PROGRAMA DE EXTRACCIÓN AUTOMATICA DE PALABRAS CON SU CONTEXTO: REAL[1]

IX congreso SEPLN
Santiago de Compostela, 8 - 10 Setiembre 1993

José Lázaro Rodrigo Mateos
Groupe de Recherche dans les Industries de la Langue
- GRIL -

Université B. Pascal
34 Av. Carnot, F - 63037 Clermont Ferrand Cedex
Fax : 73 40 63 99
jose@gril.univ-bpclermont.fr

• 0 ABSTRACT

REAL is a computer program for automatic searching and extraction of lemmas. It allows the user to extract from large corpora of texts all the different occurences of defined lemmas along with their context of appearance.

It was initially developed as part of the European project DELIS and it has been used in the framework of my thesis to study Spanish verbs which express information transfer.

• 1 INTRODUCCION

REAL es un programa que busca lemas (palabras) en sus diferentes formas flexivas, a partir de cualquier corpus de ficheros de texto, en cualquier tipo de soporte magnético, disquette, CD-ROM, etc.

REAL construye automáticamente una base de lemas con cada ocurrencia de cada lema, la forma flexiva encontrada, el contexto en que aparece, la información morfológica asociada y el número de la ocurrencia.

Al mismo tiempo crea un fichero de texto "tagged" en el que los lemas buscados quedan marcados físicamente con el fin de facilitar un trabajo posterior, si por ejemplo el contexto extraído resulta no ser suficiente para el estudio lingüístico y se desea contar con un contexto más amplio.

El programa se encuadra en el proyecto europeo DELIS - LRE project 61034[2], y ha sido concebido, con el objetivo de darse un instrumento de trabajo para estudiar corpus de textos de volumen considerable.

---

1 REAL: "Recherche et Extraction Automatique de Lemmes", programa realizado por Salah Aït-Mokhtar, Caroline Dafniet, Pascal Gangutia y José L. Rodrigo Mateos, GRIL: Clermont - Ferrand, Febrero, 1993

2 DELIS - LRE 61-034 , Descriptive Lexical Specifications. proyecto de investigación de la CEE en que participa el GRIL, con IMS-CL Universität Stuttgart, Copenhagen, Oxford University Press, HEL Helsinki, LCB London, ILC Pisa, SITE Paris, VDA - Utrecht, VUA Amsterdam.

En su primera versión, el programa se ha realizado en un espacio de tiempo realmente breve, un mes escaso, desde mediados de enero 93, lo que justifica que por el momento no abarque la totalidad de las categorías gramaticales.

En la actualidad se aplica en español a cuatrocientos veinte verbos que denomino de "desplazamiento de la información", para mi tésis, y a un corpus de verbos de percepción del francés.

Los tests realizados muestran que puede extenderse sin problema mayor al resto de las categorías gramaticales, signos de un texto en general, y a otras lenguas de grafía latina.

• 1.1. PRINCIPIOS METODOLOGICOS

De acuerdo con las directrices actuales de la investigación en lingüística computacional, los trabajos sobre el léxico deben dar cuenta de las informaciones que deberían figurar ya en los diccionarios, pero basándose en el uso concreto de la lengua; el castellano y el francés en este caso.

Ello implica el estudio de un corpus de referencia amplio que sea representativo de la lengua, con textos periodísticos, científicos, jurídicos, comerciales, etc. Dado que todavía no disponemos del corpus de referencia generalizado del español,[3] intento configurar un corpus propio lo más representativo posible, centrándome fundamentalmente en la lengua escrita por el momento.[4]

Un estudio de tal envergadura exige un tratamiento informático apropiado. Por ello el equipo GRIL se planteó, en el marco del proyecto DELIS, la creación de un útil informático, que satisficiera las necesidades de los investigadores:

- Tratar automáticamente los ficheros informáticos de texto.
- Extraer todas y cada una de las ocurrencias de un lema con su contexto.

Es importante subrayar que la concepción del programa no ha sido pensada a partir del inglés, sino a partir de los problemas de las lenguas en general y de las románicas en particular.

La arquitectura se basa en módulos independientes, que en la medida de lo posible permiten con mínimas modificaciones hacer frente a las necesidades concretas de cada nuevo caso particular. Por el momento tales modificaciones se ciñen a definir la lista de lemas que se quieren estudiar, y a especificar la lengua de trabajo.

---

3 Marcos Marín está trabajando sobre el corpus de referencia del español, previsto para 1994

4 Si bien hemos trabajado con una parte del corpus oral de la Universidad Autónoma de Madrid, que REAL ha tratado.

Frente a las particularidades de cada lengua y de la gama de sistemas informáticos, adoptamos una codificación, que nos permita reconocer:

- los caracteres ASCII de PC o de MAC o UNIX,
- la acentuación: acentos agudos, graves, circunflejos, diéresis, etc.
- signos de puntuación.
- caracteres propios a cada lengua, existentes en ASCII, como por ejemplo::
  - ß alemán
  - ñ español
  - ã õ portugués
- mayúsculas y minúsculas.

Sabiendo que en el corpus se encontrará la lengua tal y como el hablante la escribe, es previsible que haya errores, como por ejemplo en el caso de la acentuación de palabras como "*oír*".

Por ello se ha previsto un sistema de tolerancia de errores, parametrable por el investigador, de modo que REAL pueda reconocer las palabras que interesen, "*oir*" y "*oír*" en este caso

1.2. ESPECIFICACIONES TÉCNICAS.

Real, ha sido implementado en Delphia Prolog versión 2.5 sobre OPENWIN 2 en un SUN4 con un sistema SOLARIS 1.0.1.

2 ARQUITECTURA DE REAL

Partiendo de un corpus de ficheros en formato ASCII, el programa opera con cinco procesos fundamentales:

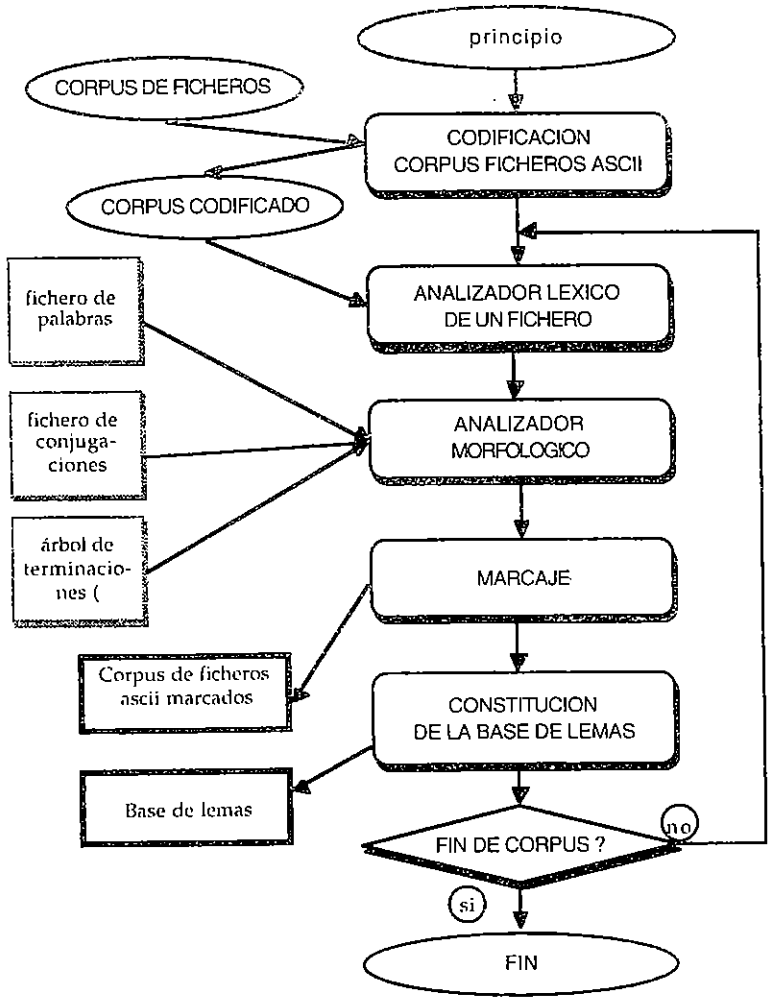
1 Codificación del corpus de ficheros ASCII.

2 Separación en palabras con un analizador léxico.

3 Verificación de cada palabra con la lista de lemas dados por medio de un analizador morfológico.

4 Si la verificación es positiva, marcaje del lema y creación de un corpus de ficheros con tales lemas marcados, y

5 Constitución de una base de lemas.



2.1 CODIFICACION

Con el sistema Unix aparecen problemas para manipular los caracteres acentuados ASCII, a partir del número 127:

"confirm¢ su participaci¢n en el Torneo Internacional de Atlestismo de Sao Paulo, que abrir† la temporada"

Por otra parte, dado que la codificación ASCII de IBM y APPLE no es totalmente igual, hemos construido una tabla de conversión de caracteres que REAL utiliza.

Según declaremos el origen "PC" o "MACINTOSH" del fichero, REAL asocia un código que permite reconocer todos los caracteres ASCII. La codificación consiste en asociar un número a cada tipo de acento o caracter particular.

Así al acento agudo corresponde el número "1", al grave, el "2", al circunflejo, el "3", a la diéresis, el "4", a la tilde, el "5", y así sucesivamente con los caracteres que son susceptibles de aparecer en alguna lengua particular.

Las mayúsculas son tratadas con el mismo código que las minúsculas.

CARAC-TER	ASCII PC	ASCII MAC	CODIFI-CACION REAL	ASCII	
é	130	142	e1	101	49
è	138	143	e2	101	50
ê	136	144	e3	101	51
ü	129	159	u4	117	52
ñ	164	150	n5	110	53
õ	-	155	o5	111	53
ª	166	187	a9	97	57
¿	168	192	?0	63	48
É	144	131	e1	101	49
Ñ	165	132	n5	110	53

De esta manera, nuestro programa está preparado para reconocer en principio los caracteres ASCII pertinentes para cualquier lengua de grafía latina.

Real construye un corpus de ficheros codificados, que podrá ser utilizado en el módulo siguiente.

• 2.2 ANALIZADOR LÉXICO

Una vez codificado, el programa extrae uno a uno cada fichero, y con el analizador léxico Lexia, de Delphia Prolog, asocia a cada cadena de caracteres un tipo de los que hemos declarado en las reglas sintácticas.

Sólo las cadenas que son susceptibles de ser lemas, reciben el tipo "mot", es decir, ,sólo serán analizadas morfologicamente las palabras.

De esta manera, REAL leerá las palabras de los ficheros sin modificar en salida las estructuras de los mismos, lo que nos permite trabajar con textos balizados SGML, etc.

• 2.3 ANALIZADOR MORFOLOGICO

En este módulo, Real va a comparar cada palabra con una lista de lemas definida de antemano.

Hemos utilizado el modelo descrito por Jacques Pitrat (83) basado en la declaración de datos lingüísticos, con escasas modificaciones que portan fundamentalmente a una búsqueda más ágil, renunciando a las posibilidades de generación, posibles con su modelo.

En el estado actual de REAL, y dados los objetivos presentes de los investigadores del GRIL, nos ceñimos a la implementación del sistema verbal, dejando por el momento la de nombres, adjetivos etc.

Sin entrar en detalle, repasaremos muy brevemente los puntos esenciales del modelo.

Hay tres tipos de datos posibles: palabras, terminaciones y conjugaciones, que corresponden a tres ficheros:

- **2.3.1 FICHERO DE PALABRAS:**

Cada entrada está constituida en este fichero de:
- el nombre de la palabra
- el nombre de la conjugación
- de una serie ordenada de las raíces necesarias utilizadas para conformar las formas flexivas:

palabra	conjugación	serie ordenada de raíces
aconsejar	amar	aconsej
sentir	sentir	sient, sent, sint
advertir	sentir	adviert, advert, advirt
haber	haber,	h, hab, hub, hay
decir	decir	dig, dic, dec, dij, d, di

Así, "*advertir*, " se conjuga según el modelo de *"sentir"*.

- **2.3.2 FICHERO DE TERMINACIONES:**

A cada conjunto de seis terminaciones, le asociamos un nombre. Por ejemplo, para el presente de indicativo (vip):

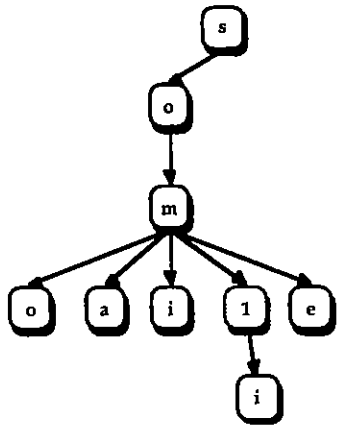
vip1	o	as	a	amos	áis	an

El orden sucesivo de cada terminación determinará la persona asociada a esa terminación; de esta manera "as" en el segundo lugar, indicará la segunda forma personal, "amos" en el cuarto, la cuarta forma personal, etc.

Cada conjunto de seis terminaciones diferentes del mismo tipo, en este caso, del presente de indicativo, (vip) tendrá un número sucesivo, 1, 2, 3...

NOMBRE DE LA TERMINACIÓN	SERIE DE TERMINACIONES					
	1	2	3	4	5	6
vip1	o	as	a	amos	áis	an
vip2	o	es	e	emos	éis	en
vip3	o	es	e	imos	is	en
vip4	e	as	a	emos	abéis	an
vip5	oy	res	s	omos	ois	on
vip6	o	s	e	emos	eis	en
vip7	o	es	e	ímos	ís	en
vip8	oy	ás	a	amos	áis	án
vip9	é	es	e	emos	éis	en

Con el conjunto de estas terminaciones, construyo unos árboles a partir del último caracter escrito de derecha a izquierda:



En el árbol hay hojas (o, a, i, i, e) y nudos (s, o, m, 1)

A cada hoja del arbol, que corresponde a una terminación, le asocio la información morfológica correspondiente en forma de pareja

[tipo de terminación,rango].

El rango indica la posición ocupada por la terminación:

s : [vip6,2], [vip5,3]

que corresponden a la segunda forma verbal del sexto tipo del presente de indicativo (vip) y a la tercera forma verbal del quinto tipo del (vip), respectivamente.

De esta forma se puede organizar rápidamente la disposición de la información que necesita el analizador.

- **2.3.3 FICHERO DE CONJUGACIONES:**

Tiene como finalidad indicar las relaciones pertinentes entre las raíces y las terminaciones. Cada entrada está compuesta de:

- el nombre de la conjugación
- el nombre de la terminación
- una serie ordenada de números de raíces

NOMBRE DE LA CONJUGACIÓN	NOMBRE DE LA TERMINACIÓN	RAÍCES					
		1	2	3	4	5	6
amar	VIP1	1	1	1	1	1	1
probar	VIP1	1	1	1	2	2	1
publicar	VIP1	1	1	1	1	1	1
decir	VIP3	1	2	2	3	3	2
sentir	VIP3	1	1	1	2	2	1
haber	VIP4	1	1	1	1	1	1

Así por ejemplo, el verbo "*sentir*" utiliza la primera raíz declarada en el fichero de palabras, "*sient*" y el tercer tipo de terminaciones "vip3" para conformar las formas flexivas 1ª, 2ª, 3ª y 6ª del presente de indicativo:

sient + o, es, e, - , - , en

y la segunda raíz declarada para las formas flexivas 4ª y 5ª

sent + -, - , - , imos , is , -,

• 2.3.4 VERIFICACION DE PALABRAS:

Con la representación en forma de árbol de las terminaciones, Pitrat propone un algoritmo (pag 8) con el que, a partir de una palabra toma dos series, de tal manera que:

palabra = ($serie_i$ + $serie_{n-i}$),

siendo **n** = número total de caracteres de la palabras

La primera serie comporta los **i** primeros caracteres
La segunda serie comporta los **n-i** caracteres finales

En el momento inicial, el valor de **i** = **n**.

Por ejemplo:

a1 a2 a3 a4 a5 a6 a7 a8
s e n t i m o s **n = i = 8**

El algoritmo verifica si la serie (a_n.....a_{i+2} a_{i+1}) existe en el árbol de terminaciones.

En caso negativo, el análisis fracasa.
En caso positivo, captura la información morfológica en forma de parejas, **[terminación, rango]** y verifica si estamos en una hoja.

Si en lugar de una hoja, es un nudo, el análisis continúa, dando a **i** el valor de **i** = (**i**-1)

A continuación, el algoritmo verifica si la serie (a_1 a_2 a_3 ... a_i) es una raíz, declarada en fichero de palabras y en caso afirmativo, construye la palabra.

• 2.3.5 MODIFICACIONES

Decidimos que además de reconocer si una palabra del texto es una forma flexiva REAL, nos de todas las informaciones morfológicas, para cada forma verbal.

Por ejemplo para, la forma "*-aba*", nos tendrá que decir que corresponde a la 1ª y 3ª forma del imperfecto de indicativo.

Por razones de agilidad, queremos el algoritmo de REAL, secuencial y determinista, de forma que no vuelva atrás en el árbol de terminaciones, según va descendiendo.

Para ello hemos numerado cada nudo y cada hoja de los árboles del terminaciones, y asociado a cada nudo una información denominada 'fils' que marcará el camino a seguir, de forma que nos damos un util que nos permitirá ir de un nudo de un árbol a otro nudo u otra hoja de otro árbol.

En realidad nuestro árbol es al menos una red, con multitud de punteros entre los nudos, a distintos niveles.

Cada árbol comienza obligatoriamente con el número cero, y está declarado en forma de hecho Prolog.

Ello implica que siempre capturamos la información asociada cada vez que un caracter de una cadena de caracteres es susceptible de pertenecer a una terminación, incluso si no hay información morfológica, ya que en alguno de los nudos indicados en 'fils', estará asociada.

Tomamos misma partición de $\mathbf{n}$ e $\mathbf{i}$ del modelo.
Verificamos, si la serie (a_n.....a_{i+2} a_{i+1}) a_n existe en el árbol

En nuestro ejemplo,comenzamos por a_n = "s" y pasarían positivamente el test, palabras como, *"las"*, *"camas"*, *"sentimos"*....

En caso negativo, el proceso fracasa, y REAL toma la palabra siguiente.

En caso positivo, capturamos la información morfológica y la información 'fils',.

Verificamos si la serie ordenada (a_1 a_2 a_3 ... a_i) se corresponde con la raíz determinada en el fichero de palabras. (sentimo-)

En caso negativo, i adquiere el valor de **i**= (**i**-1) y el análisis sigue, continuando la búsqueda directamente en los nudos indicados en "fils"

En el caso afirmativo de encontrar la raíz, construimos la palabra, y asociamos a una serie de variables, las informaciones que nos interesa retener del proceso: nombre del lema, conjugación, y contexto.

Veamos el ejemplo propuesto:

a_n = s, vemos la terminación "s" en el árbol de terminaciones.
Nuestro algoritmo busca si hay un árbol que comience por cero, = s0

noeud s0,: **[vip5,3],[vip6,2]**
[i,1],[o,2],[a,18],[1,11],[e,18]

donde **[vip5,3],[vip6,2]** es la información morfológica asociada y **[i,1],[o,2],[a,18],[1,11],[e,18]** el 'fils'

La comparación de (a_1 a_2 a_3 ... a_i) (s e n t i m o), con el fichero de raíces fracasa, y continuamos la búsqueda por el camino marcado por el 'fils' "o2"

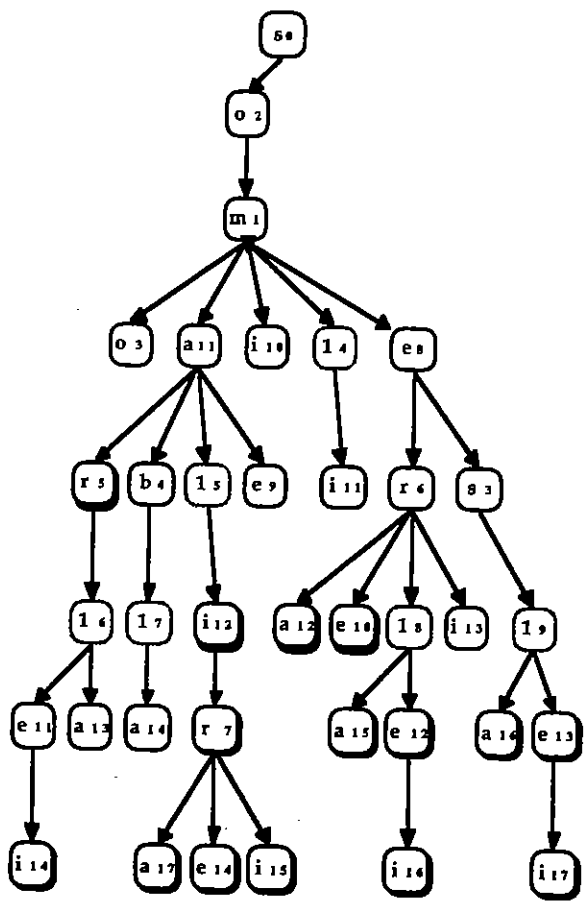
terminación "-os"
noeud o2,[ ], información morfológica asociada al nud
[m,1] 'fils'
comparación con raíz, (sentim-), fracasa

terminación "-mos"
noeud m1, [ ]
[o,3],[a,11],[i,10],[1,4],[e,8]
comparación con raíz, (senti-), fracasa

terminación "-imos"
noeud i,10,**[vip3,4]**
La comparación de (a_1 a_2 a_3 ... a_i) (s e n t), coincide con "sent", en el fichero de raíces

El análisis es positivo., con lo que en el módulo siguiente, el lema será marcado, y se recuperará el contexto en el que se encuentra.

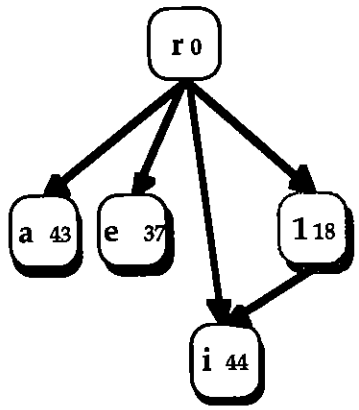
A título de ejemplo, mostramos una rama (-mos) del árbol construido a partir de la "s"



La agilidad del sistema, nos permite "saltar" entre los árboles, y además nada nos prohibe asociar la misma información morfológica a dos nudos diferentes. Este es el principio que nos permite regular un sistema de tolerancia de errores.

Por ejemplo, si queremos recuperar automáticamente el lema "oír", correctamente acentuado, tal y como viene en el diccionario, a partir de textos escritos, corremos un grave riesgo de no reconocerlo ya que en un alto porcentaje de casos, encontraremos "oir" en su lugar.

Con nuestro sistema asociamos la información correspondiente a los dos nudos, creando en realidad una red que permite extraer las ocurrencias erróneas y las correctas:



Si bien la elección de las raíces sigue en general la norma gramatical, en ocasiones tenemos la opción de reducirlas e incrementar los tipos de terminaciones. Pitrat recomienda minimizar el número de raíces, pero dado nuestro principio secuencial y descendente, no es necesario darse esta norma, sino más bien al contrario, ya que ayudará a reducir el campo de búsqueda.

2.4. MARCAJE

Cada vez que el resultado de la verificación del analizador morfológico resulte positivo, REAL marca físicamente en el espacio inmediatamente anterior a la palabra, en una copia del fichero, de modo que si necesitásemos trabajar con este fichero, reconoceríamos inmediatamente, los lemas marcados.

El marcaje para cualquier forma de un verbo consiste en una pareja

<INFINITIVO, Número de ocurrencia>

2.5. CONSTRUCCIÓN DE LA BASE DE LEMAS

Cada vez que el resultado de la verificación del analizador morfológico sea positivo, el algoritmo realizará una serie de acciones cuyo objetivo es grabar los valores de diferentes variables que se "pasean" a través de los módulos, configurando una base de lemas.

Llamamos base de lemas a una serie de hechos prolog, con una serie de variables instanciadas, que van a contener:

- 1 nombre del lema, por ejemplo: "*decir*"
- 2 ocurrencia, por ejemplo, la forma flexiva, "*decir*"
- 3 número de aparición de la ocurrencia
- 4 nombre del fichero en que se encuentra
- 5 lista de información morfológica asociada, por ejemplo: [vi3,1]
- 6 contexto de la ocurrencia:

  sobre todo en Japo1n . El an5o pasado , Australia fue el primer exportador mundial de carne vacuna (es decir ## de vaca y de ternera) , quitando ese puesto a la CEE , la cual disminuyo1 las ventas.

DECIR
Occurrence :decir
Numero :87
Fichier :a259.cod
Conjugaisons:[[vi3,1]]
Contexte :

sobre todo en Japo1n . El an5o pasado , Australia
fue el primer exportador mundial de carne vacuna (es
decir ## de vaca y de ternera) , quitando
ese puesto a la CEE , la cual disminuyo1 las
ventas.

Real marca el texto en el lugar inmediatamente posterior al ocupado por la ocurrencia

• 4 CONSULTA DE LA BASE DE LEMAS

Una vez obtenido el contexto en forma de hecho prolog, se puede trabajar con él de muchas maneras que estamos estudiando. Por el momento nos hemos conformado con poder organizar la búsqueda de los lemas que nos interesan, y así hemos creado diferentes programas que nos dan:

- todas las ocurrencias de todos los lemas declarados en REAL
- todas las ocurrencias de uno o más lemas determinados
- todas las ocurrencias de una forma flexiva de un lema determinado
- de entre la lista de lemas que han sido buscados, cuáles figuran en el texto y el número de ocurrencias

Igualmente Real crea automáticamente un fichero donde grabar todas las "fichas de la base de lemas" que nos interesan, que se pone al día, opcionalmente con cada sesión de trabajo.

• 5 EJEMPLOS

Algunas "fichas de lemas" del fichero "decir"

```
DECIR
Occurrence  :dijo
Numero      :88
Fichier     :a263.cod
Conjugaisons:[[vips5,3]]
Contexte    :

   Comunidad del Caribe ( Caricom ) aunque sus esfuerzos no
   fructifican por las normas de ese bloque econo1mico cariben5o ,
   dijo ## este mie1rcoles el vicecanciller , Fabio Herrera Cabral
   . Herrera Cabral dijo que por ello habri1a que cambiar
   algunos aspectos

-----------------------------------

DECIR
Occurrence  :dijo
Numero      :89
Fichier     :a263.cod
Conjugaisons:[[vips5,3]]
Contexte    :

   las normas de ese bloque econo1mico cariben5o , dijo este
   mie1rcoles el vicecanciller , Fabio Herrera Cabral . Herrera Cabral
   dijo ## que por ello habri1a que cambiar algunos aspectos
   de la poli1tica interna y externa del pai1s , sin
   mayores precisiones

-----------------------------------

DECIR
Occurrence  :dijo
Numero      :90
Fichier     :a263.cod
Conjugaisons:[[vips5,3]]
Contexte    :

   formacio1n de la Asociacio1n de Naciones Cariben5as con Venezuela ,
   Cuba , Haiti1 y Repu1blica Dominicana . El vicecanciller dominicano
   dijo ## que el pai1s mantiene observadores en la Caricom
   , que agrupa a 13 naciones anglo1fonas del Caribe .
   Venezuela ,
```

Algunas "fichas de lemas" de cinco lemas dados para tratar en dos documentos.

```
-----------------------------------

PERMITIR
Occurrence  :permitir
Numero      :2
Fichier     :A0002.cod
Conjugaisons:[[vi3,1]]
Contexte    :

   deberi1a ayudar a constituir " mecanismos financieros que permitan al
   Peru1 cancelar los atrasos en sus pagos , y asi1
   permitir ## al FMI la consideracio1n de nuevo financiamiento al
   pais " , explico1 la fuente monetaria . El presidente
   peruano Alberto
```

REVELAR
Occurrence :revelo1
Numero :2
Fichier :A0002.cod
Conjugaisons:[[vips1,3]]
Contexte :

FMI la consideracio1n de nuevo financiamiento al pais " , explico1 la fuente monetaria . El presidente peruano Alberto Fujimori revelo1 ## el sa1bado pasado en Lima que Estados Unidos estaba condicionando su intervencio1n en el grupo de apoyo a la resolucio1n

---

EXPLICAR
Occurrence :explica
Numero :1
Fichier :A0002.cod
Conjugaisons:[[vip1,3],[vim1,2]]
Contexte :

su intervencio1n en el grupo de apoyo a la resolucio1n de problemas de derechos humanos en Peru1 , lo cual explica ## la participacio1n del ministro de justicia en las gestiones que se realizan en Washington . La aprobacio1n del FMI al

---

CREER
Occurrence :creo
Numero :1
Fichier :A0002.cod
Conjugaisons:[[vip2,1]]
Contexte :

y dijo esperar que se hallara1n los medios de superar el obsta1culo . " Estamos trabajando sobre eso , y creo ## que tenemos posibilidades de encontrar los medios " , dijo Camdessus , al ser interrogado por la prensa al te1rmino

---

• 6 CONCLUSION[5]

REAL satisface en su primera versión, las necesidades que nos habíamos propuesto como instrumento para tratar corpus de textos. y se ha mostrado ya muy eficaz en la búsqueda de los verbos del español y francés.

Creo que es interesante subrayar que escaso tiempo de realización de la primera versión de REAL, (enero-febrero 1993).

Actualmente (abril 93) estamos en fase de estudio sobre el desarrollo del programa.

En un primer momento pensamos completarlo con el reconocimiento de otras categorías verbales, así como con un interfaz que le haga más convivial.

---

5 En el momento de redactar este artículo, he recibido el volumen del VIII congreso de la SEPLN, al que no pudimos asistir, y en el que se encuentran algunos artículos que parecen muy interesantes sobre la extracción automática de información,

En un segundo momento, hay varios proyectos para los que la arquitectura modular de REAL y su capacidad de reconocimiento de cualquier signo, parecen apropiados.

Nos parece que una vez realizado el trabajo duro de haber logrado extraer un contexto parametrable de un corpus fuerte de textos, las aplicaciones posteriores pueden ser abordadas con cierta esperanza.

En la actualidad un miembro del GRIL, utiliza REAL sobre los verbos autorizados del inglés simplificado AECMA (Asociación europea de construcciones de material aeronáutico) en el marco de un acuerdo del GRIL con AEROSPATIALE

Real ha extraído cerca de 4000 ocurrencias de cuatrocientos lemas, con sus correspondientes formas flexivas, a partir de cuatrocientos ficheros. que suponen en total una talla de 4 Megaoctets, empleando un tiempo de dos horas y media

- **BIBLIOGRAFIA**

PITRAT J. (1983) "*Réalisation d'un analyseur-générateur lexicographique général*" CNRS, Intelligence artificielle. Rapport de Recherche

PITRAT J. (1985) *Textes, ordinateurs et compréhension* Eyrolles Eds. Paris

SABAH G. (1988) *L'IA et le langage* Hermes, Paris

MORENO SANDOVAL, A; OLMEDA MORENO, C; GRISHMAN, R.; MACLEOD, C; STERLING,J. (1993) *PROTEUS: un sistema multilíngüe de extracción de información* Boletín 13 SEPLN

TORRUELLA CASAÑAS, J. (1993) *"Transcalc: del procesador de textos a la base de datos.* Boletín 13 SEPLN"