

Terminología y representación de los conocimientos: Análisis y generación automáticos de los términos derivados y compuestos de la terminología española de la informática

Albert Álvarez González
Universidad de Sonora, Hermosillo.

Debido a la rápida evolución de la información científica y técnica, se vuelve necesario para los especialistas en la planificación y el acondicionamiento terminológicos dotarse de herramientas específicas para localizar, analizar y generar los neologismos. En esta presentación, trataremos de explicitar los principales aspectos teóricos y prácticos del desarrollo de un sistema de análisis y generación por computadora basado en un estudio semántico integrado dentro del enfoque cognitivo desarrollado por el LILLA (Laboratorio de Ingeniería Lingüística y de Lingüística Aplicada) de la Universidad de Niza-Sophia Antipolis (Francia). En esta óptica, se ha desarrollado un programa informático, la ZSTATION (Zinglé 1994) que permite modelizar los conocimientos lingüísticos en los niveles inter e intralingüísticos en vista a un procesamiento automático de las lenguas, y más concretamente en nuestro caso de textos científicos y técnicos. Una de las aplicaciones posibles y correspondiente a nuestro trabajo consiste en el análisis y generación automáticos de términos derivados y compuestos mediante previo análisis morfosemántico de sus unidades morfológicas. Ejemplificaremos la presentación de este sistema con los neologismos de la terminología informática en español.

Palabras clave: ingeniería lingüística, modelización de los conocimientos, tratamiento automático, análisis morfosemántico, neologismo, terminología informática

Fecha de recepción del manuscrito: agosto de 2001

Albert Álvarez González, Universidad de Sonora, Bulevard Kino, 410, Int. 32, Colonia Pitic C.P. 83190 Hermosillo, Sonora, México, *correo electrónico*: aalvarez@lenext.uson.mx

Estudios de Lingüística Aplicada, núm. 34, 2001

As scientific and technical information is growing rapidly, specialists in terminologie planning need specific tools for a quick lookup, analysis and generation of neologisms. In this paper, we will introduce the main theoretical and practical features underlying the development of a software package for analysis and generation, which is based on a semantic approach integrated with the cognitive approach at the LILLA in the university of Nice-Sophia Antipolis (France). In this view, Henri Zinglé, founder and director of LILLA, has developed a computational program (Zingle 1994) which provides a modelization of the linguistic knowledge in the inter and intralinguistic levels for the purpose of processing natural language and, in our case, scientific and technical texts. One of the possible applications —corresponding to our work— is the automatic analysis and generation of derived and compound terms based on a previous morphosemantics analysis of the morphological units. The presentation of this system will be illustrated with selected terms of Spanish computer terminology.

El trabajo cuyos principales aspectos presentamos a continuación corresponde a un estudio realizado en el Laboratorio de Ingeniería Lingüística y de Lingüística Aplicada (LILLA) de la Universidad de Niza-Sophia Antipolis (Francia) y efectuado a petición de la Délégation Générale à la langue française. Al ser una de las misiones de este organismo el acondicionamiento terminológico, el trabajo intentó perseguir un doble objetivo: precisar los métodos que permitan ayudar a) a la localización y b) a la creación de neologismos basándose en un sistema informático a base de conocimientos.

Realizado en una óptica multilingüe dentro de la Red Panlatina de Terminología (REALITER), este proyecto se basa en una metodología desarrollada a partir de un enfoque lingüístico de la investigación terminológica. En nuestro caso, se trata de hacer un análisis lingüístico en vista al procesamiento automático de los textos científicos y técnicos y más concretamente un análisis formal y semántico de los términos derivados y compuestos de la terminología española de la informática en vista al análisis y generación automáticos de los neologismos de esta especialidad.

El LILLA ha sido creado en 1988 con la orientación principal de aplicar las aportaciones de la investigación en lingüística a la resolución de problemas concretos relacionados con las industrias de la lengua.¹ Para lograr este objetivo, el profesor Henri Zinglé, director y fundador del LILLA, ha elaborado un programa llamado ZSTATION que permite la modelización de los conocimientos lingüísticos (Zinglé 1994). Mostraremos a continuación las posibilidades de análisis y generación automáticos de términos informáticos en español que permite la ZSTATION. Expondremos, para eso, después de un breve comentario acerca de los presupuestos lingüísticos del sistema, los principios de formalización para, en un tercer punto, mencionar sus realizaciones informáticas.

1. Presupuestos lingüísticos

En el marco de un estudio sobre los neologismos, se ha desarrollado un conjunto de procesos que permiten representar el sentido de los términos derivados y compuestos y, de manera simétrica, de generarlos a partir de una representación semántica. El estudio se ha limitado a los casos de derivación y composición calculables, privilegiando un enfoque cognitivo fundado sobre el semantismo de las unidades morfológicas y el semantismo de las relaciones entre conceptos. La perspectiva adoptada une derivación y composición al introducir como únicos criterios tipológicos, la capacidad de las unidades morfológicas a admitir una expansión a la derecha y a la izquierda y la posibilidad o no de gobernar la construcción. Así, consideramos la unidad léxica derivada como el resultado de la combinación de un elemento léxico pivote con uno o varios afijos derivacionales que son o prefijos o sufijos según que se asocien con el elemento subse-

¹ Otra orientación importante del LILLA concierne la adquisición de las lenguas.

cuenta o precedente². La distinción se establece únicamente sobre la base de criterios formales de combinación:

- elemento léxico pivote: combinación a la derecha o a la izquierda
- prefijo: combinación exclusiva a la derecha
- sufijo: combinación exclusiva a la izquierda

Podemos mencionar que el elemento situado a la derecha o a la izquierda del afijo no es forzosamente un elemento léxico pivote. Por otra parte, en los afijos, se toman en cuenta los elementos llamados prefijoides (o seudo-prefijos) y sufjoides (o seudo-sufijos)³ por considerarlos una especialización afijal de elementos léxicos. Contrariamente a la unidad léxica derivada, la unidad léxica compuesta está constituida por varios elementos léxicos pivotes. Una unidad morfológica está representada por todo elemento formativo, es decir todo elemento léxico o afijal que entra en la formación de una unidad compuesta, derivada o simple.

El corpus estudiado está conformado por los siguientes términos españoles de la informática:⁴

antememoria, autodocumentado, autoedición, conversacional, depurador, didactical, digitalizar, filtrado, iconizar, indicador, infografía, interactivo, listado, logical, logicial, maxiordenador, micromizar, microordenador, multiproceso, multiprogramación, neuronal, ofimática, resaltado, robótica, teleinformática, telemantenimiento, tutorial.

Del análisis del anterior *corpus*, hemos distinguido las siguientes unidades morfológicas de la terminología española de la informática:

elementos léxicos pivotes: -memoria-, -document-, -edi-, -conversa-, -pura-, -didactic-, -digit-, -filtr-, -icón-, -indica-, -grafía-, -activo-, -list-, -logic-, -logici-, -ordena-, -microm-, -proceso-, -programa-, -neuron-, -ofi-, -resalt-, -robot-, -infor-, -manteni-, -tutori-.

prefijos: ante-, auto-, de-, info-, inter-, maxi-, micro-, multi-, tele-.

sufijos: -ado-, -al-, -cion-, -ción-, -dor-, -ica-, -izar-, -mática-, -miento-.

² Aquí, no tomamos en cuenta el caso de los interfijos por no considerarlos pertinentes en la terminología informática española.

³ Estos pseudo-afijos, de origen culto e introducción reciente en la lengua, no provienen de adverbios o preposiciones como los afijos tradicionales sino de adjetivos y sustantivos (por ejemplo, *auto-*, *info-*, *micro-*, *multi-*, *tele-* en nuestro corpus).

⁴ Este trabajo se integra dentro de un estudio más amplio sobre los problemas de traducción de los neologismos en lenguas de especialidad (Álvarez 1995).

2. Principios de formalización

2.1 El programa ZSTATION

La ZSTATION es un entorno de ingeniería lingüística que funciona como un taller de tareas lingüísticas que permite la definición y experimentación del conocimiento lingüístico. La formalización se apoya en el enfoque general sobre el cual está fundada la ZSTATION (Zinglé 1994). Así, la distinción metodológica establecida entre el nivel interlingüístico (*interlinguistic knowledge*) y el nivel intralingüístico (*intralinguistic knowledge*) nos permitirá disociar las características formales de la derivación y composición de las características semánticas. De esta manera, aunque las características formales del significante puedan variar de una lengua a otra, eso no afectará la transferencia interlingüística al ser efectuada esta última sobre la base del significado y no del significante. La formalización léxica se limitará entonces a explicitar las relaciones entre las características intralingüísticas e interlingüísticas.

En esa óptica, el objetivo del proceso de análisis radicará en asociar a una unidad léxica derivada o compuesta una representación semántica independiente de las lenguas y el proceso de generación consistirá a la inversa en expresar el contenido de una representación semántica en unidad léxica. Aquí, aparece justamente la cuestión del tipo de formalización semántica a elegir. Fundamentalmente, la ZSTATION se basa en la representación de los conocimientos que proporcionan los gráficos conceptuales (Sowa 1984) y que nos permiten expresar las características intrínsecas de los conceptos y las relaciones que los unen. Así, un gráfico conceptual básico se compone de dos conceptos unidos por una relación [Concepto 1]-»(Relación)-»[Concepto 2].⁵

Este programa se puede ver entonces como un sistema de traducción automática con un enfoque de tipo pivote que consta de dos fases análisis/generación y que está orientado hacia las redes semánticas utilizando los gráficos conceptuales como representación interlengua.

La ZSTATION funciona así a través de la modelización de los conocimientos lingüísticos gracias a 2 niveles:

- el nivel intralingüístico que concierne el análisis y la generación de las lenguas naturales. El lingüista tiene aquí la posibilidad de editar diccionarios y gramáticas que podrá utilizar después para el análisis y la creación del lenguaje natural.
- el nivel interlingüístico que permite editar el conocimiento lingüístico y hacer una compilación hacia redes semánticas o bases de datos para gráficos conceptuales.

Como ya hemos mencionado, esta distinción metodológica entre estos dos niveles es de gran importancia para nuestro trabajo.

⁵ Por ejemplo, el gráfico conceptual de *grabadora* sería [grabar]→(instrumento)→ [aparato] y se podría leer como *aparato que es el instrumento utilizado para grabar*.

2.2 Análisis y generación automáticos

La ZSTATION puede por lo tanto aportar una ayuda a la localización y a la creación de neologismos gracias a un sistema a bases de conocimientos. La idea desarrollada en el marco de las investigaciones del LILLA considera que los dos procesos (análisis/generación) pueden llevarse a cabo a partir de una sola y misma base de conocimientos en la cual están explicitados los conocimientos morfológicos (propiedades formales situadas en el nivel intralingüístico) y semánticos (propiedades conceptuales situadas en el nivel interlingüístico) sobre las unidades lingüísticas implicadas en la derivación o en la composición (Zinglé 1996).

El proceso de análisis aparece entonces como un cálculo que toma en cuenta esos conocimientos para lograr la construcción de una representación de naturaleza semántica. Se asocia pues a una unidad léxica derivada o compuesta una representación semántica independiente de las lenguas. El proceso inverso de generación aparece también como un cálculo que va a permitir derivar un término a partir de una representación semántica del mismo tipo.

3. Realización informática

El análisis y la generación automáticos de palabras derivadas constituyen unas de las aplicaciones posibles de la ZSTATION. Como todas las aplicaciones de este entorno, se nutren de la información contenida en un diccionario estructurado de unidades morfológicas, constituido mediante un editor de recursos lingüísticos. El análisis y la generación automáticos utilizan así la misma información fuente contenida en una base de conocimientos común.

3.1 Elaboración del diccionario de unidades morfológicas

El diccionario de unidades morfológicas está estructurado en artículos consagrados cada uno a un elemento formativo. Cada artículo está constituido por una o varias rúbricas, según que la unidad morfológica dé lugar a uno o varios análisis diferentes. Cada rúbrica está compuesta por 4 campos:

- <Campo de clase>
- <Campo semántico>
- <Campo morfosintáctico>
- <Campo estructural>

El *campo de clase* permite explicitar si el elemento formativo es un lexema, un prefijo o un sufijo.

El *campo semántico* se materializa a través de una fórmula que expresa la pila de conceptos objeto de la evaluación semántica. Los identificadores de conceptos son

elegidos libremente por el utilizador. Por razones de difusión, es recomendable utilizar identificadores en inglés. Podemos distinguir 3 casos:

- la pila se reduce a un concepto único: **memoria** «—» `memory0`
- la pila de conceptos está predefinida: **cálculo** «—» `result0(count0)`
- la pila de conceptos tiene asociada una pila argumento conformada por las condiciones de combinación del concepto y marcada formalmente por (*):
tele-«—» `remote0(*)`

Los dos primeros casos conciernen en regla general los elementos léxicos, el tercer caso es específico de los prefijos y sufijos.

El *campo morfosintáctico* sirve para determinar la categoría del elemento léxico formado. Teniendo en cuenta que la categoría del derivado es generalmente dada por el sufijo, este campo no tendrá información alguna para los prefijos y los lexemas. En estos casos, lo indicaremos de manera explícita con el identificador <nil>.

El *campo estructural* permite explicitar las restricciones formales y semánticas ligadas a las pilas de conceptos asociados a una pila argumento (principalmente los prefijos y sufijos). Así, cuando una misma unidad morfológica produce evaluaciones semánticas distintas, se pueden determinar las obligaciones semánticas que permiten diferenciarlos. Nuestro *corpus* de algunos términos españoles de la informática nos proporciona el ejemplo del sufijo **-al**, al cual podemos asociar la pila `relation0(*)` o `software0(*)`. En los dos casos, el elemento formativo argumento debe ser un lexema. En `relation0`, la pila argumento está validada si el lexema argumento está asociado al concepto de cualidad (`quality0`) mientras que en `software0(*)` el lexema sólo se asociará con el concepto de objeto (`object0`). Las restricciones semánticas aparecen notadas mediante el formalismo relacional adoptado para la elaboración de ontologías en la ZSTATION. Así, `conds=[r(l, isa, +quality0)]` indica que, en el cálculo estructural, la pila argumento se puede asociar con el concepto de cualidad. En el sub-campo `conds` puede figurar un conjunto de relaciones semánticas. La asociación del sufijo **-al** a la pila `relation0(*)` será así ligada a la condición `r(l, isa, +quality0)` y su asociación con la pila `software0(*)` a la condición `r(l, isa, +object0)`.

El diccionario de unidades morfológicas de los términos informáticos de nuestro corpus está conformado por los elementos formativos siguientes:

LEXEMA

activo {CLS=lex SEM=active0(*) CAT=nil arg(cls=nil,conds=[])	pura {CLS=lex SEM=bug0 CAT=nil arg(cls=nil,conds=[])	digit {CLS=lex SEM=digit0 CAT=nil arg(cls=nil,conds=[])
--	--	---

microm {CLS=lex
SEM=small0(*)
CAT=nil
arg(cls=nil,conds=[])

filtr {CLS=lex
SEM=match0(*)
CAT=nil
arg(cls=nil,conds=[])

indica {CLS=lex
SEM=prompt0(*)
CAT=nil
arg(cls=nil,conds=[])

logic {CLS=lex
SEM=logico
CAT=nil
arg(cls=nil,conds=[])

ordena {CLS=lex
SEM=compute0
CAT=nil
arg(cls=nil,conds=[])

edi {CLS=lex
SEM=publish0(*)
CAT=nil
arg(cls=nil,conds=[])

grafia {CLS=lex
SEM=graph0(*)
CAT=nil
arg(cls=nil,conds=[])

didactic {CLS=lex
SEM=educative0
CAT=nil
arg(cls=nil,conds=[])

neuron {CLS=lex
SEM=neuron0
CAT=nil
arg(cls=nil,conds=[])

robót {CLS=lex
SEM=robot0
CAT=nil
arg(cls=nil,conds=[])

list {CLS=lex
SEM=listO
CAT=nil
arg(cls=nil,conds=[])

icon {CLS=lex
SEM=icon0
CAT=nil
arg(cls=nil,conds=[])

logici {CLS=lex
SEM=logico
CAT=nil
arg(cls=nil,conds=[])

memoria {CLS=lex
SEM=memory0
CAT=nil
arg(cls=nil,conds=[])

tutori {CLS=lex
SEM=tutorO(*)
CAT=nil
arg(cls=nil,conds=[])

resalt {CLS=lex
SEM=bright0(*)
CAT=nil
arg(cls=nil,conds=[])

ofi {CLS=lex
SEM=office0
CAT=nil
arg(cls=nil,conds=[])

infor {CLS=lex
SEM=information0
CAT=nil
arg(cls=nil,conds=[])

conversa {CLS=lex
SEM=dialog0
CAT=nil
arg(cls=nil,conds=[])

document {CLS=lex
SEM=document0
CAT=nil
arg(cls=nil,conds=[])

manteni {CLS=lex
SEM=maintain0(*)
CAT=nil
arg(cls=nil,conds=[])

programa {CLS=lex
SEM=program0
CAT=nil
arg(cls=nil,conds=[])

proceso (CLS=lex
SEM=process0(*)
CAT=nil
arg(cls=nil,conds=[])

PREFIJOS

inter {CLS=pref
SEM=reciprocity0(*)
CAT=nil
arg(cls=nil,conds=[])

info {CLS=pref
SEM=dataprocessing0(*)
CAT=nil
arg(cls=nil,conds=[])

ante {CLS=pref
SEM=cache0(*)
CAT=nil
arg(cls=nil,conds=[])

multi {CLS=pref SEM=multiple0(*) CAT=nil arg(cls=nil,conds=[])	auto {CLS=pref SEM=automatic0(*) CAT=nil arg(cls=nil,conds=[])	tele {CLS=pref SEM=remote0(*) CAT=nil arg(cls=nil,conds=[])
maxi {CLS=pref SEM=powerfull0(*) CAT=nil arg(cls=nil,conds=[])	micro {CLS=pref SEM=small0(*) CAT=nil arg(cls=nil,conds=[])	de {CLS=pref SEM=contraryof0(*) CAT=nil arg(cls=nil,conds=[])

SUFIJOS

dad{CLS=suff
SEM=quality0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+quality0)])

ica{CLS=suff
SEM=science0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+object0)])

izar{CLS=suff
SEM=imitation0(*)
CAT=vb
arg(cls=lex,conds=[r(1,isa,+action0)])

al{CLS=suff
SEM=relation0(*)
CAT=adj
arg(cls=lex,conds=[r(1,isa,+quality0)])

}
{CLS=suff
SEM=software0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+object0)])

}
dor{CLS=suff
SEM=device0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+action0)])

ado{CLS=suff
SEM=result0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+action0)])

mática{CLS=suff
SEM=automatic0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+quality0)])

}
{CLS=suff
SEM=dataprocessing0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+object0)])

}
ción{CLS=suff
SEM=action0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+action0)])

cion{CLS=suff
SEM=action0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+action0)])

miento{CLS=suff
SEM=action0(*)
CAT=sub
arg(cls=lex,conds=[r(1,isa,+action0)])

Indicamos a continuación un ejemplo de edición de artículo con la ayuda del editor de recursos específicos de la ZSTATION:

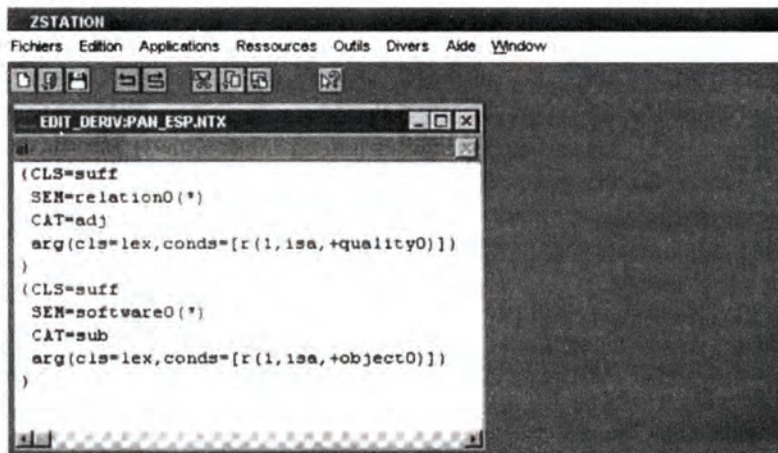


Figura 1

La elaboración de ontologías aparece entonces como indispensable para la toma en cuenta de las restricciones semánticas. Básicamente, una ontología está representada por una entramada red de conceptos de tipo jerárquico multidimensional. Cada concepto al interior de esta red está relacionado con sus hiperónimos por relaciones formalizadas, definidas por el ingeniero lingüista quien las elige en función de los conocimientos que se tengan que modelizar. Las propiedades conceptuales están por lo tanto formalizadas mediante redes multidimensionales a herencia múltiple donde cada nudo de la red está caracterizado por un conjunto de relaciones conceptuales lo que permite conocer a todo momento el conjunto de propiedades de un concepto dado o de verificar si está en medida de satisfacer una restricción semántica cualquiera.

Estas relaciones están expresadas con el relator r **Valor, Relación, Concepto*+donde *Valor* indica si se trata de una relación obligatoria (*Valor*= 1) o facultativa (*Valor*=0); *Relación* representa el identificador de relación que el utilizador ha seleccionado (isa, partOf, cont, char, etcétera) y *Concepto* un identificador de concepto precedido de + ó -, según su aserción o refutación. La refutación interviene esencialmente cuando un concepto está asociado a un hiperónimo del cual no hereda todas sus propiedades. El mecanismo utilizado es un mecanismo de multiherencia transitiva que incluye la propagación de la negación sobre los hipónimos; se da la prioridad a la negación cuando un concepto hereda de sus hiperónimos propiedades contradictorias.⁶ La transitividad permite obtener las propieda-

⁶ ""Por ejemplo, el pingüino (hipónimo) es un pájaro (hiperónimo) que no tiene la capacidad de volar.

des eventualmente heredadas por el hipónimo de un concepto. Ese doble mecanismo de multiherencia transitiva nos permite conocer el conjunto de propiedades de un concepto o bien saber si un concepto responde a una propiedad dada. Fundado en la programación orientada a objetos, el resultado está obtenido en los dos casos por el envío de un mensaje al concepto implicado. Evidentemente, esto implica que sea necesaria una descripción cuidadosa de los diferentes conceptos para poder obtener una validación eficiente de las restricciones semánticas.

El corpus estudiado nos ha permitido así elaborar la ontología necesaria, conformada por la lista de conceptos siguientes:

CONCEPTOS	CONDICIONES	LEXEMAS
bright0	r(0,isa,+action0)	resalt
bug0	r(0,isa,+action0)	pura
compute0	r(0,isa,+action0)	ordena
dialog0	r(0,isa,+action0)	conversa
digito	r(1,isa,+qualityO)	digit
documento	r(0,isa,+action0)	document
educative0	r(1,isa,+objectO)	didactic
icon0	r(0,isa,+action0)	icon
information0	r(1,isa,+qualityO)	infor
list0	r(0,isa,+action0)	list
logic0	r(1,isa,+objectO)	logic
logici0	r(1,isa,+objectO)	logici
maintain0	r(1,isa,+actionO)	manteni
match0	r(0,isa,+action0)	filtr
neuron0	r(1,isa,+qualityO)	neuron
office0	r(1,isa,+objectO)	ofi
program0	r(0,isa,+action0)	programa
prompt0	r(0,isa,+action0)	indica
publish0	r(0,isa,+action0)	edi
robot0	r(0,isa,+object0)	robót
small0	r(0,isa,+action0)	microm
tutor0	r(1,isa,+objectO)	tutori

A continuación, proporcionamos un ejemplo de modelización informática correspondiente a la definición de concepto:

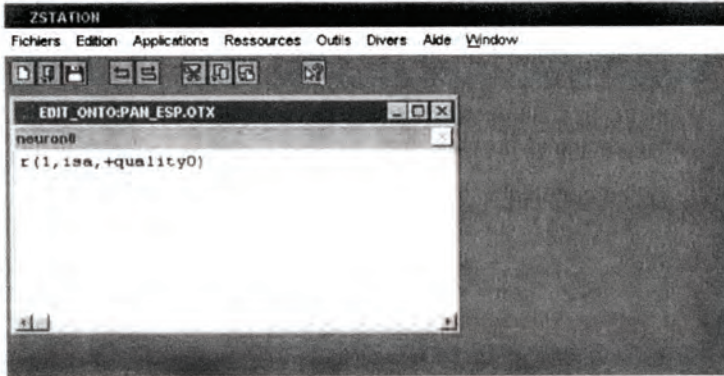


Figura 2

3.2 *Análisis y generación automáticos*

Los procesos de análisis y generación automáticos han sido implantados en PDC-Prolog (versión Windows 4.1). Son fundamentalmente independientes de las lenguas naturales puesto que las estrategias de procesamiento de cada lengua están codificadas en la base de datos morfológicos. Se ha tratado de favorecer en lo máximo la interactividad para permitir al ingeniero lingüista pasar fácilmente de la formalización de los datos lingüísticos al análisis o a la generación.

3.2.1 *Análisis automático de términos derivados*

El interfaz de análisis se presenta como una ventana de diálogo activada a partir del menú de la ZSTATION (Aplicaciones/Análisis/Análisis léxico) y que contiene:

- una zona de toma de datos (para la adquisición del término a analizar)
- una zona de visualización del resultado del análisis
- un botón para activar el análisis después de la toma de datos
- un botón para salir del interfaz.

El proceso de análisis utiliza por una parte un diccionario compilado de unidades morfológicas (recurso intralingüístico) y una ontología compilada (recurso interlingüístico) y se desarrolla mediante las siguientes fases:

- localización de las unidades morfológicas dentro del término a analizar
- construcción de hipótesis sobre la base de las unidades morfológicas localizadas
- validación de las hipótesis (en particular de las restricciones semánticas).

El análisis automático se sirve por lo tanto de un generador de hipótesis que permite construir una jerarquización de las unidades morfológicas detectadas. Para cada construcción donde están implicadas dos unidades morfológicas, se verifica si las restricciones formales y semánticas están satisfechas, en función de la combinación a izquierda o a derecha.

Mostramos a continuación una muestra del interfaz de análisis con el resultado obtenido para el término didactical:

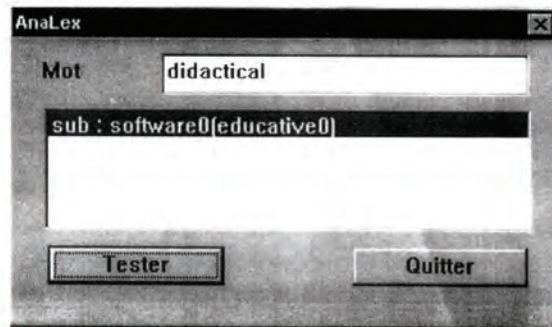


Figura 3

3.2.2 Generación automática de términos derivados

El interfaz de generación se presenta también bajo la forma de una ventana de diálogo activada desde el menú general de la ZSTATION (Aplicaciones/Generación/Generación léxica). Utiliza los mismos recursos que el análisis y contiene:

- una zona de toma de datos (para la adquisición de la descripción semántica del término a crear)
- una lista de edición para la selección de la categoría morfosintáctica del término a crear (este control de diálogo sirve a la vez para la visualización de la lista de los identificadores morfosintácticos identificados en el diccionario de unidades morfológicas y para la selección de la categoría deseada)
- una lista de conceptos identificados en el diccionario de unidades morfológicas
- una zona de visualización del resultado de la generación
- un botón para activar la generación después de la toma de datos
- un botón para salir del interfaz.

Este proceso de generación consta de las siguientes fases:

- localización de los conceptos utilizados en la pila de conceptos que describe la información que debe expresar el término a crear
- búsqueda de las unidades morfológicas que se pueden asociar a los conceptos localizados

- construcción de hipótesis sobre la base de los conceptos localizados y de las unidades morfológicas asociadas
- validación de las hipótesis.

Los dos ejemplos siguientes de generación corresponden al adjetivo *neuronal* y al sustantivo *tutorial*:

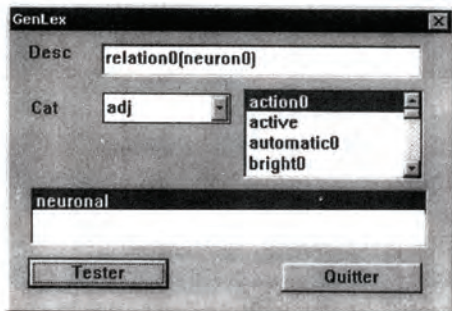


Figura 4

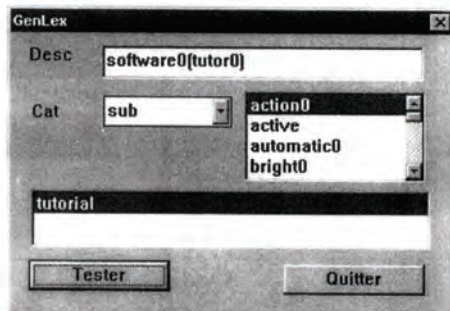


Figura 5

El análisis automático de los neologismos de la terminología informática española contenidos en nuestro corpus nos permiten obtener las descripciones semánticas siguientes:

antememoria:	nil: cacheO(memoryO)
autodocumentado	sub: automaticO(resultO(documentO))
autoedición	sub: automaticO(actionO(publishO))
conversacional	adj: relationO(actionO(dialogO))
depurador	sub: contraryofO(deviceO(computeO))
didactical	sub: softwareO(educativeO)
digitalizar	vb: imitationO(digitO)
filtrado	sub: resultO(matchO)
iconizar	vb: imitationO(relationO(iconO))
indicador	sub: deviceO(promptO)
infografía	nil: dataprocessingO(graphO)
interactivo	nil: reciprocityO(activeO)
listado	sub: resultO(listO)
logical	sub: softwareO(logicO)
logicial	sub: softwareO(logicO)
maxiordenador	sub: powerfillO(deviceO(computeO))

micromizar	vb: imitation0(small0)
microordenador	sub: small0(device0(compute0))
multiproceso	nil: multiple0(process0)
multiprogramación	sub: multiple0(action0(program0))
neuronal	adj : relation0(neuron0)
ofimática	sub: dataprocessing0(office0)
resaltado	sub: result0(bright0)
robótica	sub: science0(robot0)
teleinformática	sub: remote0(automatic0(information0))
telemantenimiento	nil: remote0(action0(maintain0))
tutorial	sub: software0(tutor0)

En lo referente a la determinación de la categoría morfosintáctica de los derivados, podemos mencionar que la asociación de un preñjo y un lexema (**antememoria** por ejemplo) provoca la indeterminación de la categoría morfosintáctica del término así formado (marcada por el identificador <nil>). En los otros casos, el sufijo determina la categoría del derivado.

El proceso de generación automática, al desarrollarse de manera inversa y con las mismas informaciones que el proceso de análisis, se llevará a cabo a partir de las descripciones semánticas que permitirán así generar el término deseado.

El programa permite evidentemente (es uno de sus objetivos) generar o analizar términos inexistentes a partir de la activación de una misma base de conocimientos. Así, tendremos los ejemplos siguientes:

teledepuración	sub:remote0(contraryof0(action0(bug0)))
telemultiprogramacional	adj: remote0(multiple0(relation0(action0(program0))))
maxiprogramador	sub: powerfull0(device0(program0))

4. Perspectivas

Lo que se propone aquí es, por lo tanto, un sistema de análisis y generación automáticos de términos derivados y compuestos que utiliza una base de conocimientos en la cual están explicitadas las restricciones semánticas y formales de los elementos formativos gracias a un método de representación del sentido basado en la teoría de los gráficos conceptuales. Este enfoque se puede adaptar a lenguas diferentes a condición de construir las bases de conocimientos necesarias. Para eso, siguiendo la óptica multilingüe de la Red Panlatina de Terminología, se han elaborado en el LILLA diccionarios de unidades morfológicas relativamente completos del francés, del español, del portugués y del italiano. Además, se están

llevando a cabo estudios para la inserción del análisis de los neologismos en un sistema de vigilancia terminológica. En efecto, además de la modelización lingüística y de sus aplicaciones evidentes en la enseñanza de la lexicología, los procedimientos desarrollados abren perspectivas interesantes en el acondicionamiento terminológico, bien para automatizar la vigilancia terminológica, bien para crear neologismos oficiales. En ese aspecto, el sistema presentado puede resultar muy útil para el terminólogo. El análisis de la literatura científica y técnica evidencia el hecho de que muchos utilizadores generan ellos mismos los nuevos términos que necesitan. Por esa razón, convendría verificar la conformidad de esos neologismos desde un punto de vista estrictamente lingüístico para poderlos consignar y validar en su campo de especialidad. Éste sería el paso previo necesario a la introducción de estos términos dentro de una terminología oficial pdr crear o revisar. Además, al ser capaz de emitir una interpretación semántica, este sistema de vigilancia terminológica va a permitir también evaluar la relevancia del nuevo término antes de iniciar su validación por parte de los expertos de la especialidad. Por su parte, la generación interviene principalmente cuando un término existe ya en una lengua —en la mayoría de los casos, en inglés—y se necesita proponer rápidamente un equivalente en otra lengua. El término creado deberá ser no sólo conforme a las características morfológicas y ortográficas de la lengua sino también aceptable semánticamente. El enfoque propuesto representa también por lo tanto una herramienta muy valiosa para el traductor de textos especializados confrontado a menudo con la inexistencia de correspondiente para un neologismo técnico ya que permite crear un neologismo en una lengua meta a partir del análisis de un término en lengua fuente por lo que responde a una preocupación no sólo de los numerosos especialistas del acondicionamiento terminológico sino también de los traductores en el ámbito de las lenguas de especialidad.

Bibliografía

- ÁLVAREZ, A. (1995) *Terminologie informatique et problèmes de néologie en espagnol*. Tesis de Maestría, LILLA, Universidad de Niza-Sophia Antipolis, Francia.
- SOW A, J. (1984) *Conceptual structures: information processing in man and machine*. Addison-Wesley.
- ZINGLÉ, H. (1994) “The ZStation workbench and the modelling of linguistic knowledge”. En: *Current issues in mathematical linguistics*. Elsevier-NHLS, 423-432.
- ZINGLÉ, H. (1996) “Analyse et génération automatiques de mots dérivés en français”. *Travaux du LILLA*. Francia: Universidad de Niza-Sophia Antipolis, 27-42.