tmorf' que tiene como subtipos 'tno_flexivo' (para las categorías que no flexionan), 'traíz_verbal' y 'traíz_nominal'. Las terminaciones son todas del tipo 'tmorf'.



La relación entre tipos y subtipos es de herencia simple, por lo que los rasgos del supertipo son heredados por sus subtipos.

Estos tipos llevan los rasgos siguientes:

tmorf	
morfema	(rasgo atómico que contiene la cadena de caracteres)
ms_key	(rasgo atómico que contiene la forma lematizada del morfema.)
último = si/no	(rasgo que indica la posición que el morfema puede ocupar dentro de la palabra
eliminación_e = sí/no	(rasgo que hace referencia a la necesidad de borrar el infijo '-e-' en la formación de plural cuando este no lleva información morfo-sintáctica (p.ej. *ordenador-e-s* vs. *profesor-es*)
tno_flexivo	(no contiene rasgos específicos de su tipo)
último = sí	
traíz_verbal	
diptongo = sí/no	
diptongo_red = sí/no	
cierre = sí/no	
último = no	
traíz_nominal	(no contiene rasgos específicos de su tipo)
último = sí/no/_	(la variable anónima '_' es para raíces que no requieren necesariamente un sufijo flexivo)

De la combinación de estos rasgos resultan cuatro grupos de raíces nominales:

1.	último = no, eliminación_e = no	p.ej. *president* que produce los siguientes resultados de segmentación: *president+e, president+es, president+a, president+as*
2.	último = _ eliminación_e = sí	p.ej. *acción* o *profesor* que producen los siguientes resultados de segmentación: *acción+, accion+s, profesor+, profesor+s, profesor+a, profesor+as*
3.	último = sí eliminación_e = sí	p.ej. *crisis* o *miércoles* (que no segmentan)
4.	último = _ eliminación_e = no	p.ej. *mercado* que produce los siguientes resultados de segmentación: *mercado+, mercado+s*

Y cinco grupos de raíces verbales:

1.	diptongo = no cierre = no diptongo_red = no	p.ej. *am+* (amar), *acced+* (acceder), *inclu+* (incluir), verbos regulares sin variaciones alomórficas
2.	diptongo = sí diptongo_red = no cierre = no	p.ej. *comEnz+* (comenzarar, *mOstr+* (mostrar), que presentan diptongación en todas las formas posibles.
3.	diptongo = no diptongo_red = si cierre = sí	p.ej. *tEN+* (tener) y *vEN+* (venir) y compuestos, que presentan diptongación reducida (sólo en indicativo: *tiene*) y además cierran la vocal temática (*tuvo, tuviste*, etc.)
4.	diptongo = sí diptongo_red = no cierre = sí	p.ej. *invErt+* (invertir), *pOd+* (poder), *prefEr+* (preferir), que diptongan y además cierran la vocal temática
5.	diptongo = no diptongo_red = no cierre = sí	p.ej. *compEt+* (competir), *rEg+* (regir), que cierran la vocal temtica en determinadas formas (*compito, rige*, etc.)

4 Las reglas de dos niveles

4.1 Las tres reglas básicas

Para cada aplicación de las reglas de dos niveles se necesitan tres reglas básicas, una primera para marcar el final de la palabra, una segunda para segmentar la palabra en raíz y terminación (si la hay) y una tercera que copia un carácter de superfície a un carácter léxico. Nosotros hemos aumentado ligeramente las reglas añadiéndoles restricciones de tipo. De esta manera las tres reglas quedan expresadas como sigue:

1. Regla que cambia el final de palabra (=) en un linde morfemático (+):

```
tlm_regla(final_palabra,
        [ ] [=] [ ] ⇒ [ ] [+] [ ],
        tlmws:{tmorf:{último = sí}}).
```

2. Regla que produce la segmentación de palabras en raíces y terminaciones. Esta regla sólo se puede aplicar a aquellas palabras que tienen el rasgo último = no. Como teníamos el rasgo eliminación_e = sí/no, que sólo es relevante en el caso de los nombres, la regla ha sido divivida en dos, una específicamente para verbos, y otra específicamente para nombres y adjetivos con el rasgo eliminación_e = no. (La regla específica que elimina la 'e' inserta un linde morfemático a la vez que elimina la 'e' del plural.)

```
tlm_regla(segment_v,
        [ ][ ][ ] ⇒ [ ][+][ ],
        tlmws:{tmorf ⇒ traíz_verbal:{último = no}}).

tlm_regla(segment_n,
        [ ][ ][ ] ⇒ [ ][+][ ],
        tlmws:{tmorf ⇒ traíz_nominal:{último = no, eliminación_e = no}}).
```

3. Regla que copia un carácter de superfície a un carácter léxico.

```
tlm_regla(identidad,
        [ ][X][ ] ⇒ [ ][X][ ]).
```

4.2 Reglas basadas principalmente en la descripción del contexto

Estas se aplican a aquellos casos en que el contexto es la base para restringir la aplicación de la regla. Por ejemplo, en los casos de variación ortográfica, como *c/z, j/g, gu/g, qu/q* o la pérdida de acento en las terminaciones -ón al formar el plural en -ones (*acción, acciones, millón, millones,* etc.), el contexto da la información necessaria para la aplicación de la regla recurriendo a las estructuras de rasgos sólo para determinar el tipo de morfema al cual se debe aplicar la regla. La regla para la pérdida del acento en el plural de los nombres acabados en -ón es como sigue:

```
tlm_regla(acento,
        [ ] [o] [n,e,s,=] ⇔ [ ] [ó] [ ],
        traíz_nominal).
```

Un caso algo más complicado es el siguiente. Hay una serie de verbos que cambian la vocal de la terminación en futuro y condicional por una *d*. Son: *poner, tener, valer, salir* y sus compuestos, que tienen formas como *pondré, tendré, valdría, saldría*, etc.[1]

poner, tener, valer dr ⟶ er
salir, venir dr ⟶ ir

Estos fenómenos irregulares se dan en el mismo contexto derecho, es decir, con las terminaciones -é, -ás, -á, -emos, -éis, -án, -ía, -ías, -íamos, -íais, -ían. Para describir este contexto unívocamente es suficiente especificar el primer carácter de la terminación, o sea, {á,é,e,í}. Para que las reglas se apliquen únicamente a aquellas formas a las que se tienen que aplicar, es necesario especificar también el contexto izquierdo. El resultado final son dos reglas, una que sustituye la *d* por una *e* (para los verbos *poner, tener y valer* y sus compuestos), y otra que sustituye la *d* por una *i* (para los verbos *salir* y *venir* y sus compuestos):

```
tlm_regla(conversión_dr_1,
        [P,Q,R] [d,r] [X] ⇔ [ ] [e,r] [ ],
        tlmws:{tmorf ⇒ traíz_verbal:{}},
        P no en {s},
        Q en {a,e,o},
        R en {l,n},
        X en {á,é,e,í}).

tlm_regla(conversión_dr_2,
        [P] [d,r] [X] ⇔ [ ] [i,r] [ ],
        tlmws:{tmorph ⇒traíz_verbal:{}},
        P en {l,n},
        X en {á,é,e,í}).
```

4.3 Reglas basadas principalmente en la información léxica

Las reglas han sido aumentadas con estructuras de rasgos para describir correctamente otros tipos de fenomenos, para los cuales no basta la descripción del contexto. Son cambios que no responden a necesidades fonológicas u ortográficas, sino que forman parte de las características idiosincráticas de cada raíz verbal, como la diptongación y el cierre vocálico. Como tales quedan codificadas en forma de rasgos en la entrada léxica de las raíces verbales y la aplicación de la regla de dos niveles queda restringida a la presencia de estos rasgos.

Por otra parte estos cambios se producen sólo en determinadas formas, por lo cual los contextos, aunque no suficientes para restringir la aplicación de la regla sí son necesarios para su correcta aplicación.

El tratamiento de fenómenos como la diptongación y el cierre vocálico, pues, se basa en combinar las estrategias tanto de especificación de contexto como de restricción mediante estructuras de rasgos en las reglas de dos niveles. Queremos ejemplificarlo mediante algunas reglas que tratan la diptongación:

La vocal temática de algunos verbos se diptonga en el presente de indicativo (primera, segunda y tercera persona singular y la tercera persona plural) y en el presente de sub-

[1] Otros verbos sólo pierden la vocal: *caber, haber, saber, querer, poder* y sus compuestos. Para ellos tenemos otras reglas.

juntivo (primera y segunda persona singular y tercera persona plural), es decir, cuando la raíz va seguida de las terminaciones -o, -as, -a, -an, -e, -es, -en. La descripción exacta del contexto derecho requiere de dos reglas, una para el contexto {o,a,e} seguido de final de palabra (=) y otra para el contexto {a,e}, {s,n}, {=}. Para la diptongación de la vocal temática 'o' en 'ue' resulta en las reglas siguientes:

```
tlm_regla(ue_diptongo_1
        [ ] [u,e] [A,B,=] ⇔ [ ]['O'][ ],
        tlmws:{tmorf⇒ traíz_verbal:{diptongo=sí}},
        A en {b,c,d,g,l,n,ñ,r,v,z},
        B en {o,a,e}).

tlm_regla(ue_diptongo_2
        [ ] [u,e] [A,B,C,=] ⇔ [ ]['O'][ ],
        tlmws:{tmorf⇒ traíz_verbal:{diptongo=sí}},
        A en {b,c,d,g,l,n,ñ,r,v,z},
        B en {a,e}
        C en {s,n}).
```

La variable A se refiere al conjunto de consonantes con que terminan las raíces verbales. Su especificación contribuye a una mayor restricción en la aplicación de las reglas. Las dos reglas anteriores se aplican cuando la raíz termina en una consonante (*rodar, doler,* etc.). De forma análoga se han escrito dos pares de reglas más, uno para raíces terminadas en dos consonantes (*dormir, volver*) y otro para aquellas terminadas en tres (*demostrar, encontrar*):

```
tlm_regla(ue_diptongo_3,
        [ ] [u,e] [A,B,C,=] ⇔ [ ]['O'][ ],
        tlmws:{tmorf⇒ traíz_verbal:{diptongo=sí}},
        A en {l,r,s,n,m,b},
        B en {v,d,t,z,g,l,p,m,c},
        C en {o,a,e}).

tlm_regla(ue_diptongo_4,
        [ ] [u,e] [A,B,C,D,=] ⇔ [ ]['O'][ ],
        tlmws:{tmorf⇒ traíz_verbal:{diptongo=sí}},
        A en {l,r,s,n,m,b},
        B en {v,d,t,z,g,l,p,m,c},
        C en {a,e},
        D en {s,n}).

tlm_regla(ue_diptongo_5,
        [ ] [u,e] [A,t,r,B,=] ⇔ [ ]['O'][ ],
        tlmws:{tmorf⇒ traíz_verbal:{diptongo=sí}},
        A en {s,n},
        B en {o,a,e}).

tlm_regla(ue_diptongo_6,
        [ ] [u,e] [A,t,r,B,C,=] ⇔ [ ]['O'][ ],
        tlmws:{tmorf⇒ traíz_verbal:{diptongo=sí}},
        A en {s,n},
        B en {a,e},
        C en {s,n}).
```

Nótese que proyectamos el diptongo a un diacrítico ('O' en este caso). El uso del diacrítico no es necesario para la correcta aplicación de las reglas, pero usándolo aumentamos la

eficiencia, puesto que el acceso al léxico se produce primero a partir de las cadenas de caracteres que resultan de la aplicación de las reglas y sólo en el último estadio del análisis se produce la unificación de la estructura de rasgos especificada en la regla de dos niveles con la entrada léxica. Haciendo uso de diacríticos reducimos las posibilidades de selección léxica desde el principio. Sin embargo ello requiere que formulemos una regla que proyecte la vocal temática hacia el diacrítico que la representa:

```
tlm_regla(o_diacrítico,
        [ ] [o] [ ] ⇔ [ ]['O'][ ],
        tlmws:{tmorf⇒ traíz_verbal:{diptongo=sí}}).
```

De la misma manera que tratamos la diptongación *o/ue* presentada aquí, tratamos los demás fenómenos de diptongación (*e/ie* y *i/ie*), cuyos contextos a la derecha son idénticos.[2]

5 Cobertura

El módulo de segmentación consta de un conjunto de 78 reglas de dos niveles con la cobertura siguiente:

- fenómenos regulares de la flexión nominal y verbal
- fenómenos irregulares de la flexión verbal:
  - diptongación de la vocal temática
  - cierre de la vocal temática
  - pérdida de vocal de la terminación en futuro y condicional, p.ej. *poder, podré*
  - pérdida de la *i* en determinadas terminaciones de verbos (de la segunda y tercera conjugación) cuya raíz termina en *ll, y, ñ* y *j*, p.ej. *engullir, engulleron.*
  - variación *ng/n*
  - variación *i/y*
  - pérdida de los caracteres *g, y, z, ig*
  - variación *b/p*
  - verbos altamente idiosincráticos *ser, estar, decir, caber, saber*
- variaciones ortográficas:
  - pérdida del acento gráfico de los nombres terminados en *-ón*
  - variación *c/z*
  - variación *qu/c*

[2]Hay algunos verbos que sólo diptongan con algunas de las terminaciones mencionadas. Son los verbos *venir, tener* y sus compuestos, que diptongan en presente indicativo (segunda y tercera persona del singular y tercera del plural).
En estos casos, aparte de regularizar el diptongo, hay que evitar que se apliquen las reglas de diptongación anteriores (sobretodo en generación). Para ello hemos caracterizado estos verbos con el rasgo distintivo 'diptongo_red = sí' en el léxico, a la vez que les asignamos 'diptongo = no'.

- variación *gu/g*
- variación *j/g*
- variación *u/ü*

El número de entradas léxicas es de 429, de las cuales 119 son terminaciones.

6 Conclusión

La experiencia ha demostrado que el formalismo de dos niveles de ALEP constituye una herramienta expresiva y eficiente para segmentar y regularizar las palabras flexionadas del castellano. Se trata de una herramienta mixta, que aún siendo independiente del parsing puede acceder a la información léxica contenida en el signo lingüístico.

La estrategia de implementación ha tenido como principales objetivos la eficiencia, resultando en unos tiempos de segmentación para una frase media (15 palabras) de 0.04 segundos en una máquina Sparc Ultra.

Para ello ha sido decisivo trabajar sobre la base de dos unidades de segmentación, tratando la terminación como una única unidad. posibilidad que ofrecen las lenguas románicas ya que los morfemas flexivos se concatenan a la derecha. Para el tratamiento de la alomorfia y las variaciones ortográficas ha significado explotar al máximo la descripción del contexto de las reglas de dos niveles y también el uso de diacríticos.

Cabe señalar, sin embargo, que hemos utilizado las estructuras de rasgos principalmente para restringir la aplicación de las reglas de dos niveles sin explotar a fondo la posibilidad de guardar información sobre qué reglas se han aplicado. Esto es particularmente necesario cuando la regularización de distintos fenómenos resulta en segmentaciones idénticas, como es el caso con *puedo* $\longrightarrow$ *pOd+o*, *pudo* $\longrightarrow$ *pOd+o*.

Bibliografía

Alshawi, H., Arnold, D.J.; Backofen, R., Carter, D.M.; Lindop, J.; Netter, K; Pulman, S.; Tsuji, J. y Uskoreit, H., 1991. "Eurotra ET6/1: Rule Formalism and Virtual Machine Design Study (final Report)". CEC, Luxemburgo.

Bear, J., 1988. "Morphology with Two Level Rules and Negative Rule Features". En: *Proceedings of the 12th International Conference on Computational Linguistics (Coling-88)*, Budapest, Hungría.

Koskenniemi, K., 1983. "Two Level Morphology". Tesis doctoral, Universidad de Helsinki, Helsinki, Finlandia.

Pollard, C. y Sag, I. A., 1992. "Head-Driven Phrase Structure Grammar". Center for the Study of Language and Information, Stanford, U.S.A.

Trost, H., 1990. "The Application of Two Level Morphology to Non-Concatenative German Morphology". En: *Proceedings of the 13th International Conference on Computational Linguistics (Coling-90)*, Helsinki, Finlandia.