MORFOLOGÍA DE ESTADOS FINITOS

Iñaki Alegria
Informatika Fakultatea (UPV/EHU)
acpalloi@si.ehu.es

En este artículo se presentan los distintos modelos computacionales para la morfología y dentro de ellos se profundiza en el modelo más extendido de estados finitos: la morfología (fonología) de dos niveles.

Antes de examinar los diferentes formalismos morfológicos se introducen las características y fundamentos morfológicos elementales que van a ayudar a analizar los distintos modelos. Tomando como base esas características se propone una clasificación, ubicando dentro de la misma distintos sistemas que aparecen en la bibliografía, algunos de los cuales son presentados antes de profundizar en uno de ellos: la morfología de dos niveles.

Una vez detallados los elementos básicos de la morfología de dos niveles se entra a discutir sus ventajas e inconvenientes, pasándose, a continuación, a examinar las mejoras propuestas para su parte más discutible, la morfotáctica. Por último, se examina someramente la complejidad del modelo computacional de dos niveles para comprobar su adecuación a aplicaciones reales.

Finalmente se ofrece una bibliografía básica entre la que cabe destacar el libro de R. Sproat "Morphology and Computation" (1992), del cual se han obtenido parte de la información que aquí se presenta.

I Análisis morfológico: introducción.

I.1 Aplicaciones

Un analizador morfológico automático es una herramienta básica para cualquier sistema de procesamiento de lenguaje natural (PLN). Aunque en algunos sistemas se venía prescindiendo del tratamiento morfológico —sobre todo porque el idioma a tratar era el inglés— usándose diccionarios completos de formas también llamados léxicos desplegados, su uso se ha

hecho prácticamente universal como base a cualquier otro tratamiento lingüístico automatizado por las siguientes razones:

- En lenguas de flexión rica, y sobre todo en lenguas aglutinantes, el uso de diccionarios completos es totalmente inviable.
- Aún en lenguas con morfología sencilla el tratamiento morfológico además de la ventaja evidente de la compactación del léxico, aumenta la flexibilidad del sistema, consiguiéndose también una descripción más precisa del idioma a tratar.

Por lo tanto el procesamiento morfológico suele ser una tarea básica para sistemas PLN como paso previo a otros tratamientos más complejos: sintáxis, semántica, etc. Pero no solo eso, el tratamiento morfológico en sí también es útil para desarrollar ciertas herramientas, pudiéndose nombrar como aplicaciones directas del mismo las siguientes:

- verificación/corrección ortográfica básica: para reconocer si las palabras corresponden al idioma, hay dos opciones básicas: contrastarlas con un léxico desplegado o ver si tienen descomposición morfológica correcta. Este último tratamiento es el más conveniente y se usa tanto en verificación de textos, lectura OCR y reconocimiento del habla.
- lematización: muchas veces, sobre todo en lenguas de flexión rica, para aplicaciones de recuperación de la información, así como para aplicaciones lexicográficas, la base del tratamiento no es la forma sino el lema, y una descomposición morfológica sirve para obtener el lema a partir de la forma.
- otras aplicaciones: en la enseñanza de idiomas asistida por ordenador, la generación de textos, el análisis y generación de diálogos, así como en otro gran número de aplicaciones el tratamiento morfológico tiene un papel fundamental.

Además de las aplicaciones vistas para la morfología cabría añadir las aplicaciones de la fonología, ya que como se expondrá posteriormente, parte de los formalismos morfológicos son aplicables al tratamiento fonológico.

I.2 Conceptos

Sin intención de ser exhaustivos vamos a repasar los conceptos fundamentales de la morfología, ya que serán necesarios para comparar y clasificar los distintos formalismos que se van a presentar.

El camino que hemos elegido para exponer estos conceptos es responder a las preguntas que se hacen en el libro mencionado (Sproat, 92:17):

> "In particular, I shall discuss the following issues:
> What sort of things can morphology mark in different languages?
> How are words built up from smaller meaningful units-morphemes? ...
> What are the constraints on the order of morphemes within words?
> Do phonological rules complicate the problem of morphological analysis?"

Contestando a estas preguntas aparecen los conceptos fundamentales de morfología que nos interesan para su automatización:

- Respecto a las funciones de la morfología tres son las fundamentales: *flexión*, *derivación* y *composición*. La primera responde a la sintáxis, normalmente es regular según la categoría y no modifica la función sintáctica. La derivación no es regular y puede provocar cambio de categoría. La composición se produce al unir varios lemas y su tratamiento suele ser más complejo ya que puede salirse del ámbito de la palabra.
- Respecto a la forma de combinar los morfemas se pueden producir distintos fenómenos: desde el simple *encadenamiento* en base a prefijos y sufijos habitual en las lenguas cercanas, hasta modelos más complejos como el modelo *raiz-patrón* que se da por ejemplo en el árabe. Hay otros fenómenos intermedios como el encadenamiento con afijos, la duplicación, etc.
- Al conjunto de las posibles combinaciones o restricciones al combinar morfemas se suele llamar *morfotáctica*, y aunque en general solo suele depender de los morfemas contiguos, dependiendo del idioma se dan diferentes grados de regularidad y distancia.
- Los cambios fonológicos pueden ser directamente de carácter fonológico o de carácter eminentemente ortográfico, y por eso los vamos a denominar *morfofonológicos* . En la bibliografía al hablar sobre este fenómeno también se usa el término *morfografémica*. El número de cambios y las condiciones en que se producen difieren muchos entre las distintas lenguas. Algunos sistemas morfológicos, al hacer frente a este fenómeno, utilizan *alomorfos*, es decir, distintas representaciones para un mismo morfema. Como ejemplo de cambio morfofonológico complejo tenemos la

armonía vocal que se da algunas lenguas, donde un cambio en una vocal genera cambios en el resto de las vocales.

II Modelos computacionales de morfología y algunos ejemplos.

En este apartado pasamos a describir los modelos computacionales que se han definido cuando se han diseñado analizadores y/o generadores morfológicos. Nos vamos a centrar en la resolución de morfología basada en encadenamiento de morfemas, aunque nombraremos los problemas que se dan en combinaciones más complejas como las que se dan en lenguas semíticas.

II.1 Modelos computacionales: criterios de clasificación

Como consecuencia de que la flexión del inglés es muy sencilla[1] el análisis y la síntesis morfológica por ordenador no había sido estudiada a fondo (Winograd, 83). Hasta hace pocos años el uso de sistemas primitivos que no separaban conocimiento lingüístico y programa era habitual. En los últimos años, sin embargo, debido tanto al desarrollo de sistemas automáticos para otras lenguas como al análisis basado en corpus, han proliferado los trabajos en este campo.

Hoy en día se puede encontrar amplia bibliografía sobre procesadores[2] morfológicos donde se detallan distintas visiones y características sobre este problema. Para llevar a cabo una comparación entre las distintas aproximaciones al problema, vamos a prefijar una serie de criterios. Siguiendo la linea de las preguntas enunciadas en el apartado anterior podemos establecer los siguientes criterios:

1) ***Poder expresivo*** del formalismo o modelo propuesto, es decir, conjunto de fenómenos morfológicos que pueden ser expresados o analizados por el modelo. En este mismo criterio vamos a integrar la posibilidad del modelo de llevar a cabo tanto análisis

[1] Aunque la morfología del inglés se suele considerar simple esto debe ser matizado ya que como subraya Sproat (1992:152-53) la complejidad morfológica se suele definir en función de la longitud de las palabras o el número de morfemas pero no en función de la regularidad de los encadenamientos y los cambios que se producen.

[2] Optamos por el término procesador morfólogico para describir el programa que pude llevar a cabo tanto análisis como generación morfológica.

como síntesis o generación morfológica, en contraposición con solo poder realizar uno de los procesos.

2) Forma de abordar la morfología. Influenciados principalmente por las teorías lingüísticas y también por la estructura lingüística y el modelo computacional en que se apoya el formalismo se suelen distinguir dos aproximaciones:
   - basadas en léxico: raíces y afijos (morfemas) son las unidades básicas y los elementos que gobiernan el proceso morfológico.
   - basadas en paradigma: los paradigmas son la base del sistema y el resto de elementos están en función de ellos (Calder, 89; Anick & Artemieff, 92). En estos sistemas la organización del léxico suele estar en función del paradigma correspondiente.

   La mayoría de los sistemas se basan en el primer modelo de los descritos y los sistemas que vamos a describir lo seguirán.

3) Modo de resolución de la **morfotáctica**: forma de especificar las relaciones posibles entre morfemas. Se suelen dividir en dos grandes grupos: morfotáctica de estados finitos y mecanismos de unificación. En el primer caso las relaciones entre morfemas pueden ser vistas como un grafo siendo los morfemas los nodos, y los encadenamientos posibles los arcos. Los mecanismos de unificación se basan en las gramáticas que se suelen utilizar en sintáxis y son más potentes y flexibles que los de estados finitos, aunque computacionalmente son más complejos. Los mecanismos de unificación se usan basicamente en el modelo paradigmático (de Smedt, 84; Calder, 89; Anick & Artemieff, 92).

4) Especificación de los **cambios fonológicos**. Aunque en la bibliografía aparecen varios métodos, sobresalen entre ellos dos: métodos *ad-hoc* por programa que eran habituales hasta hace algunos años, y métodos basados en *traductores*[1] *de estados finitos* bastante habituales hoy en día.

[1] *traductores o transducers*: automatas que tienen n-tuplas como etiquetas, o lo que lo mismo, grafos dirigidos finitos con n-tuplas como arcos.

5) Elementos que se almacenan en el **léxico**. Aunque hay sistemas que funcionan sin las raíces, lo habitual es almacenar morfemas (raíces o lemas y afijos) en el léxico. De todas formas en sistemas que no manejan los cambios fonológicos lo que se almacena son segmentos de palabras y no morfemas. Además de las posibilidades anteriores también existen otras opciones como la basada en sílabas (Cahill, 90).

   Dentro de este criterio también cabe distinguir las siguientes características:

   - posible almacenamiento de alomorfos, es decir, si se utiliza más de una representación para una misma unidad léxica.
   - deformación de los morfemas, es decir, almacenamiento en forma no convencional o canónica. P.ej. en el sistema KIMMO se usan diacríticos mezclados con los morfemas para controlar la aplicación de las reglas morfofonológicas.

La **eficiencia** no la hemos incluido como criterio de clasificación porque, aún siendo una característica muy importante, además de depender del resto de los criterios y sobre todo de la implementación, hemos querido centrarnos fundamentalmente en la formalización.

La **sobregeneración** más que del modelo depende de la implementación —es fundamental la granularidad elegida— aunque en determinados modelos evitarla es más complejo.

Otro concepto importante es el de **cobertura** del idioma, que hace la descripción morfológica que, como la sobregeneración, suele depender más de la descripción concreta que del formalismo. La cobertura también suele estar relacionada con la aplicación para la que se desarrolla el procesador, ya que mientras que para un verificador ortográfico nos interesa una cobertura del idioma estándar, para un etiquetador nos interesará la máxima cobertura posible.

II.2 Ejemplos

A continuación vamos a presentar una serie de procesadores morfológicos. La presentación no es exhaustiva ya que solo se describen algunos ejemplos representativos.

II.2.1 DECOMP

Este analizador aplicado a tratamiento del habla se desarrolló a mediados de los 60 en el MIT dentro de un proyecto llamado *MITalk* (Allen *et al.*, 87) y es uno de los primeros analizadores morfológicos conocidos. Aunque usándolo se aumentaba la cobertura del sistema y se reducía el tamaño del léxico, la razón fundamental para su uso es la relación que existe en inglés entre morfología y pronunciación.

Las características fundamentales de DECOMP son las siguientes:

- Trata flexión derivación y composición a nivel de palabra. No vale para generación morfológica.
- Utiliza morfotáctica de estados finitos, basada en tipos de morfemas y definida por un conjunto de reglas simples.
- Los cambios fonológicos se describen por reglas muy simples. Estos cambios están limitados a la frontera entre morfemas y solamente se manejan cambios elementales.
- El léxico lo forman 12.000 que se obtuvieron a partir de 50.000 palabras del Brown corpus. A cada morfema se le asigna un código, que define su tipo y controla la aplicación de las reglas, indicando si los cambios son obligados, prohibidos u opcionales.

El algoritmo de análisis procesa la palabra de derecha a izquierda, es recursivo y asigna distintos pesos a las transiciones entre estados que componen la morfotáctica; con lo que se consigue que ciertos análisis tengan prioridad y que mejore la eficiencia.

II.2.2 ATEF

ATEF es un modelo morfológico general para cualquier idioma desarrollado dentro de un entorno de traducción automática en los laboratorios GETA (GETA, 82) al final de la década de los 70, y está estrechamente ligado con el analizador/generador sintáctico ROBRA. Los principales componentes de ATEF son las siguientes:

- Variables: variables simbólicas que obtienen los resultados que se van produciendo durante el análisis morfológico.
- Diccionarios: léxicos que recogen información respecto a los morfemas. Son un máximo de siete y la información que contiene cada elemento es la siguiente: segmento de palabra, formato correspondiente, forma canónica e informaciones morfológicas.

- Formatos: El formato que se especifica en el diccionario sirve para agrupar los elementos del mismo que tienen características morfotácticas comunes.
- Gramática: reglas que se activan en función de los formatos para actualización de variables, apertura/cierre de subléxicos, etc.

El programa, en función de los diccionarios abiertos, trata constantemente de identificar segmentos en las palabras a analizar, y cuando así sucede, además de asignar valores a las variables aplica las reglas correspondientes al formato.

Aunque la descripción de la morfotáctica y del tratamiento morfosintáctico es fácil y flexible, el tratamiento de los cambios morfofonológicos es inadecuado, ya que, aunque sólo sea para cambios fonológicos muy simples, se deben usar trucos y métodos indirectos.

II.2.3 KIMMO

La principal novedad del formalismo de dos niveles que propuso Koskenniemi (1983) en su tesis, es la descripción de los cambios morfofonológicos cuya declaración era engorrosa o imposible en sistemas anteriores. Esta característica lo ha hecho atractivo a muchos idiomas con rica morfología, y hace que sea un modelo mucho más general que los anteriores. Las razones de su éxito son las siguientes: formalismo potente, general y eficiente[1].

Para expresar los cambios morfofonológicos se utilizan reglas paralelas de dos niveles que se compilan como traductores (*transducers*) de estados finitos. De todas formas KIMMO no fue el primer modelo que propuso reglas paralelas para describir los cambios morfofonológicos.

Aunque las características de la morfología de dos niveles la examinaremos en profundidad, nombraremos aquí las más importantes:

- Es válida tanto para análisis como para generación.
- Morfotáctica lineal basada en subléxicos. Es un mecanismo muy simple pero que, a veces, no resulta suficientemente expresivo.
- Los cambios morfofonológicos son el aspecto más sobresaliente de este modelo.

[1] Sobre esta característica nos extenderemos posteriormente.

- Los elementos del léxico son morfemas y pueden ser utilizados alomorfos. El uso de diacríticos —caracteres que regulan la aplicación de reglas— es casi obligado en principio.

Aunque es fácil encontrar más referencias en la literatura, a continuación nombramos un conjunto importante de las mismas:

- Mejoras sobre el modelo: (Karttunen *et al.*, 87), (Kay, 87), (Ritchie *et al.*, 87), (Bear, 88), (Trost, 90), (Karttunen *et al.*, 92), (Karttunen, 93).
- Implementaciones (sin grandes modificaciones): (Karttunen & Wittenburg, 83), (Karlsson, 92), (Clemenceau & Roche, 93), (Oflazer, 94), (Kim *et al.*, 94), (Kiraz, 94), (Alegria, 95).
- Software de libre distribución: PC-KIMMO (Antworth, 90), (Karp *et al.*, 92), Pc-parse (mensajes a *pc-parse-owner@sil.org*).

II.2.4 Tzoukermann & Liberman

Tzoukermann y Liberman (1990) proponen un sistema de análisis y generación que une algunas de las características de los dos previamente presentados. En él la especificación lingüística y la información que trata el procesador son bastante diferentes ya que entre ambos se produce una compilación.

En esta propuesta para el tratamiento del castellano, hecha en los laboratorios AT&T, se utilizan las ideas del formalismo de dos niveles para hacer la especificación lingüística, pero de cara a la eficiencia las reglas se compilan con el léxico y se obtienen automáticamente todos los segmentos de palabras correspondientes a los alomorfos, evitándose el tratamiento morfofonólogico on-line. Además, como resultado de la compilación se obtiene también el grafo morfotáctico. Por lo tanto este sistema es similar al KIMMO a nivel de especificación pero más parecido a ATEF a la hora de la ejecución.

II.3 Una clasificación

Después de los ejemplos, en la Figura 1 proponemos una clasificación para los distintos sistemas basada en una clara división entre los tratamientos morfotácticos y morfofonológicos.

Un sistema para poder ser incluido en esta clasificación debe cumplir los siguientes requisitos: estar basado en léxico de morfemas y describir

fenómenos morfológicos a nivel de encadenamiento de morfemas. Cumpliéndose lo anterior cualquier nuevo sistema puede ser incluido en dicha clasificación. Cuando los sistemas se basan en otros modelos hacer comparaciones resulta muy complicado.

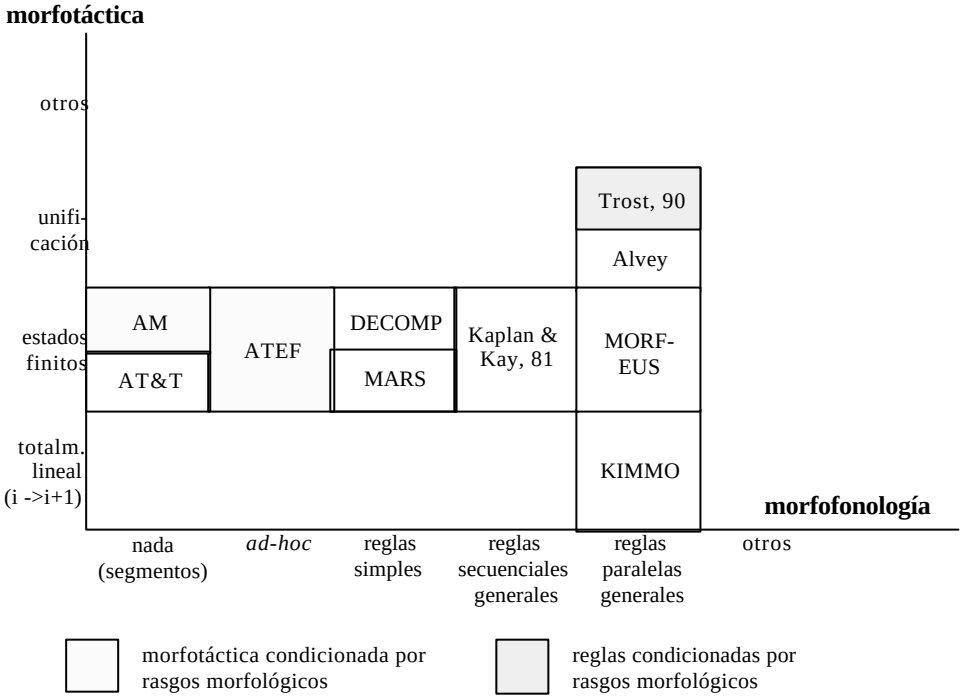

Fig. 1.- Clasificación para los procesadores morfológicos

III Morfología de dos niveles.

Como ya hemos mencionado, en 1983 Koskenniemi definió el modelo computacional para morfología conocido como modelo de dos niveles. Este modelo dio un gran impulso al tratamiento morfo(fono)lógico y ha tenido gran aceptación debido a las siguientes características:

- Es un modelo general, por lo menos a nivel de encadenamiento de morfemas, y aplicable a la mayoría de las lenguas.
- Diferencia totalmente el conocimiento lingüístico y el algoritmo, por lo que el mismo programa es aplicable a diferentes lenguas.
- Es válido tanto para análisis como para generación.
- Se diferencian para cada palabra dos niveles o representaciones, el convencional o de *superficie* y el de profundidad o *léxico*. Debido a esta característica se pueden evitar los alomorfos.
- Para el tratamiento de los cambio fonológicos, en lugar de utilizar las reglas de reescritura propias de la fonología generativa se

utilizan reglas paralelas, lo que resulta más sencillo tanto desde el punto de vista conceptual como computacional.
- La complejidad computacional no es demasiado elevada, lo que permite la construcción de sistemas reales en microordenadores.

A continuación se examinarán los componentes fundamentales y las mejoras propuesta y críticas hechas a este formalismo.

III.1 El sistema léxico.

El sistema léxico almacena el conjunto de morfemas y la información morfotáctica. Está compuesto por los siguientes tres elementos básicos: entradas léxicas, subléxicos y clases de continuación.

Cada **entrada léxica** se compone de tres campos:
- Expresión léxica: es una secuencia de caracteres léxicos. Estos caracteres pueden ser los *convencionales* o de superficie, *morfofonemas* y *marcas de selección* (diacríticos).
- *Clase de continuación*. sirve para indicar el conjunto de morfemas que pueden encadenarse posteriormente.
- *información morfológica*, información que se quiere obtener como resultado del análisis.

Los elementos del léxico están agrupados en **subléxicos** según sus características morfotácticas, y más concretamente en función del conjunto de morfemas que les pueden preceder. Debido a esto la organización en subléxicos resulta a veces artificial desde el punto de vista lingüístico. La definición de un subléxico consiste en su nombre o identificador, sus características y el conjunto de entradas que lo componen. Las características sirven para marcar subléxicos con rasgos morfológicos comunes.

Una **clase de continuación** es un conjunto de subléxicos, que constituye una unidad desde el punto de vista de la morfotáctica y puede ser asimilada a un paradigma. Especifica que cualquiera de los morfemas incluidos en los subléxicos pertenencientes a la clase de continuación pueden ser encadenados inmediatamente después del especificado —en la figura 2 se ofrece un ejemplo simplificado para el euskera.

El poder descriptivo de este formalismo para la morfotáctica puede ser insuficiente —dependencias a distancia por ejemplo—, por lo que, como veremos después, se han propuesto diversas mejoras para aumentarlo.

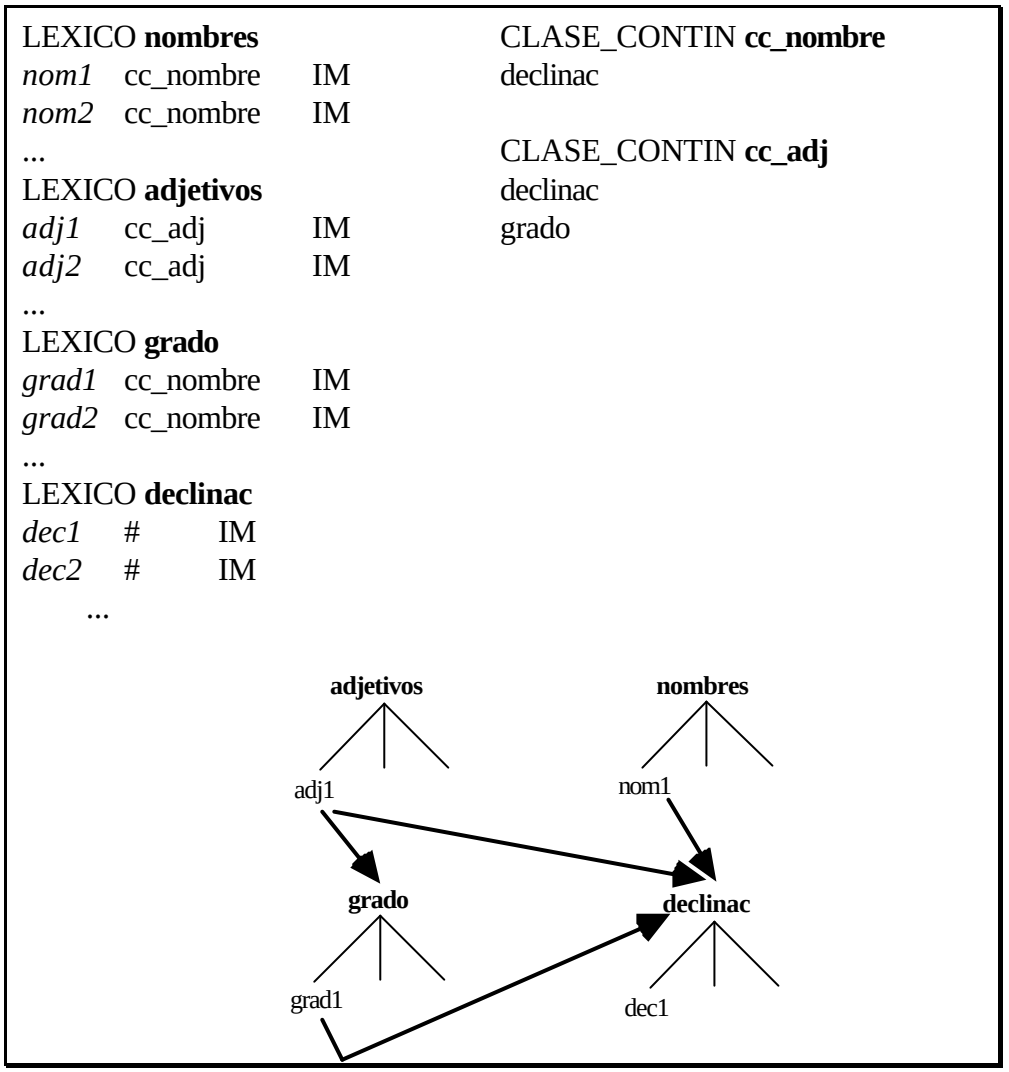

Fig. 2.- Ejemplo de un pequeño sistema léxico

Debido a la existencia de las reglas para expresar los cambios morfofonológicos, no es necesario definir **alomorfos** en el léxico. Sin embargo, Koskenniemi no los descarta en los siguientes casos:

- cuando existe mucha distancia entre el nivel léxico y de superficie, y/o cuando los cambios no son regulares.
- cuando la morfotáctica obliga a una dispersión excesiva de los subléxicos puede ser más interesante repetir ciertas entradas.

También se suelen duplicar las entradas léxicas para resolver casos de morfotáctica que no pueden ser resueltos de modo natural.

Los subléxicos se organizan en una **estructura *trie***, es decir como árboles de caracteres donde los arcos son caracteres léxicos, y los nodos terminales guardan la información morfológica y la clase de continuación de la entrada correspondiente. Esta estructura compacta la información y propicia un acceso incremental carácter a carácter.

III.2 Reglas de dos niveles.

III.2.1 Introducción.

La mayor aportación de Koskenniemi fueron las reglas de dos niveles para describir los cambios morfofonológicos por lo que algunos autores prefieren para el formalismo el nombre de fonología de dos niveles.

Las reglas de dos niveles controlan el emparejamiento entre la representación léxica y la de superficie. Las reglas se compilan en traductores de estados finitos (FST)[1] paralelos y un par de caracteres (léxico-superficie) será aceptado solamente si se acepta en todos los autómatas. No hay ningún tipo de representación intermedia entre los dos niveles mencionados y esta es la diferencia fundamental con la fonología generativa. Durante el análisis se buscan las representaciones léxicas correspondientes al nivel de superficie dado, y durante la generación lo contrario.

Como ya se ha mencionado anteriomente, Kaplan y Kay (1981) habían propuesto un modelo basado en reglas de reescritura secuenciales. Aunque las reglas también se compilaban en autómatas de estados finitos[2] hay que hacer notar las siguientes diferencias:

- El modelo de Kaplan y Kay sigue el modelo generativo por lo que teoricamente es bastante simple.
- En el tratamiento surgen representaciones intermedias por lo que la secuencia de las reglas es muy importante.
- Aunque es un modelo válido tanto para análisis como para generación el proceso de generación pude resultar no determinista.

---

1 La diferencia entre un traductor de estados finitos (finite state transducer - FST) y un autómata de estados finitos (finite state automaton - FSA) es que mientras en el autómata las etiquetas son simples en el traductor son pares de elementos (caracteres en nuestro caso). De todas formas, durante el texto, el término *autómata* también se utiliza como generalización de ambos.

2 La idea original es de Jhonson (1972) .

En la figura 3 se muestra la diferencia entre los modelos mencionados.

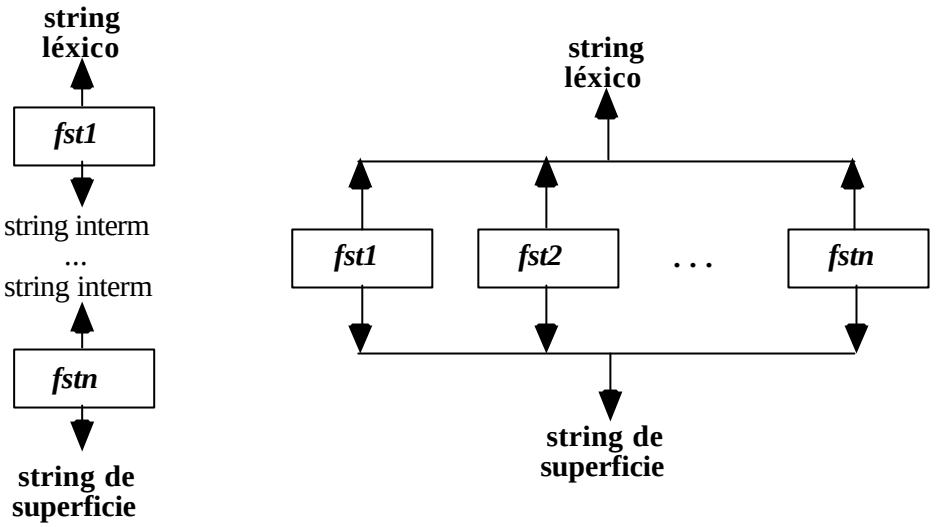

Fig. 3.- Comparación entre los modelos propuestos por Kaplan&Kay (izda.) y por Koskenniemi (dcha.)

III.2.2 Formato

La sintáxis de las reglas ha sufrido distintas modificaciones (Koskenniemi, 85; Ritchie *et al.*, 92; Karttunen, 93) sobre todo enfocadas a facilitar el trabajo a un compilador automático. Vamos a exponer aquí la última de las mencionadas —sin grandes diferencias con la original.

El formato es el siguiente:

```
cp  op  lc _ rc
```

Correspondencia (cp), es el par de caracteres (léxico:superficie) que controla la regla.

Operador (op), especifica que tipo de relación se establece entre la correspondencia y el contexto que se le asigna. Esta relación puede ser de cuatro tipos: *restricción de contexto* (=>), *coerción de superficie* (‹=), composición de ambas (<=>) y *establecimiento de prohibición* (/‹=). Las reglas de coerción de superficie son equivalentes a las de reescritura. Las de prohibición se proponen para facilitar el manejo de excepciones.

El significado de cada operador se expresa en la figura 4. El operador más utilizado es el compuesto, aunque cuando un cambio es opcional se suele utilizar el de restricción de contexto.

op	ejemplo	interpretación
=>	l:s => lc_rc	el carácter léxico *l* se convierte a nivel de superficie en *s* **si** el contexto es *lc_rc*.
<=	l:s <= lc_rc	en el contexto *lc_rc l* **siempre** se convierte en *s*.
<=>	l:s <=> lc_rc	el carácter *l* se convierte en *s* **solamente** en el contexto *lc_rc*.
/<=	l:s /<= lc_rc	*l* **nunca** se convierte en *s* en el contexto *lc_rc* .

Fig. 4.- Significado de los operadores

Contexto (lc_rc), delimita en que casos se produce el cambio especificado. El carácter _ separa el contexto a izquierda (*lc*) y el contexto a derecha (*rc*). En ambas partes del contexto se especifican series de pares de caracteres, y en esta sintáxis se pueden simplificar de la siguiente forma:

- cuando los dos caracteres del par son iguales basta con especificar el carácter
- si no se especifica un carácter de un nivel se supone cualquiera del alfabeto
- el símbolo # expresa frontera de palabra
- se pueden utilizar expresiones regulares para especificar contextos complejos
- el carácter % se utiliza como carácter de escape

No conviene sobrecargar los contextos y muchas veces es más interesante solo especificar uno de los dos niveles. Así si el cambio es de razón fonológica conviene especificar contexto a nivel de superficie y si es de carácter morfológico a nivel léxico.

III.2.3 Ejemplos

Se presentan tres reglas, la primera muy general para describir el cambio *g*/*gu* delante de *e* o *i* (no tenemos en cuenta los acentos), y las dos últimas para la generación de la *e* y el cambio *z*/*c* en los plurales.

```
0:u <=> g %+: _ [ e | i ]  ! seg+ir:seguir
0:e <=> Cons %+: _ s #     ! lugar+s:lugares
z:c <=> _ %+: :e s #       ! capaz+s:capaces
```

Como hemos visto son reglas generales que en algunos casos pueden tener excepciones. Un método para tratar excepciones consiste en el uso de diacríticos. Así, como algunas raíces terminadas en consonante no generan la *e* en el plural —p.ej. clip y récord— o se especifican los casos uno por uno como excepciones (utilizando la regla /<=) o si son numerosos se les marca con un diacrítico —p.ej. *$*, por lo que en el léxico aparecerán *clip$*, *record$* etc.— y se escribirá la regla siguiente:

```
0:e /<= $: %+: _ s #        ! clip$+s:clips
```

Si quisiéramos analizar textos con ciertas faltas de ortografía (p.ej. perdida de la *h* entre vocales) aumentando la cobertura aunque también la sobregeneración podríamos escribir la siguiente regla (cambio opcional):

```
h:0 => Vocal _ Vocal        ! ahondar:aondar
                            ! ahondar:ahondar
```

Para conseguir más ejemplos de distintas casuísticas es interesante consultar el libro de Antworth (1990) y utilizar la lista electrónica mencionada (*pc-parse-owner@sil.org*).

III.2.4 Compilación en autómatas

Aunque existen compiladores automáticos (Koskenniemi, 85), (Ritchie *et al.*, 92) (Karttunen & Beesley, 92) es interesante comprender el paso de las reglas a los autómatas ya que así se comprenderá mejor su funcionamiento. En el capitulo tres del libro de Antworth (1990) PC-KIMMO se explican profundamente.

III.3 Críticas y propuestas sobre el modelo.

Sobre el modelo de dos niveles se ha trabajado mucho desde su propuesta, aunque lo fundamental, las reglas morfofonológicas de dos niveles, permanece y es muy utilizado en morfología y fonología computacional. Además de las mejoras ya mencionadas, se han realizado un número importante de propuestas y críticas, y las vamos a agrupar en dos bloques: expresividad y morfotáctica.

III.3.1 Expresividad.

Aunque las críticas más importantes se han realizado sobre el tratamiento de la morfotáctica también en la morfofonología se han hecho propuestas de mejora. Black (Black *et al.*, 87) hace la siguiente crítica: al limitar la

correspondencia a un único carácter en cada nivel se producen interacciones entre diferentes reglas y esto crea confusión e inseguridad a la hora de escribir las reglas. Ante esto se propone un nuevo sistema de reglas (Pullman & Hepple, 93).

El grupo de Ritchie (Ritchie *et al.*, 92:181) ve la necesidad de reglas basadas en rasgos morfológicos necesitándose para ello un trabajo de investigación previo. Sobre esta base Carter (1995) trabajando en el proyecto *Core Language Engine* (Alshawi, 92) propone cambios en el formato de las reglas incluyendo los rasgos morfológicos con lo que se pueden eliminar los diacríticos. También Bear (1989), Trost (1991) y Karttunen (1994) hacen propuestas en este sentido.

Para el tratamiento de lenguas semíticas se pueden nombrar los trabajos de Kay (1987), Beesley (1990) y Kiraz (1994).

III.3.2 Morfotáctica: clases de continuación versus mecanismos de unificación.

El poder expresivo del sistema morfotáctico propuesto por Koskenniemi es muy pobre, ya que es totalmente lineal y, por lo tanto, lo único que se puede asociar a un morfema es el conjunto de morfemas que pueden encadenarse inmediatamente después de él. Aunque hay sistemas que mantienen este método morfotáctico —Karttunen (1983), PC-KIMMO (Antworth, 1989), Lexc (Karttunen, 1993)— ha sido bastante habitual proponer mejoras o cambiarlo totalmente —el mismo Antworth en la nueva versión de PC-KIMMO lo ha cambiado.

Una limitación importante que tiene el sistema lineal es que no es suficiente para expresar las *dependencias a distancia*, es decir, que un morfema no solo condicione al inmediatamente posterior sino también a otros. Por ejemplo, en inglés *en*, *joy* y *able* son morfemas correctos y aunque pueden encadenarse de forma correcta los tres seguidos *joyable* no es una forma correcta.

Este problema tiene difícil solución si no se cambia el mecanismo morfotáctico, y hay que recurrir a ciertos trucos. Para evitarlo algunos autores optan por modificar el mecanismo, proponiéndose ligeras modificaciones al mecanismo original (Agirre *et al.*, 92) o cambios drásticos. Entre estos últimos los más conocidas son los de Bear (1986), Trost (1990, 1994), y Alvey (Ritchie et al., 87; Ritchie et al., 92). En estas

tres propuestas se proponen mecanismos de unificación, pero mientras Bear se basa en PATR, el grupo de Ritchie ha elegido GPSG[1].

IV Complejidad del modelo de dos niveles y mejoras propuestas.

Koskenniemi, cuando propuso el modelo de dos niveles, defendió que una de las cualidades del mismo era su eficiencia. Este tema ha sido muy discutido pudiéndose seguir el razonamiento usando las siguientes referencias: (Barton, 86), (Barton *et al.*, 87), (Koskenniemi & Church, 88), (Sproat, 92: 3.5).

IV.1 Problemas de eficiencia.

La complejidad computacional básica del modelo de dos niveles viene del uso de reglas on-line. Debido a que un mismo carácter de un nivel se puede emparejar con varios del otro nivel y a que puede haber elipsis de caracteres, en muchos momentos se deben seguir varios caminos de análisis por lo que se utilizará *backtracking*. De modo intuitivo se puede concluir que cuantos más cambios se expresen por reglas y cuanto menos restricciones se especifiquen en el contexto menor será la eficiencia. Además si los caracteres léxicos pueden desaparecer a nivel de superficie la complejidad se puede disparar.

A pesar de que la complejidad estructural viene de las reglas, el léxico, al tener incluida la información morfotáctica, también es causa de perdida de eficiencia. La razón es la siguiente: una clase de continuación es un conjunto de subléxicos por lo que cuando se llega al final de un morfema se debe seguir por distintos puntos del sistema léxico aunque la mayoría de los caminos no tendrán éxito

IV.2 Mejoras propuestas.

Sin introducir cambios estructurales se puede mejorar la eficiencia del sistema siguiendo una serie de consejos "tácticos" —algunos de los cuales contradicen criterios de claridad y elegancia de la descripción:

---

[1] Estos formalismos son ampliamente utilizados en tratamiento sintáctico. Una buena referencia es la siguiente: Shieber S.M. *An introduction to unification-based approaches to grammar*. CSLI Lecture Notes 4. Chicago U. Press. 1986.

- Intentar especificar todas la restricciones que sean posibles en el contexto de la izquierda, con lo que se conseguirá descartar posibilidades de análisis antes.
- Evitar reglas que describan la perdida de los caracteres léxicos más comunes, aunque para ello se utilicen más diacríticos.
- Promocionar clases de continuación del mínimo posible de subléxicos, aunque para ello, a veces, se almacenen alomorfos.

Dentro de las propuestas estratégicas se pueden destacar dos: la fusión de léxicos propuesta por Barton y los traductores léxicos propuestos por Karttunen.

IV.3 Fusión de léxicos.

La idea básica es muy sencilla: se juntan todos los subléxicos en uno solo —o lo que es más lógico, en tres: prefijos, lemas y sufijos— con lo que se aumenta la eficiencia y se compacta el sistema léxico.

Sin embargo, surge un problema: el del control morfotáctico. Si no se hacen otros cambios, la información morfotáctica se ubicará en las hojas del árbol de la estructura *trie*, con lo que durante cierto tiempo se pueden estar recorriendo morfemas que serán luego invalidados por la morfotáctica. Esto se agrava si el tratamiento morfotáctico se hace posterior e independiente al morfofonológico.

IV.4 Traductores léxicos.

La propuesta de Karttunen de los llamados traductores léxicos (Karttunen *et al.*, 92), (Karttunen, 93), (Karttunen, 94) va mucho más allá, y además de una mejora espectacular de la eficiencia introduce cambios y mejoras de fondo: por un lado se pueden evitar las representaciones arbitrarias utilizando las correspondientes formas canónicas acompañadas de rasgos morfológicos; y por otro se pueden componer una serie de sistemas de reglas permitiendo mayor expresividad, claridad y modularidad en la descripción[1].

En los traductores léxicos se pueden emparejar en el léxico formas flexionadas con sus correspondientes canónicas —p.ej. *better* y

[1] Carter (1995) dentro del proyecto *Core Language Engine* hace una propuesta similar en algunos aspectos.

good+COMP—, aunque con ello se aumente la distancia entre representaciones léxicas y de superficie; ya que para hacer frente a este aumento de distancia se permiten niveles intermedios y se generan automáticamente autómatas para el léxico. Todo esto se basa en gran parte en las ideas expuestas en diversos artículos por Kaplan y Kay (Kaplan, 88) (Kaplan & Kay, 94).

Los resultados son espectaculares. Por ejemplo para el francés se dan los siguientes resultados: 50 K-estados y 100 K-arcos almacenados en menos de 1 Mbyte. Resultados similares han sido obtenidos para otras lenguas. La velocidad es de varios miles de análisis por segundo cientos de veces superior a las que se conseguían anteriormente.

A nivel de especificación podemos decir que los traductores léxicos son una mejora del modelo de dos niveles aunque mucho más potente, general y flexible. Los artículos de Chanod (1994), Kwon y Karttunen (1994) y el nuestro (Alegria *et al.*, 95) notifican distintas aplicaciones de este modelo.

Bibliografía

Agirre E., Alegria I., Arregi X., Artola X., Díaz de Ilarraza A., Sarasola K., Urkia M. Aplicación de la morfología de dos niveles al euskara, *SEPLN*, vol. 8, 87-102. 1989.

Agirre E., Alegria I., Arregi X., Artola X., Diaz de Illarraza A,. Maritxalar M., Sarasola K., Urkia M. XUXEN: A spelling checker/corrector for Basque based on Two-Level morphology, *Proc.of the Third ANLP*, 119-125. 1992.

Agirre E., Arregi X., Arriola J.M., Artola X., Diaz de Illarraza A., Insausti J.M., Sarasola K. Different issues in the design of a general-purpose Lexical Database for Basque. *First workshop on application of Natural Language to Data Bases, NLDB'95*. 1995.

Alegria I. *Euskal Morfologiaren tratamendu automatikorako tresnak*. Tesia. UPV/EHU. 1995.

Alegria I., Artola X., Sarasola K. Improving a robust morphological analyzer using lexical transducers. RANLP, Bulgaria. 1995.

Allen J., Hunnicutt M., and Klatt D. *From text to speech: the MITalk System*. Cambridge University Press. 1987.

Alshawi, H (ed.) *The Core Language Engine*. MIT Press. 1992.

Anick P. and Artemieff S. A high-level morphological description language exploiting inflectional paradigms, *Proc. of COLING´92*, 67-73. 1992.

Antworth E.L. PC-KIMMO: A two-level processor for morphological analysis. *Occasional Publications in Academic Computing, No. 16*, Dallas, Texas. 1990.

Barton G.E. Computational Complexity in two-level Morphology, *ACL Proceedings, 24th Annual Meeting*. 1986.

Barton G., Berwick R., and Ristad E. *Computational Complexity and Natural Language*. MIT Press. 1987.

Bear J. A morphological recogniser with syntactic and phonological rules. *Proc. of COLING´86*, 272-276. 1986.

Bear J. Morphology with two-level rules and negative rule features. *Proc. of 12th. COLING*, 28-31. 1988.

Beesley K. Finite-state descriptions of arabic morphology. *Proc. of the 2nd Conference on bilingual computing in Arabic and English*. 1990.

Black A., Ritchie G., Pulman S and Russel G. Formalisms for morfographemic description. *Proc. of 3th Conference of the EACL*, 11-16.1987.

Black A., van de Plassche,J., Williams B. Analysis of Unkown Words through Morphological Descomposition. *Proc. of 5th Conference of the EACL*, vol. 1, 101-106.1991.

Byrd R. Klavans J., Aronoff M., Anshen F. Computer methods for morphological analysis, *Proc. of 24th ACL*. 1986.

Cahill L.J. Syllable based morphology. *Proc. of 13th COLING*, vol.3, 48-53. 1990.

Calder J. Paradigmatic morphology. *Proc. of the 4th EACL*, 58-65. 1988.

Carter D. Rapid development of morphological descriptions for full language processing system. *Proc. of EACL´95*. 1995.

Chanod J.P. *Finite-state composition of french verb morphology*. Xerox MLTT-005. 1994.

Clemenceau D. and Roche E. Enhancing a large scale dictionary with a two-level system. *Proc. of EACL'93*, p.465. 1993.

Dalrymple M., Kaplan R., Karttunen L., Kay M., Kornai A., Koskenniemi K., Shaio S., Wescoat M. *DKIMMO/TWOL: a development environment for morphological analysis.* Xerox Palo Alto. 1987.

Domenig M. Lexeme-based morphology: a computationally expensive aproach intended for a server-architecture. *Proc. of 13th COLING*, vol 2, 77-82. 1990

GETA. Le point sur Ariane-78 debut 1982, 1 , partie 1: le logiciel 28-75. 1982.

Goñi J.M and Gonzalez J.C. A framework for lexical representation. *AI-95, Language Engineering*, 243-252. Montpellier, 1995

Hajic J. Morphotactics by attribute grammar, *Proc. of 4th EACL*. 1989.

Hopcroft J. and Ullman J. *Introduction to Automata Theory, Languages and Computatuion.* Addison-Wesley. 1979.

Jhonson C. *Formal aspects of phonological description.* Mouton. 1972.

Kaplan R. M. and M. Kay. Phonological rules and finite-state transducers. *Paper read at the annual meeting of the Linguistic Society of America in New York City.* 1981.

Kaplan R. M. Regular models of phonological rule systems. Alvey Workshop on parsing an pattern recognition. Oxford University. 1988.

Kaplan R. M. and M. Kay. Regular models of phonological rule systems. *Computational Linguistic s*, vol.20(3), 331-380. 1994.

Karlsson F. SWETWOL: A comprehensive morphological analyser for Swedish. *Nordic Journal of Linguistics*, 15, 1-45. 1992.

Karlsson F., Voutilainen A., Heikkila J,. Anttila A. *Constraint Grammar: A Language-independent System for Parsing Unrestricted Text*.. Mouton de Gruyter. 1995.

Karp D., Schabes Y., Zaidel M., Egedi D. A freely available wide coverage morphological analyzer for English. *Proc. of COLING´92*, Nantes, 950-954. 1992.

Karttunen L. KIMMO : A two-level Morphological Analyzer, *Texas Linguistic Forum*, Vol 22, 165-186. 1983.

Karttunen L. and Wittenburg K. A two-level morphological analysis of English, *Texas Linguistic Forum*, Vol 22, 217-228. 1983.

Karttunen L. Finite-State Constraints, *Proc. of Current Issues in Computational Linguistics*, 23-40. Malaysia, 1991.

Karttunen L., Kaplan R.M., Zaenen A. Two-level morphology with composition. *Proc. of COLING´92*. 1992.

Karttunen L. and Beesley K.R. *Two-Level Rule Compiler*. Xerox ISTL-NLTT-1992-2. 1992.

Karttunen L. *Finite-State Lexicon Compiler*. Xerox ISTL-NLTT-1993-04-02. 1993.

Karttunen L. Constructing Lexical Transducers, *Proc. of COLING´94*, 406-411. 1994.

Karttunen L., Yampol T. *Interactive Finite-State Calculus*. Xerox. 1994.

Kay M. Morphological Analysis, *Proc. of the Int. Conference on Computational Linguistics*. Pisa, 1973.

Kay M. Morphological and syntactic analysis. *Linguistics Structures Processing*. Noth Holland. 1977.

Kay M. Nonconcatenative finite-state morphology. *Proc. of 3th Conference of the EACL*, 2-10. 1987.

Kay M. and Kaplan R. Word Recognition. Manuscript, Xerox. 1983.

Kim D., Lee S., Choi K., Kim G. A two-level morphological analysis of Korean, *Proc. of COLING´94*. 1994.

Kiraz G.A. Multi-tape two-level morphology: a case study in semitic non-linear morphology, *Proc. of COLING´94*, 180-186. 1994.

Koskenniemi K. Two-level Morphology: A general Computational Model for Word-Form Recognition and Production. Ph.D. thesis, University of Helsinki. *Publications nº 11*. 1983.

Koskenniemi K. Compilation of Automata from Morphological Two-level Rules. University of Helsinki, *Publication nº 15*. 1985.

Koskenniemi K. and Church K.W. Complexity, two-level morphology and Finnish. *Proc. of 12th COLING*, 335-340. 1988.

Kwon H-C, Karttunen L. Incremental construction of a lexical transducer for Korean, *Proc. of COLING´94*, 1262-1266. 1994.

Maruyama H. Backtracking-free dictionary access method for Japanese morphological analysis, *Proc. of COLING´94*, 208-213. 1994

Martí M.A. Un sistema de análisis morfológico por ordenador. *SEPLN*, vol. 4, 105-110. 1987.

Matthews P.H. *Morpholgy: An introduction to the theory of word-structure*. Cambridge textbooks in linguistics. Cambridge University Press. 1974.

Meya M. Análisis morfológico como ayuda ala recuperación de información. *SEPLN*, vol. 4, 91-103. 1987.

Moreno Sandoval A. *Un modelo computacional basado en unificación para el análisis y generación de la morfología del español*. Tesis Doctoral. Universidad Autonoma de Madrid. 1991.

Moreno A. and Goñi J.M. GRAMPAL: A morphological processor for Spanish inplemented in Prolog. GULP-PRODE95 (Joint Conference on Declarative Programming).1995.

Nobesawa S., Tsutsumi J., Nitta T., Ono K., Jian S., Nakanishi M. Segmenting into morphemes using statistic information betwen words, *Proc. of COLING´94*, 227-232. 1994.

Oflazer K. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, vol.9, No. 2, 137-148. 1994.

Pulman S. and Hepple M. A feature-based formalism for two-level morphology: a description and inplementation. *Computer Speech and Language*, 7. 1993.

Ritchie G., S.G. Pulman, A.W.Black and G.J. Russell. A Computational Framework for Lexical Description, *Computational Linguistics*, vol. 13, ns 3-4. 1987.

Ritchie G. Languages generated by two-level morphological rules, Computational Liguistics, vol 18, No.1. 1992.

Ritchie G., A.W.Black, G.J. Russell and S.G. Pulman. *Computational Morphology*.The MIT Press. 1992.

Schiller A. Steffens P. A lexicon for a German two-level morphology, *Euralex* 1990 (Benalmádena). 1990.

de Smedt, K. Using Object-Oriented Knowledge-Representation Techniques in Morphology and Syntax Programming. *EACI-84*, 181-184. 1984.

Sproat R. *Morphology and Computation*. The MIT Press. 1992.

Tapanainen P. and Voutilainen A. Ambiguity resolution in a reductionistic parser. *Proc. of Sixth Conference of the EACL,* Utrech. 1993.

Trost H. The application of two-level morphology to non-concatenative German morphology, *Proc. of COLING-90,* Helsinki, vol.2 371-376. 1990.

Trost H. Recognition and generation of word forms for natural language understanding systems: integrating two-level morphology and feature unification, Applied Artificial Intelligence, 5(4), 411-458. 1991.

Trost H. Coping with derivation in a morphological component, *Proc. of the 6th Conference of the EACL,* 368-376. 1993.

Trost H. Morphology with a null-interface. *Proc. of COLING´94,* 141-147. 1994.

Tzoukermann E. and Liberman M. A finite-state morphological processor for Spanish. *Proc. of COLING-90,* Helsinki, vol.3, 277-281. 1990.

Winograd T. *Language as a cognitive process. Vol.1: Syntax*, 544-549. Addison-Wesley, 1983.