DEMOSTRACIONES

SEISD: UN ENTORNO PARA LA EXTRACCIÓN DE INFORMACIÓN SEMÁNTICA DEL DICCIONARIO VOX

Alicia Ageno, Francesc Ribas, German Rigau

Universitat Politècnica de Catalunya, Departament de LSI, Barcelona

Abstract

Presentamos con esta demostración un entorno interactivo para la obtención y representación de información semántica de una Base de Datos Léxica (BDL) construida a partir de diccionarios en soporte magnético. El entorno posibilita el tratamiento de la información léxica, realizéndose de forma semiautomática y dotando al lexicógrafo de las ayudas precisas para completar, de forma interactiva, el proceso de extracción y representación. La presente demostración se organiza del siguiente modo: tras una breve exposición del proyecto **ACQUILEX**(*), marco en el que se desarrolla nuestro sistema, expondremos el sistema SEISD de forma general. Luego nos centraremos en la extracción de taxonomías, a partir del diccionario Vox, y su representación en la Base de Conocimiento Léxico (BCL). Por último, estableceremos una serie de conclusiones y perspectivas de futuro.

Los diccionarios automatizados constituyen una fuente de adquisición de conocimiento léxico y conceptual que, potencialmente, permite abordar algunos aspectos especialmente costosos de la construcción de una base de conocimiento para un sistema de procesamiento del lenguaje natural (PLN) de forma rápida y competitiva. Se trata de un campo relativamente poco explorado del área de la adquisición del conocimiento, debido a la dificultad que supone el tratamiento complejo de grandes volúmenes de información no siempre sistemática y a las limitaciones de las teorías lingüísticas que abordan el tema del léxico.

El objetivo básico del proyecto **ACQUILEX** es el desarrollo de técnicas y métodos que permitan la utilización de diccionarios en soporte magnético (MRD, Machine Readable Dictionaries) para la construcción de componentes léxicos para sistemas PLN. Nos centramos, por tanto, en el desarrollo de un prototipo de BDL y de BCL multilingües para un subconjunto manejable, pero significativo, del vocabulario y en el desarrollo de técnicas para la extracción semiautomática de información léxica del diccionario.

(*) **ACQUILEX** *es un proyecto integrado, en el que participan el Instituto de Lingüística Computacional de Pisa y las universidades de Amsterdam, Cambridge, Dublin y Politécnica de Catalunya . El proyecto esta financiado por la C.E.E. a través del programa ESPRIT (Acción BRA 3030).*

DEMOSTRACIÓN DE LAS GRAMÁTICAS DEL ESPAÑOL Y DE LA TRADUCCIÓN AL ALEMÁN DESARROLLADAS EN EUROTRA (1987-1992)

Toni Badia, Marta Carulla, David Soler — Eurotra-Barcelona

Rita Nuebel — Eurotra-Saarbrücken

1. Introducción

Esta demostración conjuntamente con la comunicación presenta los resultados del proyecto Eurotra. Permite mostrar a los asistentes la ejecución del proceso de análisis, traducción y generación. La demostración tendrá dos partes. En la primera se traducirá una muestra del tipo de texto que puede tratar el sistema implementado, con un léxico proveniente del campo de las telecomunicaciones. En la segunda se ofrecerá a los asistentes la posibilidad de formar sus propias frases (a partir de un léxico restringido y teniendo en cuenta la cobertura de nuestra gramática). Estas frases serán analizadas por el sistema ante los asistentes, de manera que se podrá inspeccionar tanto las reglas usadas como las representaciones intermedias del proceso de análisis.

A continuación especificamos la cobertura de la gramática española. Conviene recalcar que ésta es exactamente igual para análisis que para generación, dado que nuestra gramática es básicamente reversible. La reversibilidad consiste en que las gramáticas particulares de cada nivel de descripción son las mismas tanto para el proceso de análisis como para el de generación. En consecuencia las operaciones características de cada uno de los procesos (eliminación o creación de nudos, modificación de estructura, etc.) se han relegado a los traductores que relacionan los niveles. Por lo tanto, no distinguiremos entre estos dos procesos al describir la cobertura.

2. Cobertura de la gramática española desarrollada en Eurotra

En el módulo morfológico, se trata totalmente la flexión de los verbos, nombres y adjetivos, incluyendo los paradigmas irregulares correspondientes. La información resultante de la flexión (tiempo, persona, número, género, etc.) se convierte en rasgos, de manera que en los niveles posteriores sólo aparece la forma canónica de las palabras (el infinitivo, para los verbos; el singular, para los nombres; etc.). En este mismo módulo se trata la presencia de clíticos en las distintas formas verbales, así como la aparición de "se" con los verbos pronominales.

En los módulos restantes, se tratan los siguientes fenómenos:

- la oración:
  - oraciones principales con verbos en cualquier tiempo, tanto en voz activa como en pasiva,
  - oraciones subordinadas finitas, tanto adverbiales como nominales,
  - oraciones subordinadas en infinitivo, gerundio y participio.
- las categorías sintagmáticas
- las categorías léxicas
- los complementos oracionales de nombres y adjetivos
- la coordinación de oraciones y de sintagmas nominales, adjetivales y preposicionales (la coordinación se basa en las categorías sintácticas, por lo cual no se trata la coordinación de categorías sintácticas diferentes)

- el tiempo y el aspecto verbales, tanto en las oraciones principales como en las subordinadas

- la modalidad

- los fenómenos qué: las oraciones de relativo y las interrogativas, con uno o más niveles de profundidad (no obstante, la extracción múltiple no ha sido tratada)

- los fenómenos de elevación y de control, con la coindexación correspondiente de los nudos vacíos

- la referencia pronominal sólo en el caso de los sujetos vacíos de oraciones adverbiales finitas (de momento, no hay tratamiento para los pronombres reflexivos o recíprocos)

- las construcciones no oracionales cuya estructura no es fundamentalmente diferente de la de los constituyentes de las oraciones

- la comparación adjetiva

- la negación

- la determinación nominal

- algunas de las construcciones de verbo soporte

- los modificadores temporales no oracionales

- algunos de los transconstruccionales

3. Conclusión

Con esta demostración se brinda la oportunidad de conocer un poco cómo se han tratado estos fenómenos lingüísticos en nuestra gramática, examinando las representaciones en los niveles intermedios y las reglas gramaticales, léxicas, de traducción entre niveles de una lengua y de transferencia que se han aplicado.

RECUPERACIÓN DE INFORMACIÓN EN DICCIONARIOS

Octavio Santana Suarez
Margarita Díaz Roca
Antonio Ballester Monzón
Francisco J. Santana Pérez

0) Introducción

La localización de palabras en texto libre constituye una parte fundamental de un problema práctico de gran importancia: la organización y utilización de información procedente de fuentes heterogéneas.

Existen dos aproximaciones a este problema. En la primera, se efectúa la búsqueda directamente sobre el documento sin ningún tipo de preparación previa; si se trata con actas de longitud mediana o las consultas son algo complejas, esta manera de proceder puede resultar muy costosa en cuanto a tiempo de respuesta. Desde la segunda perspectiva, se someten anteriormente los escritos a un tratamiento; la intención es generar un índice que haga viable los accesos posteriores al ejemplar cuando sea necesario.

Siguiendo este último enfoque, se utiliza como índice la estructura DITE [SD87,89,DS90], construida a partir del conjunto de las palabras diferentes que se obtienen del documento al descartar las cadenas consideradas *vacías*. El corpus que se va a indizar para llevar a cabo las localizaciones textuales es un Diccionario de Medicina [JV87].

De las 993.753 palabras de que consta el Diccionario de Medicina 424.689 se consideran vacías _589 es la cardinalidad de este conjunto_ y de las 569.064 restantes, solamente 53.334 son distintas. En la figura 1, se muestra la distribución de frecuencias por longitudes de este conjunto de palabras diferentes.

Se modifican los nodos-SIT[2] añadiendo un puntero a cada sinónimo-DIT, de forma que señale a una lista de posiciones en las que aparece esa palabra en el diccionario, como se observa en la figura 2.

Sobre la estructura DITE así construida se permiten los siguientes tipos de búsquedas:

- Exacta
- Más similares
- Con operadores booleanos: *o*, *y*, *y-no*.
- Máscaras
- Truncamientos: a la derecha, a la izquierda, a ambos lados.
- Cercanía
- Antecedencia
- Párrafos
- Sentencias
- Frases
- Compleja

[2] Véase el apéndice.

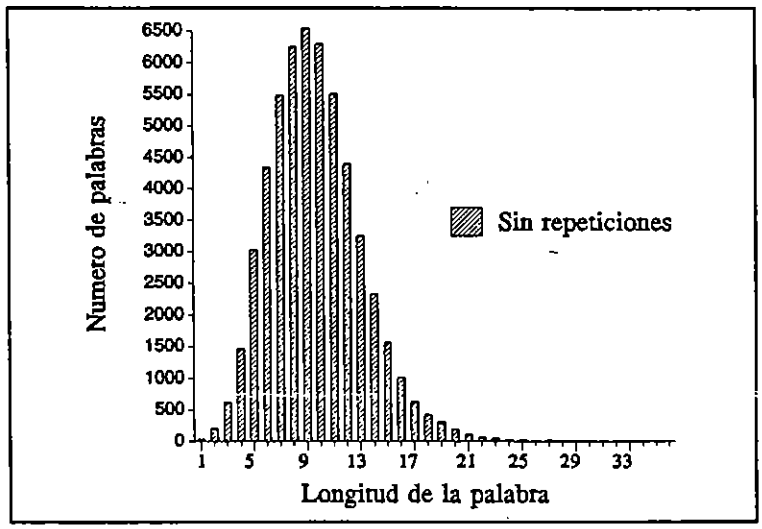

Figura 1

Un Diccionario está dividido de forma natural en *bloques de texto* donde cada uno se compone de una palabra y su significado. La lista de posiciones asociada a cada sinónimo-DIT indica los distintos bloques en los que está presente. Si una palabra se repite dentro de un mismo bloque, su lista de posiciones hace referencia una sola vez a éste bloque.

Las búsquedas de: **cercanía, antecedencia, párrafos, sentencias y frases,** así como las complejas que incluyan a alguna de las citadas, requieren el tratamiento de cada bloque candidato al conjunto respuesta con el fin de verificar las restricciones especificadas en la petición en cuanto a la situación de las palabras.

Cuando se solicita una búsqueda del tipo: **más similares, máscaras** o **truncamientos,** que tienen respuesta múltiple se muestran en la pantalla las palabras que satisfacen la petición. Se permite que el usuario seleccione un subconjunto de ellas para ojear los bloques asociados.**1) Búsqueda Exacta**

La búsqueda más elemental consiste en determinar todas las localizaciones de una palabra en el documento. La petición se expresa simplemente indicando cuál es la que se solicita. Por ejemplo: *sida*, aparece en ocho bloques.

2) Búsqueda de las palabras más similares

Proporcionada una métrica en el espacio de las palabras, DD [LE66,UK83,85,LV85,86], las más similares [GG86,SD87,DS90] a una cadena dada (esta puede pertenecer o no al diccionario) son todas las que se encuentran en el diccionario a distancia mínima de ella.

El formato de la petición es: *+cadena*. Por ejemplo, al solicitar *+rida* se obtienen: *ria, vida, rica, risa, sida, oida, brida.*

3) Búsquedas con operadores booleanos

Cuando se quieren realizar búsquedas a partir de varias palabras se pueden utilizar como conectores los operadores lógicos: *o, y, y-no.*

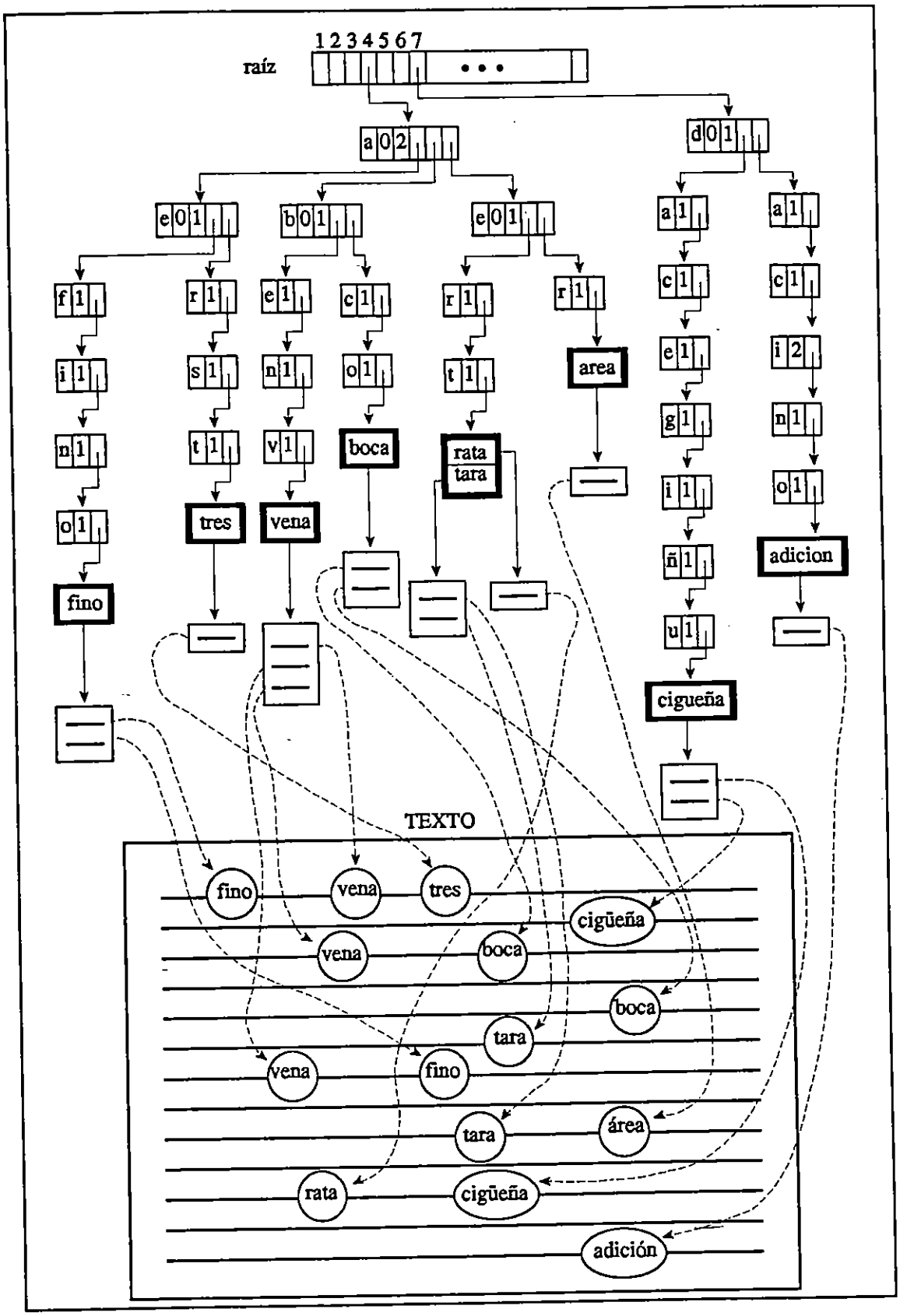
raíz
1 2 3 4 5 6 7
a 0 2
d 0 1
e 0 1
b 0 1
e 0 1
a 1
a 1
f 1
r 1
e 1
c 1
r 1
r 1
c 1
c 1
i 1
s 1
n 1
o 1
t 1
area
e 1
i 2
n 1
n 1
t 1
v 1
boca
rata
tara
g 1
o 1
o 1
tres
vena
i 1
adicion
fino
ñ 1
u 1
cigueña
TEXTO
fino
vena
tres
cigüeña
vena
boca
boca
tara
vena
fino
tara
área
rata
cigüeña
adición

Figura 2

3.1) Búsqueda disyuntiva

Una búsqueda disyuntiva proporciona todos los bloques en los que aparece al menos una de las palabras especificadas en la petición. Se solicita de la siguiente forma: *cadena1 o cadena2*. Por ejemplo: *cáncer o tumor.*

3.2) Búsqueda conjuntiva

La respuesta de una petición conjuntiva está formada por aquellos bloques que contienen a todas las palabras especificadas. Esta petición se formula como: *cadena1 y cadena2*. Por ejemplo: *cáncer y tumor.*

3.3) Búsqueda con operador *y-no*

En una búsqueda de una petición de la forma *cadena1* ***y-no*** *cadena2* se obtienen aquellos bloques en los que se encuentra la palabra situada a la izquierda del operador y que a su vez no contienen a la especificada a la derecha. Por ejemplo: *tos* ***y-no*** *fiebre.*

4) Búsqueda con máscaras

Si se desea hacer una búsqueda de una palabra en la que se desconocen caracteres en determinadas posiciones, se puede formular una petición en la que se sustituyan los caracteres desconocidos por *comodines* (*).

En una búsqueda con máscaras, se recuperan todos los bloques en los que se encuentra una palabra de longitud igual a la especificada y que difiera solamente en los caracteres indicados por los asteriscos. Por ejemplo para *t*m*r* se recuperan los bloques en los que aparecen palabras como *tumor, temor, tomar, timor.*

5) Búsquedas con truncamientos

Las búsquedas con truncamientos a la derecha, a la izquierda y a ambos lados permiten recuperar bloques que contienen palabras que empiezan, terminan o presentan alguna coincidencia central con la petición.

Cada tipo de truncamiento se expresa de la siguiente forma:

a la derecha:	*cadena!*
a la izquierda:	*!cadena*
a ambos lados:	*!cadena!*

Por ejemplo:

a la derecha: *tos!*,	se obtienen: *tos, tose, tosa, toser, tosco, tosca, toscos, toscas, tossach, tostada.*
a la izquierda: *!tipo*,	se recuperan: *subtipo, ojotipo, idiotipo, optotipo, fenotipo, genotipo, serotipo, fagotipo, prototipo, cariotipo, inmunotipo, somatotipo.*
a ambos lados: *!cubo!*,	se localizan: *cubo, cuboide, cuboides, cuboideo, cuboidal, cuboidea, cuboideum, cuboideos, cuboideas, calcaneocuboidea, calcaneocuboideo, calcaneocuboideas.*

6) Búsquedas de cercanía y antecedencia

Cercanía

La búsqueda de cercanía se caracteriza por limitar el número máximo de palabras que pueden existir entre las dos especificadas en la petición. Las peticiones son del tipo: *cadena1* ***c/n*** *cadena2*. Por ejemplo: *fiebre* ***c/5*** *aguda*, donde si se encuentra *fiebre*, *aguda* ha de estar como mucho en la quinta palabra contando a partir de la siguiente a *fiebre*, o si se localiza primero *aguda*, *fiebre* no ha de estar más allá de la quinta palabra después.

Antecedencia

La antecedencia entre palabras presenta una restricción más con respecto a la búsqueda anterior ya que establece el orden de aparición de las mismas. Las peticiones son del tipo: *cadena1* ***a/n*** *cadena2*. Por ejemplo: *nervio* ***a/3*** *facial*, lo que indica que primero se ha de encontrar *nervio* y como mucho la tercera palabra, después ha de ser *facial*.

7) Búsquedas en párrafos y sentencias

Párrafos

La búsqueda en párrafos es una alternativa a la de cercanía entre palabras en la que se requiere que las palabras especificadas se encuentren en el mismo párrafo. Se solicita de la siguiente forma: *cadena1* ***p/*** *cadena2*. Por ejemplo: *pies* ***p/*** *manos*.

Sentencias

Es análoga a la búsqueda en párrafos pero restringida a una sentencia. La petición se expresa: *cadena1* ***s/*** *cadena2*. Por ejemplo: *pies* ***s/*** *manos*.

8) Búsqueda de frases

La búsqueda de frases se puede ver como una búsqueda de cercanía entre palabras en las que la máxima separación entre las especificadas en la petición es uno, es decir, *c/1*. La frase se denota entre comillas. Por ejemplo: *«de pies y manos»*.

9) Búsqueda compleja

Cualquier combinación de los anteriores tipos de búsqueda en una expresión utilizando los conectores lógicos (*y*, *o*, *y-no*) es a lo que se denomina una petición compleja. Por ejemplo: *s**a* ***y*** *(can! o tumor)* ***y-no*** *sana*.

Cuando interviene algún tipo de búsqueda con respuesta múltiple se realiza una fusión de las listas de posiciones asociadas que haya seleccionado el usuario.

El orden de prioridad de los conectores es de izquierda a derecha. Si se desea alterar esta prioridad, se deben utilizar paréntesis y en el caso de que existan paréntesis anidados, los más interiores son los que se resolverán en primer lugar.

La obtención de la respuesta de una búsqueda compleja se realiza de forma recursiva para cada par de abre-paréntesis y cierra-paréntesis. A partir del abre-paréntesis, se actúa de izquierda a derecha ejecutando cada búsqueda no-booleana y resolviendo los operadores lógicos una vez que se alcanza el cierra-paréntesis. Por defecto, se considera que la petición compleja está encerrada entre paréntesis.

Búsqueda en la que intervienen peticiones anteriores

Tras echar una ojeada a las peticiones previas, es posible modificar una ya existente para confeccionar otra más compleja. El formato *@n* indica que la n-ésima consulta realizada forma parte de la nueva petición. Supóngase que se desea añadir a la 5ª búsqueda, cuya expresión de contenido es: *cáncer y tumor*, un nuevo término: *y-no pulmón*, la nueva petición se escribe de la siguiente forma: *@5 y-no pulmón*.

También se pueden hacer combinaciones de peticiones previas sin añadir ningún término de búsqueda nuevo, utilizando los operadores booleanos. Por ejemplo: *@2 y (@4 o @6)*, combina los resultados de las consultas 2, 4 y 6.

9.1) Analizador sintáctico de la petición

El analizador sintáctico cumple un doble cometido: diferenciar las componentes _cada tipo de subpetición_ y los operadores lógicos que las conectan y, a la vez, determinar la correctitud sintáctica de la petición.

Se consideran erróneas las siguientes situaciones:

- Si alguna petición no-booleana presenta un formato incorrecto, bien porque se especifica una búsqueda no permitida, por ejemplo: *t*m!*, o bien porque le falta alguna condición, por ejemplo: si se detecta la presencia de *'/'* precedida de *a* o *c* y no le sigue un valor entero o se especifica una frase y falta cerrar las comillas, etc.

- Si se utilizan conectores no permitidos.

- Si después de un operador lógico no existe una búsqueda no-booleana o un abre-paréntesis.

- Si se especifica la búsqueda exacta de una palabra considerada como vacía.

- Si existe un abre-paréntesis para el que no se encuentra el correspondiente cierra-paréntesis.

En caso de que se detecte un error tal, se informará al usuario de ello y se señalará en la pantalla la posición en la que se ha encontrado dicho fallo.

Para analizar la petición _formada por una cadena de 100 caracteres de longitud máxima_ se recorre de izquierda a derecha identificando las diversas búsquedas no-booleanas, operadores lógicos y paréntesis, comprobándose que no ocurre ninguna de las situaciones indicadas anteriormente.

Una vez que se ha verificado que la petición es gramaticalmente correcta e identificado sus componentes, se pasa a la fase de ejecución de la búsqueda.

Bibliografía:

[DS90] DIAZ, M.; SANTANA, O.; RODRIGUEZ, J.C.:»Búsqueda de las Cadenas Mas Similares: Esquema Decreciente con Radio de Búsqueda Ascendente, Esquema Creciente». XVI Conferencia Latinoamericana de Informática (1990), Vol I, 90/97.

[GG86] GALIL, Z.; GIANCARLO, R.:»Improved String Matching with k Mismatches», SIGACT News, 17, 4, 52/54, (1986).

[JV87] JOVEN, J.; VILLABONA, C.; JULIA, G.; GONZALEZ-HUIX, F.:»Diccionario de Medicina». Tercera edición (1987). Editorial MARIN, S.A.

[LE66] LEVENSHTEIN, V. I.: «Binary Codes Capable of Correcting, Insertions and Reversals». Soviet Phys. Dokl. 10 (1966), 707/710.

[LV85a] LANDAU, G. M.; VISHKIN, U.: «Efficient String Matching in the Presence of Errors». Proc. 26th IEEE FOSC, 126/136, (1985).

[LV85b] LANDAU, G. M.; VISHKIN, U.: «Efficient String Matching with k Differences». TR_36/85, Departament of Computer Science, Tel Aviv Univ., Submitted for Journal Publication, (1985).

[LV86a] LANDAU, G. M.; VISHKIN, U.: «Efficient String Matching with k Mismatches». Theoretical Computer Science, 43, 239/249, (1986).

[LV86b] LANDAU, G. M.; VISHKIN, U.; NUSSINOV, R.: «An Efficient String Matching Algorithm with k Differences for Nucleotide and Amino Acid Sequences». Nucleic Acid Research 14 (1), 31/46, (1986).

[SD87] SANTANA, O.; DIAZ, M.; MAYOR, O.; REYES, J.: «Esquemas y Estructura para la Búsqueda de las Palabras Más Similares a una Dada». XIII Conferencia Latinoamericana de Informatica (1987), Vol. II, 1169/1189.

[SD89] SANTANA, O.; DIAZ, M.; DUQUE, J.D.; RODRIGUEZ, G.:»Estructuración de las Componentes de la Distancia Invariante Trasposicional, DIT, con Compartición de la Zona No_Discriminante en la Búsqueda de las Cadenas Mas Similares». XV Conferencia Latinoamericana de Informática (1989), Vol. II, 335/341.

[SP89] SANTANA. O.; PEREZ, J.; RODRIGUEZ, J. C.: «Increasing Radius Search Schemes for the Most Similar Strings on the Burkhard_Keller Tree». International Workshop on Computer Aided Systems Theory, EUROCAST'89, (1989).

[SR90] SANTANA, O.; RODRIGUEZ, J. C.; DIAZ, M.: «Búsqueda de las Cadenas Más Similares: Incidencia de la Subsecuencia Común Más Larga en los Esquemas Decreciente y Creciente». XVI Conferencia Latinoamericana de Informática (1990), Vol I, 98/104.

[UK83] UKKONEN,E.:»On Approximate String Matching». Proc. Int. Conf. Found. Comp. Theor., Lecture Notes in Computer Science 158, Springer-Verlag, (1983), 487/495.

[UK85] UKKONEN, E.:»Finding Approximate Pattern in Strings». J. of Algorithms, 6, 132/137, (1985).

APÉNDICE

DD -. Se llama **distancia direccional** de edición, DD(X,Y), entre las cadenas *X* e *Y*, al mínimo costo de todas las secuencias de edición que transformen *X* en *Y*.

DIT -. La **distancia invariante trasposicional** entre dos cadenas *X* e *Y* se calcula como la suma acumulada del valor absoluto de las diferencias de frecuencias de los caracteres de *X* e *Y* incrementada por la diferencia de longitudes.

DITE -. El árbol tiene un nodo raíz que discrimina por longitudes de cadenas. El encadenamiento asociado a cada longitud señala a una parte-árbol constituida exclusivamente por nodos discriminantes que contienen un carácter discriminante, _, y varios enlaces e_i, donde cada e_i es un enlace que apunta a un subconjunto de cadenas D^i, tal que la frecuencia del carácter _ en cada una de las cadenas de D^i, es *i*. Cuando una rama de la parte-árbol ya no discrimina se pasa a la parte-cadena que son listas encadenadas formadas por los restantes pares *_f*, donde _ es un carácter, *f* es la frecuencia con que aparece _ y el enlace *e* de la lista direcciona a un nodo del mismo tipo o a un nodo, **nodo-SIT**, de **sinónimos-DIT**, es decir, cadenas formadas por los mismos caracteres.

Debido a que se considera que la no aparición de la tilde y la diéresis supone un costo cero de edición a efectos de evaluación de DIT y DD, es por lo que no se han tenido en cuenta estas puntuaciones en la construcción del índice. Ello no es óbice para localizar en el texto las palabras correctamente escritas.

SISTEMA DE DESAMBIGUACION MORFOLOGICA PARA ORDENADORES PERSONALES

Agustin Reinaldos Meca
Ignacio Moreno-Torres Sánchez

Jean Piero Zacci
Giovanna Turrini

Depto de Lenguajes y CC
Facultad de Informatica
Universidad de Malaga
España

Giuseppe Cappelli
ILC Pisa

RESUMEN

Presentamos a continuación una herramienta cuyo objetivo, junto con otros programas que están siendo desarrollados actualmente, posibilitar la creación de corpora lematizados, una de cuyas aplicaciones será la obtención de diccionarios de frecuencia (del español e italiano).

Entendemos por corpus lematizado un texto en el que cada palabra tiene asociada una etiqueta de información lingüística en principio su lema (que suele coincidir con la raiz), y información morfológica sobre la clase de palabras, el número, género, persona...

Para obtener un corpus lematizado seguimos los siguientes pasos. En primer lugar un analizador morfológico o diccionario de formas debe asignar a cada palabra tantas etiquetas como soluciones el analizador morfológico ofrezca (e.g. a la palabra «vino» le podríamos asignar etiquetas «nombre» y «verbo»).

En segundo lugar, dada una palabra con más de una etiqueta, debemos seleccionar la correcta en cada caso. El programa que aquí presentamos es el que resuelve este problema utilizando información estocástica y reglas de contexto. La herrmienta es, en nuestra opinión, de gran valor por diversas posibilidades que ofrece tales como la modificación del conjunto de etiquetas (a partir de un conjunto predefinido), la ayuda a la creación de reglas de contexto, o la posibilidad de modificación de la fórmula de transición (a partir de la cual obtenemos las estadísticas) y de mantener diferentes estadísticas para diferentes tipos de textos.

SOFTWARE TOOLS FOR CORPORA ON PC

INTRODUCTION

At the ILC of Pisa and the University of Malaga we are preparing a set of programs which should serve to create a grammatically- tagged corpora of Spanish/Italian and a frecuency dictionary of each language. The system will work under a PC-environment and it has been divided in two independent modules: dictionary management and disambiguation. Both modules are language independent.

Our main objective has been to create an easily transportable tool that could be distributed to different research groups. For that it needed a «packed» dictionary and a language independent, theory-neutral disambiguation system.

1. THE DICTIONARY MANAGER

The morphological analyzer (M.A.) the former of two modules which act in cascade and associate each word in a text with one lemma and morphological code. Given a word, the M.A. associate it with the lemma, or lemmas, which it comes from, each with the corresponding

morphological code (or codes). Thus it does not resolve any lexical ambiguity; that is, in fact, the purpose of the disambiguating module.

M.A. is indeed a dictionary manager in that it uses or organizes one or more dictionaries, represented as lists of forms rather than by means of inflexion paradigms. Each form appears only once and is the key to achieve the associated information: lemmas and morphological codes.

Searching a word follows a pattern matching criterion, hence the accepted lexicon is constituted by the collection of words explicitely inserted, mostly, in one dictionary.

The format of the dictionary may be defined in terms of a BNF grammar as follows:

<Dict>::<Line>|<line>sep1<Dict>
<Line>::<Form>sep2<Info>
<Info>::<Lemmainfo>|<Lemmainfo>sep3<Info>
<Lemmainfo>::<Lemma>sep4<PoSInfo>sep5<MorphoInfo>
<MorphoInfo>::<MorphoCode>|<MorphoCode>sep6<MorphoInfo>

<Form>,<Lemma>,<PoSInfo>,<MorphoCode> are secuences of asciicodes that do not contain special symbols: sep1, sep2, sep3,sep4, sep5, sep6.

Here are some examples from the Italian dictionary:

a#a\E[$a\SN[NN
abate#abate\SM[MS
abbagli#abbagliare\VTIP[S2IP,S1CP,S2CP,S3CP$abbaglio\SM[MP

Actually, one may also provide the presence of use counters, which may be useful for later adaptations upon dictionary contents, with the further aim of building frecuency lexicons. The use of counters should be adjusted as a consecuence of text analysis; in this phase there also exists the possibility of finding new words and thereby increasing the dictionary.

The M.A. module includes some useful tools that allow the user to examine and/or manage both dictionaries and other kinds of data such as the lists of part of speech and morphological codes or the like. Besides it is possible to extract collection of words depending on their counter values.

Each dictionary may be compressed and indexed, to increase performance and save space (more than 50% theoretically, it might be reached space savings greater than 75%, for instance by ulteriorly coding the compressed dictionary with the Huffman method). At present, the compression method retains byte alignment (that is why Huffman method is not applied); furthermore dictionaries may be accesed and used in their compressed form. The compression technique prunes the form leading characters that match those in the preceding form, and replaces it with an integer which counts the the number of cut characters; analogously, it prunes the lemma's leading characterswith respect to the form they belong to. A translation table will store the one byte code for part of speech and morphological codes.

2. DISAMBIGUATION

We have found two different types of disambiguating systems for LN:

- rule based systems such as MORFSIN (ILC, Pisa 198?): a complex set of context rules should decide which is the correct analysis for one word.
- stocastic systems such as TAGGIT (Garside,Leech,Sampson 1987): the posibilities that a category A is followed by another category B are studied and represented in a Transition Matrix (TM). In case we have an ambiguous secuence of words:

Henry likes stews
NP NNS NNS
VBZ VBZ

we find different possibilities or secuences of categories; in this case we find the following secuences:

NP NNS NNS
NP VBZ NNS
NP NNS VBZ
NP VBZ VBZ

(NP = proper noun
NNS = singular common noun
VBZ = 3rd person of lexical verb)

If we multiply the values that we have in the TM for:

NP NNS

by the value that we have for:

NNS NNS

we get a vaue for the first secuence.

In this way we may obtain different value for each secuece the higher of which should be the one of the correct secuence.

Garside et al. (1987) describe the ways in which such idea may be improved to obtain quite good results for English real texts.

Aparently a rule based system should be easier to control, since it is the linguist who adds whatever he knows to be correct; but stocastic methodologies have, in fact, proved to work well for such problems. We feel that in both lines one might come to a robust system; but the problem is not so much with the computational methodology as it is with the correct classification of the words of the language that you wish to disambiguate.

That is a problem which we would not want to solve now; but we wanted to avoid the possibility that: in case you want to change the categories or subcategories, you have to change the whole system.

At the same time we needed a system that would transportable to another language (at least Italian, Spanish).

That led us to the idea that we needed a 'disambiguator generator'. That is, a system which would generate a different set of rules or MT for different languages (or even sublanguages). The difficulties to obtain a good set of context rules, for one language alone, made us decide in favour of a stocastic methodology. But, as we knew that statistics could not solve everyting, the possibility to create context rules was added. Moreover, the system suggests possible context rules to the linguists for those cases where statistics are not enough.

Briefly the system offers these possibilities:

1. Given a disambiguated text (it should have a minimun of 5000 words) and a set of grammatical categories it creates a MT for that set of categories. It also extracts «every

ambiguous context» and keep it in a separate ambiguous-context-file. After examining 15.000 words it should be possible for the user to create out of this file a set of context-rules.

2. The set of categories may be changed any moment. Obviously, the transition matrix should be generated again, but it is immediate if you have already tagged texts.

3. The context rules include up to 5 categories. These secuences of 'more than two' categories work like transitions with value = 0 . They can be of the following types:

1- those which say that a secuence is incorrect;

2- those which say that among two possibilities one is correct; for instance, the word «que» after a verb is 'always' a conjuntion and not a relative pronoun.

3- those wich say when there are certain types of ambiguity the system should not decide which one is correct. For instance «que» in Spanish seems in some contexts imposible to disambiguate unless you do syntactic analysis.

The '*' with its usual meaning can be used to define contexts more freely.

With these rules we expect to increase the reliability and robustness of the system; Making it give a solution when it is possible, and making it ask a question when it may not solve the problem.

4. A 'rarity marker' (as described by Garside et al.) has been absolutely necessary to correctly predict that, for instance,

era (WAS/NOM)

is almost always a form of the verb 'ser' (to BE).

Some other cases are:

pero (CONJ/NOM) but/apple tree
como (CONJ/VER) how/I eat
eras (WAS/NOM) was/age
nada (PRON/NOM) nothing/nothingness
para (PREP/VER) for,to../he stops,stop(imperative form)

5. The default transition function is multiplication, but it is possible to define another transition function using common arithmetic operators and some parameters of the system: number of times any category has appeared to obtain the TM total number of words processed to obtain the MT In this way we can avoid the 0s of rare secuences.

UN SISTEMA DE RECONOCIMIENTO DEL ESPANOL CON UN LEXICO DE 30.000 UNIDADES

Anne DEMEDTS

Dragon Systems, Inc en Boston

En 1989 Dragon Systems, Inc. presentó la primera versión del DragonDictate para el inglés americano en Speech Tech en Nueva York : se trata de un sistema de reconocimiento del habla que se adapta a la fonética particular del usuario y que reconoce en tiempo real un amplio léxico de 30.000 unidades en palabra desconectada. Actualmente se están finalizando las versiones española, francesa, italiana y neerlandesa ; el sistema alemán ya está en fase de «beta-testing» en Alemania.

Según esta tecnología, primero hay que crear modelos acústicos de todos los fonemas de una lengua determinada, considerados en la variedad de todos sus contextos posibles. A partir de ahí se elaboran Modelos Ocultos de Markov (Hidden Markov Models) cuyos parámetros pueden ser re-evaluados a base de una información fonética mínima. En el léxico, pues, la ortografía de una unidad fonética va acompañada del correspondiente modelo acústico y, también, de un indicio de frecuencia. Los datos estadísticos son actualizados constantemente de acuerdo con el idiolecto del usuario.

En el proceso de reconocimiento intervienen tres componentes básicos, que suponen una aproximación gradual entre un conjunto de probabilidades y la palabra dicha por el usuario : un algoritmo de verificación rápida, un conjunto de modelos acústicos de palabras basados en Modelos Ocultos de Markov y el ya referido modelo lingüístico estadístico.

ADAPTACIÓN FONÉTICA

En cualquier momento DragonDictate dispone de un conjunto de 30.000 palabras caracterizadas en tramas de 20 milisegundos por medio de 8 parámetros acústicos. Esto no implica que un hablante de referencia haya tenido que decirlas todas ni para el usuario que haya que hacer lo mismo a fin de personalizar el léxico porque un vocable se compone de cierto número de 'fonemas en contexto' (PIC : Phonemes In Context) que no le son privativos. A su vez un fonema siempre consta de, hasta 6 diferentes, elementos fonéticos (PEL : Phonetic Element) que se encuentran compartidos por varios alófonos del mismo. De esta manera se entablan relaciones, a menudo muy complejas, entre diferentes unidades. También explica que el DragonDictate 'aprende' a medida que el usuario lo utilice, gracias a esta extrapolación de datos.

A esta información fonética básica se añade otra de tipo durativo relacionada con tanto la ubicación del acento como la estructura de la palabra. Considérese, por ejemplo, que en español la vocal acentuada es relativamente larga en palabras agudas que no terminen en 'n' o 'l' («papá»), mientras la vocal inacentuada es generalmente breve. Si bien es cierto que el papel de estos factores varía, las mismas herramientas infórmaticas son válidas para cada lengua.

En cuanto a reconocimiento bastan para la caracterización fonética del español 11 vocales y diptongos, y 22 consonantes. Para las vocales se distinguen, además, la ocurrencia en sílaba tónica y la realización átona.

PROCESO DE RECONOCIMIENTO

Tampoco los componentes integrantes del mecanismo de reconocimiento dependen fundamentalmente de la lengua utilizada.

El algoritmo de verificación rápida efectúa una primera selección dentro del conjunto léxico comparando la secuencia inicial de la unidad detectada con la de unos cientos o, aun, miles de grupos

en los que ha reunido previamente palabras que empiezan de la misma forma. Así el número de palabras eventualmente dichas se reduce a unos doscientas.

El algoritmo de programación dinámica hace uso de Modelos Ocultos de Markov estableciendo una compleja red de probabilidades entre diversos PELos acompañados por la expresión en milisegundos de su duración calculada en base al entorno lingüístico.

De momento la función del modelo lingüístico se limita a la estimación de la probabilidad de una palabra, teniendo en cuenta la palabra anterior. Se basa en un cálculo de frecuencias, que podrá incluir también frecuencias de pares de palabras.

LEXICO

En el sistema de reconocimiento para el inglés americano, el vocabulario consta de las 25.000 unidades léxicas más frecuentes, a las que el usuario puede añadir hasta 5.000 otras nuevas.

En una etapa intermedia el léxico español peninsular se construyó a partir de las 5000 formas recogidas en el diccionario de frecuencias de Juilland. A estas formas básicas, se les añadieron las derivadas morfológicamente : las formas verbales existentes, el plural de los sustantivos, el plural y, eventualmente, el femenino de los adjetivos. Dentro del conjunto así obtenido, se seleccionaron las 20.000 formas verbales más frecuentes, todas las sustantivales (5025), todas las adjetivales (4137) y todas las demás (312).

Para que no resultaran completamente arbitrarios, los indicios de frecuencia correspondientes se calcularon sobre la base de los recuentos detallados que figuran en el mismo diccionario para un número limitado de palabras. Considérese, por ejemplo, el caso de los sustantivos. Mediante la adición y posterior división de las frecuencias facilitadas para el singular y plural de 200 nombres, se obtiene la frecuencia 'genérica' del singular relativo al plural. Luego, la frecuencia de uso de todo sustantivo del léxico se puede repartir entre singular y plural según las mismas proporciones.

Así, por ejemplo, suponiendo que un 60% de las ocurrencias de un nombre corresponden al singular y un 40% al plural, tendríamos el siguiente resultado :

«casa», supuesta frec. 120 -> sg.'casa', frec. 72
pl.'casas', frec. 48
«almidón», supuesta frec 10 -> sg.'almidón', frec. 6
pl.'almidones',frec. 4

Hicimos unas pruebas acerca de la representatividad del léxico obtenido de esta manera. Así, tomamos, entre otros, un artículo de «El País», - «Cara y cruz de hablar mi idioma» - que consta de 614 palabras distintas. Para 416 palabras del artículo la forma morfológica básica figura también en el Juilland ; para las 198 palabras restantes, esta forma no está en el Juilland. No obstante, 59 de este grupo de palabras son nombres propios.

De todas formas y teniendo en cuenta el carácter dinámico de la lengua, será necesario delimitar el vocabulario mediante métodos estadísticos aplicados a la mayor cantidad de textos posible, aunque ya parece que se planteará un problema de diversificación por la índole predominantemente sociopolítica y económica de los textos disponibles en soporte digital.

INFORMACIÓN PRÁCTICA

- Dragon Systems, Inc.
  320 Nevada Street
  Newton, MA 02160
  USA

Tf 1-617-965-5200
Fax# 617-527-0372

– A.Juilland, «Frequency Dictionary of Spanish Words», 1964, The Hague : Mouton.

– Bamberg P.G., «Adaptable Phoneme-based Models for Large-Vocabulary Speech Recognition», Proceedings of the ESCA Tutorial and Research Workshop on Speaker Characterization, University of Edinburgh, U.K., June, 1990.

– Bamberg P. et al., «Incorporating natural-language information into a statistical language model for large-vocabulary recognition», Proceedings of SpeechTech, New York, 1992.