INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Generador de los grafos conceptuales a partir del texto en español

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

MACARIO HERNÁNDEZ CRUZ

Director:

Dr. Alexander Gelbukh

México, D.F.
Mayo, 2007

SIP-14

INSTITUTO POLITECNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 16:00 horas del día 30 del mes de Mayo de 2007 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"GENERADOR DE LOS GRAFOS CONCEPTUALES A PARTIR DEL TEXTO EN ESPAÑOL"

HERNÁNDEZ	CRUZ	MACARIO
Apellido paterno	materno	nombre(s)

Con registro: B 0 1 1 3 8 4

aspirante al grado de: **MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente

Dr.Igor Bolshakov

Secretario

Dr. Sergio Suárez Guerra

Primer vocal
(Director de tesis)

Dr. Alexandre Guelboukh Kahn

Segundo vocal

Dra. Sofía Natalia Galicia Haro

Tercer Vocal

M. en C. Miguel Jesús Torres Ruiz

EL PRESIDENTE DEL COLEGIO

Dr. Hugo César Coyote Estrada

DIRECCION

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la Ciudad de MEXICO el día 30 del mes MAYO del año 2007, el que suscribe MACARIO HERNANDEZ CRUZ alumno del Programa de MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN con número de registro B011384, adscrito al CENTRO DE INVESTIGACIÓN EN COMPUTACION, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del DR. ALEXANDRE GUELBOUKH KAHN y cede los derechos del trabajo intitulado GENERADOR DE LOS GRAFOS CONCEPTUALES A PARTIR DEL TEXTO EN ESPAÑOL, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección mahernandezc@ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

MACARIO HERNANDEZ CRUZ

Nombre y firma

Resumen

Se presenta un sistema que convierte un texto libre en español en una representación semántica formal, a saber, un conjunto de los llamados grafos conceptuales. Un grafo conceptual representa, básicamente, una red de predicados y sus argumentos, que describe ciertos hechos sobre el universo del discurso; en este caso los hechos comunicados en el texto analizado. La estructura semántica formal así obtenida tiene numerosas aplicaciones en las tareas computacionales relacionados con el texto: la recuperación de información, la respuesta a preguntas, la minería de texto, el agrupamiento de documentos, la traducción automática, entre otras (estas aplicaciones están fuera del alcance de la presente tesis; algunas de éstas fueron objeto de otras tesis de Maestría y Doctorado realizadas por los integrantes del mismo grupo). A pesar de la gran utilidad de un método para la obtención automática de los grafos conceptuales de texto en español, según nuestro conocimiento antes del presente trabajo eso no fue posible; las aplicaciones conocidas de los grafos conceptuales trabajaban sólo sobre las bases de grafos construidos manualmente, lo cual fue un proceso muy costoso y poco confiable.

La conversión que realiza el presente sistema sigue la metodología clásica de procesamiento de texto: el texto libre pasa por un analizador morfológico y luego por un analizador sintáctico, ambos desarrollados previamente por los integrantes del mismo grupo. La salida del analizador sintáctico es un conjunto de árboles sintácticos. El sistema efectúe las siguientes operaciones sobre esta salida: 1) identificación de los tipos semánticos de los nodos (palabras o frases) y sobre todo sus relaciones; éstos se convierten en los elementos correspondientes de la estructura resultante; 2) operaciones necesarias sobre los grafos obtenidos, que garanticen que éstos cumplan con las restricciones del formalismo de los grafos conceptuales; 3) generación de la estructura resultante en el formato especificado por los estándares internacionales correspondientes para el formalismo de los grafos conceptuales.

Como un ejemplo de la aplicación del sistema, los grafos generados por el sistema se usaron como entrada a un sistema de minería de texto existente. Este sistema, desarrollado en el marco de una tesis doctoral, no se ha operado sobre los textos reales ya que requería de la construcción manual de sus grafos de entrada; con el presente trabajo se hace posible su explotación masiva sobre textos abiertos no preparados sin intervención manual. Como trabajo futuro, el sistema abre al camino al desarrollo de otros sistemas aplicados mencionados arriba que funcionarán en base de la representación semántica de texto y no sólo de su representación estadística como los sistemas existentes.

Abstract

A system is presented capable of converting open plain text in Spanish into a formal semantic representation, namely, a set of so-called conceptual graphs. A conceptual graph represents, basically, a network of predicates and their arguments that describes certain facts about the universe of the discourse; in our case, the facts communicated in the text being analyzed. The obtained semantic structure has numerous applications in text-related computational tasks such as information retrieval, question answering, text mining, document clustering, and machine translation, to mention a few (these applications are beyond the scope of the present dissertation; some of them have been the object of other MSc or PhD dissertations by the members of the same team). In spite of grate usefulness of a method for automatically obtaining conceptual graphs from a plain text in Spanish, to the best of our knowledge there this was not possible previously; existing applications that rely on conceptual graphs are only used over databases of manually constructed conceptual graphs, which implies a highly expensive and error-prone process.

The conversion process implemented in our system follows the classic methodology of text processing: the plain text is processed with a morphological analyzer and then a parser, both previously developed by members of the same team. The output of the parser is a set of syntactic trees. The system performs the following operations on this output: (1) identification of semantic types of the nodes (words or phrases) and most importantly their relations; these are converted into the corresponding elements of the resulting structure; (2) transformations of the obtained graphs that guarantee their compliance with the constraints of the formalism of the conceptual graphs; (3) generation of the resulting structure in the format specified by the corresponding international standards on conceptual graph formalism.

As an example of application of the system, the conceptual graphs generated by the system were used as the input to an existing text mining system. This system, developed in frame of a PhD thesis, was not previously applied to real texts since it required manual construction of the graphs used as input; the present work makes it possible to apply the system to raw open texts with no manual intervention. As a future work, the system opens the way to development of other applied systems mentioned above, which will rely on semantic representation of text and not just statistical representation as existing systems do.

Agradecimientos

A mi director de tesis, el *Dr. Alexander Gelbukh*, por su apoyo incondicional y su generosidad.

A mis profesores del Laboratorio de Procesamiento de Lenguaje Natural, los Doctores *Grigori Sidorov* e *Igor Bolshakov,* por sus múltiples consejos.

A mis profesores, los doctores *Leonid Cheremetov*, *Álvaro de Albornoz*, *Matías Alvarado*, *Ricardo Barrón y Pablo Manrique* por sus enseñanzas.

A los Doctores *Sergio Suárez Guerra*, *Sofía Galicia Haro*, *Manuel Montes*, y *Miguel Torres Ruiz*, por sus valiosos comentarios y sugerencias.

A mis compañeros *Hugo Jiménez*, *Marcelo Rodríguez*, *Flavio Sánchez* e *Yventz Entzana*, por su amistad y solidaridad.

Al *Consejo Nacional de Ciencia y Tecnología* (CONACYT) y *al Programa Institucional de Formación de Investigadores* (PIFI) del IPN por el apoyo económico que me permitió estudiar la Maestría.

Al *Centro de Investigación en Computación* por darme la oportunidad de realizar mis estudios en la mejor institución de computación del país.

A *Gabriel Durán* por su respaldo, apoyo y sobre todo por su valiosa amistad.

Finalmente, a mi familia, mi *esposa* y mis *hijos*, por su gran cariño y paciencia.

ÍNDICE GENERAL

ÍNDICE DETALLADO

I. Introducción

1.1 Motivación

El lenguaje es el medio de comunicación más eficaz de que dispone la humanidad, utilizado de diversas maneras, sirve para expresar sentimientos y emociones, explicar conceptos e ideas, entablar negocios, narrar historias y como medio de transmisión de la cultura. El lenguaje es una parte fundamental en la vida de todas las personas. Asimismo, la presencia de computadoras es cada vez más frecuente en el desarrollo de las actividades cotidianas. La interacción con máquinas capaces de comprender el lenguaje humano ha dejado de ser un tema de ciencia ficción.

La ciencia que se encarga de estudiar el lenguaje humano y establecer modelos formales que permitan su análisis, es la lingüística. La formulación de modelos computacionales del lenguaje natural que permitan a las máquinas comprender el lenguaje humano es la tarea de la *lingüística computacional*. Para llevar a cabo su labor, la lingüística computacional ha establecido varios niveles de análisis, a saber: morfológico, sintáctico, semántico y pragmático. El nivel morfológico tiene como objeto establecer las categorías gramaticales de las palabras que conforman las oraciones; la tarea principal del nivel sintáctico es describir las relaciones entre las palabras de una oración y la función que cada palabra realiza dentro de esta; el nivel semántico estudia el significado de palabras y oraciones, generando una representación formal del conocimiento contenido en los textos; en el nivel pragmático la estructura de representación obtenida se interpreta para determinar su significado real y puntual dentro del contexto específico.

El presente trabajo está enmarcado dentro del nivel semántico y dedicado a la generación de estructuras de representación del conocimiento semántico de las oraciones de textos en español. Las estructuras de representación están basadas en el formalismo de los grafos conceptuales des-

arrollado por John F. Sowa, los cuales son una forma de representación de conocimiento basado en la lingüística, la psicología y la filosofía. Mediante los grafos conceptuales se representan los conceptos de una oración y las relaciones entre estos.

1.2 Descripción del problema

Existen diversas formas de representar el conocimiento contenido en textos, entre otros se puede mencionar la representación basada en marcos, reglas, lógica de primer orden y redes semánticas. Los grafos conceptuales son un tipo de red semántica, donde sólo existen dos tipos de nodos: los nodos concepto y los nodos relación.

Los grafos conceptuales tienen el potencial de representar de forma simple y directa los detalles finos del lenguaje. Por ejemplo, permiten representar dependencias contextuales que no pueden se representadas por predicados lógicos.

El proceso de generación de los grafos conceptuales a partir del texto en español está dirigido por la sintaxis, teniendo como entrada las estructuras de representación sintáctica proporcionadas por un analizador sintáctico (parser), identificando los roles de las palabras y establecer así el tipo de relación conceptual entre ellas.

1.3 Objetivo general

Desarrollar un convertidor de árboles sintácticos a grafos conceptuales, de tal manera que puedan ser aprovechados por herramientas que utilicen grafos conceptuales como entrada.

1. 4 Aportaciones de la tesis

Esta tesis es un paso adelante en la cadena de las herramientas de análisis de texto que se desarrollan en el Laboratorio de Lenguaje Natural del Centro de Investigación en Computación.

Los antecedentes de esta tesis son:

- Un analizador sintáctico desarrollado en el Laboratorio que convierte un texto en un conjunto de árboles sintácticos, y
- Un sistema de minería de texto cuya entrada es un conjunto de grafos conceptuales que representan el texto bajo análisis (hasta ahora principalmente se probó con los textos en inglés para los cuales se dispone de un programa que construye dichos grafos).

Las **aportaciones** de esta tesis son:

- Se desarrolló un **convertidor** automático de las estructuras sintácticas producidas por el analizador mencionado a los grafos conceptuales que pueden ser entrada al sistema mencionado.
- Se demostró que este convertidor sirve como el **vínculo** anteriormente faltante para la implementación de la cadena completa de procesamiento automático de texto con el sistema de minería de texto mencionado.
- Se **abrió camino** para el desarrollo de otras herramientas que usen la representación simbólica semántica del texto (y no puramente estadística), tales como sistemas de recuperación de información, respuesta a preguntas, etc., los cuales serán desarrollados, o están en desarrollo actualmente, en el Laboratorio de Lenguaje Natural del Centro de Investigación en Computación.
- El convertidor desarrollado es **desacoplado** de un analizador sintáctico específico, y aprovechará del futuro desarrollo de la tecnología de análisis sintáctico, mejorando así los resultados obtenidos cada vez cuando esté disponible una nueva versión del analizador. Eso tam-

bién permite la aplicación del sistema a diversos dominios específicos distintos de los textos con los cuales hemos hecho nuestros experimentos, siempre y cuando esté disponible, o se desarrolle, un analizador sintáctico para este dominio.

– El método desarrollado se basa en las gramáticas de **dependencias** (no constituyentes) –el tipo de gramáticas que atrae cada vez más atención y permite el desarrollo de mejores y más completos analizadores sintácticos (con los cuales será compatible nuestro convertidor).

1.5 Organización de la tesis

El resto de este documento se organiza de la manera siguiente:

– En el capítulo 2 se aborda brevemente el estado del arte.
– El capítulo 3 está dedicado a la descripción de las herramientas necesarias para la instrumentación del programa propuesto.
– El capítulo 4 introduce al usuario al formalismo de los grafos conceptuales.
– En el capítulo 5 se detalla el procedimiento de transformación de los árboles sintácticos en gráficos conceptuales.
– En el capítulo 6 se describe brevemente la implementación del programa.
– En el capitulo 7 se discuten los resultados obtenidos.
– El capítulo 8 está dedicado a las conclusiones y trabajo futuro.

II Estado del arte

2.1 Ubicación

El ser humano posee características que lo distinguen del resto de las especies, una de estas características, y quizá una de las más valiosas, es el dominio del lenguaje. Asimismo, uno de los instrumentos que incrementa su presencia conforme pasa el tiempo, es la computadora. La lingüística computacional tiene la tarea de formular modelos computacionales del lenguaje natural, así como lograr que las computadoras comprendan la información que manipulan y permitan la interacción mediante la comunicación en lenguaje humano.

2.1.1 El lenguaje y la lengua

Ferdinand de Saussure hace una clara diferenciación entre los conceptos "lengua " y "lenguaje", este autor considera al lenguaje como una *totalidad* que tiene dos componentes: la *lengua* y el *habla*.

"El estudio del lenguaje entraña, por tanto, dos partes: una esencial, tiene por objeto la lengua, que es social en su esencia e independiente del individuo; este estudio es únicamente psíquico; la otra, secundaria, tiene por objeto la parte individual del lenguaje, es decir, el habla con la fonación incluida; esta parte es psíquico-física"[1]

[1] Saussure Ferdinand, *Curso de lingüística general,* Fontamara, México,1980. pp. 45-51

Sin embargo estos dos componentes del lenguaje están estrechamente vinculados y son recíprocos: la lengua es necesaria para que el habla sea inteligible y produzca todos sus efectos, así como el habla es necesaria para que la lengua se establezca.

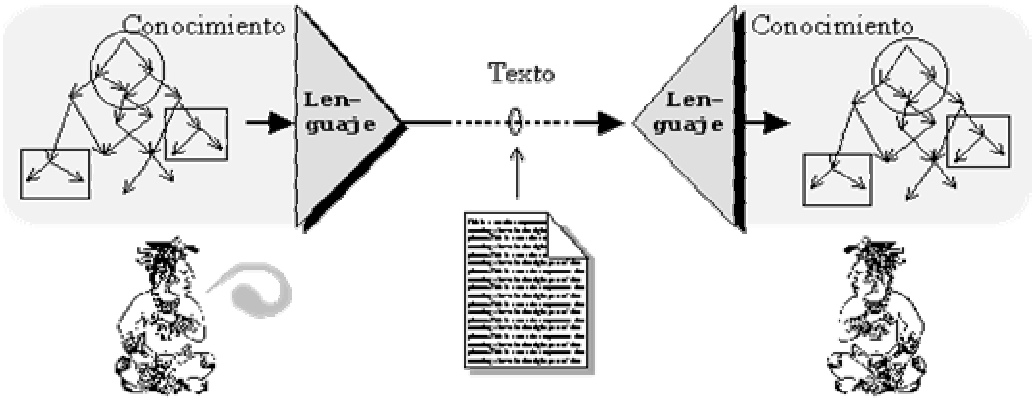


Fig. 2.1 El lenguaje es un codificador-descodificador[2]

2.1.2 La lingüística general

La lingüística es la ciencia que estudia los lenguajes naturales[3], para ser más precisos, abarca un amplio conjunto de diferentes ciencias relacionadas.

La lingüística general estudia la estructura general de varios lenguajes naturales y descubre las leyes universales de su funcionamiento. La lingüística general es una ciencia fundamental, desarrollada por muchos investigadores durante los últimos dos siglos y está basada en gran parte en los métodos y resultados de los gramáticos antiguos.

[2] Tomado de Gelbukh A., *Avances en procesamiento de lenguaje natural*.

Las partes más importantes de la lingüística general son:

Fonología, estudia los sistemas fónicos de las lenguas, trata con los sonidos que componen el habla.
Morfología, estudia la estructura interna de las palabras individuales y las leyes concernientes a la formación de nuevas palabras.

Sintaxis, considera la estructura de las oraciones y las formas cómo las palabras individuales están conectadas entre sí.

Semántica y pragmática, estas están estrechamente relacionadas. La semántica trata con el significado de las palabras individuales y textos enteros, y la pragmática estudia las motivaciones de la gente para producir textos u oraciones específicas.

Hay muchas otros componentes especializados de la lingüística como un todo (ver fig. 2.2).

Lingüística histórica o comparativa, estudia la historia de los lenguajes a través de la comparación entre ellos, por ejemplo, estudiando la historia de sus similaridades y diferencias. El segundo nombre es explicado por el hecho de que la comparación es el método principal en esta rama de la lingüística. La lingüística comparativa es incluso más antigua que la lingüística general, se originó en el siglo XVIII.

[3] Cfr. Gelbukh A. Bolsakov. I, *Computational Linguistics*. IPN, 2004, pp. 17-26.

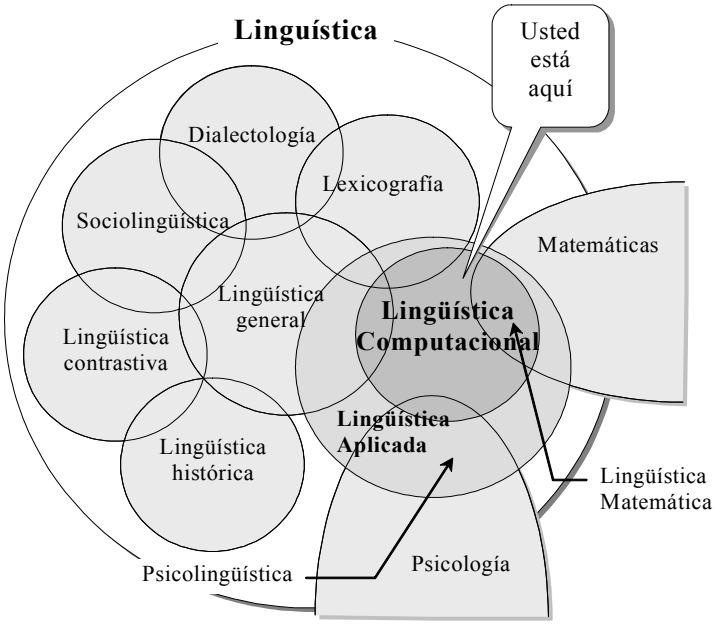


Fig. 2.2 Estructura de la ciencia de la lingüística.[4]

Varias de las nociones de la lingüística general fueron adoptadas directamente de la lingüística comparativa.

Desde los tiempos de Ferdinand Saussure, la historia de los lenguajes ha sido llamada *diacronía* del lenguaje, en oposición a la *sincronía* del lenguaje que trata con los fenómenos de los lenguajes modernos únicamente.

Lingüística contrastiva o lingüística tipológica, clasifica una variedad de lenguajes de acuerdo a la similaridad de sus características sin interesarse en el origen de los lenguajes.

La *sociolingüística* describe las variaciones de un lenguaje a través de la escala social. Es bien conocido que varios estratos sociales utilizan frecuentemente sublenguajes dentro de un lengua-

[4] *ibid*. (reproducido con autorizaciòn).

je común, toda vez que una misma persona utiliza diferentes sublenguajes en diferentes situaciones.

La *dialectología* compara y describe los varios dialectos o sublenguajes de un lenguaje común, el cual es usado en diferentes áreas de un territorio donde algún lenguaje es usado oficialmente. Por ejemplo, en diferentes países de habla hispana, muchas palabras, combinaciones de palabras o incluso formas gramaticales son usadas diferentemente, sin mencionar las significativas diferencias en la pronunciación.

La *lexicografía* estudia el léxico o el conjunto de todas las palabras de un lenguaje específico, con sus significados, características gramaticales, pronunciación, etc, así como los métodos de compilación de varios diccionarios basados en dicho conocimiento.

La *psicolingüística* estudia comportamiento del lenguaje de los seres humanos a través del significado de una serie de experimentos de tipo psicológico. Entre las áreas de especial interés, la psicolingüística estudia la enseñanza del lenguaje a los niños, enlaza la habilidad de lenguaje en general y el arte del habla, así como otras características psicológicas conectadas con el lenguaje natural y lo expresado a través de él.

Lingüística matemática. Hay dos diferentes vistas en la lingüística matemática. En la vista más estrecha, el término lingüística matemática es usado por la teoría de las gramáticas formales de un tipo especial llamadas *gramáticas generativas*. Esta es una de las primeras teorías puramente matemáticas dedicadas al lenguaje natural. Alternativamente, en la vista más amplia, la lingüística matemática es una intersección entre las matemáticas y la lingüística, por ejemplo, la parte matemática que toma el fenómeno lingüístico y las relaciones entre ellos como objetos de su posible aplicación e interpretación.

La lingüística aplicada desarrolla los métodos para la aplicación de las ideas y nociones de la lingüística general en la práctica humana. Hasta mediados del siglo XX, las aplicaciones de la lingüística estaban limitadas al desarrollo y mejoramiento de gramáticas y diccionarios impresos orientados al uso extensivo por no especialistas, así como los métodos racionales para la enseñanza de los lenguajes naturales, su ortografía y estilo. Este fue sólo un producto puramente práctico de la lingüística.

En la segunda mitad del siglo XX surge una nueva rama de la lingüística aplicada llamada *lingüística computacional* o *ingeniería lingüística*.

2.1.3 La lingüística computacional

La *lingüística computacional* es el estudio de los sistemas computacionales para comprender y generar el lenguaje natural[5]. Aunque los objetivos de investigación de la lingüística computacional son ampliamente variados, la motivación primaria ha sido siempre el desarrollo de sistemas específicos prácticos que involucran lenguaje natural. Existen tres clases de aplicaciones que han sido centrales en el desarrollo de la lingüística computacional:

Traducción automática. En los últimos años, la calidad de la traducción automática ha mejorado drásticamente. En el caso ideal, el traducir un texto consiste en entender este texto –en el sentido de transformarlo en una representación formal– y luego generar el texto, según el sentido entendido, en el otro idioma.

Actualmente, por lo general no es posible entender todo el texto, con todas las relaciones entre los conceptos mencionados en él. Entonces, los traductores automáticos entienden algunas partes

[5] Cfr. Grihsman R, *Computacional Linguistics*, Cambrigde University Press, 1986.

del texto, más grandes o más pequeñas, y las traducen en el orden en que aparecen en el texto fuente.

En muchos casos este no es suficiente. Por ejemplo, para traducir oraciones como:

John took a cake from the table and ate it.
John took a cake from the table and cleaned it.

Se necesita realmente entender qué hizo John: tomó un pastel de la mesa y ¿*lo comió* o *la comió*? ¿*lo limpió* o *la limpió*? Al revés, para traducir el texto *Juan le dio a María un pastel. Lo comió*, hay que elegir entre las variantes *He ate it, She ate it, It ate him*, *She ate him*, etc.

Búsqueda y recuperación de información. La aplicación del procesamiento de lenguaje natural más obvia y quizá más importante en el momento actual es la búsqueda de información (se llama también recuperación de información). Por un lado, en Internet y en las bibliotecas digitales se contiene una cantidad enorme de conocimiento que puede dar respuestas a muchísimas preguntas que tenemos. Por otro lado, hay mucha información que no sirve porque ya no se puede encontrarla. Hoy en día la pregunta ya no es "¿sí se sabe cómo...?" sino "ciertamente se sabe, pero ¿dónde se halla esta información?".

Técnicamente, rara vez se trata de decidir cuáles documentos son relevantes para la petición del usuario y cuáles no. Usualmente, una cantidad enorme de documentos se puede considerar como relevantes en cierto grado, siendo unos más relevantes y otros menos. Entonces, la tarea se entiende como medir el grado de esta relevancia para proporcionar al usuario primero el documento más relevante; si no le sirvió, el segundo más relevante, etc.

El problema más difícil de la recuperación de información es, sin embargo, no de índole técnica sino psicológica: entender cuál es la necesidad real del usuario, para qué formula su pregunta. Este problema se complica ya que no existe un lenguaje formal en el cual el usuario podría formular claramente su necesidad.

Las técnicas más usadas actualmente para la recuperación de información involucran la búsqueda por palabras clave: se buscan los archivos que contengan las palabras que el usuario teclee. Es decir, la representación formal usada es el conjunto de las cadenas de letras (palabras), usualmente junto con sus frecuencias en el texto (en número de ocurrencias). La claridad matemática de la tarea causó mucho avance en la teoría de estos métodos. Las ideas más usadas son los modelos probabilísticos y los procedimientos iterativos e interactivos: tratar de adivinar qué necesita el usuario preguntándole cuáles documentos le sirven.

Sin embargo, los métodos que involucran sólo las palabras (como cadenas de letras) pero no el sentido del texto son muy limitados en su capacidad de satisfacer la necesidad informática del usuario, es decir, de hallar la respuesta a la pregunta que tiene en mente. Se puede mejorar mucho aplicado las siguientes operaciones, desde las más sencillas hasta más complejas:

- Coincidencia de las formas morfológicas de palabras: buscando *pensar*, encontrar *piénsalo*.

  Este problema es bastante simple de resolver en el lenguaje inglés, al cual se dedica la mayor parte de investigación en el mundo. Sin embargo, para el español se convierte en un problema moderadamente serio, debido a la gran variedad de las formas de palabras en español.

  Los métodos de la morfología computacional –la rama del procesamiento de lenguaje natural que se encarga del modelado de las formas morfológicas de palabras– varían desde el uso de diccionarios que especifican las formas para cada palabra, hasta las heurísticas que ayudan a adivinarlas.

- Coincidencia de los sinónimos, conceptos más generales y más específicos: buscando *cerdo*, encontrar *puerco*, *mascota*, *animal*, etc.

Este problema prácticamente no depende del lenguaje (es tan importante para el inglés como para el español), aunque los diccionarios que se usan sí son específicos para cada lenguaje.

La idea principal es, como ya se dijo, el uso de diccionarios jerárquicos que especifican los sinónimos en el mismo nivel del árbol y los conceptos más específicos debajo de los conceptos más generales. Uno de los problemas que aún no reciben una solución adecuada es medir las distancias en este árbol: ¿qué tan parecida es la palabra *cerdo* a *puerco*?, ¿y a *mascota*?, ¿*animal*?, ¿*objeto*?

Una generalización de esta idea son los diccionarios de las palabras conceptualmente relacionadas, por ejemplo, *cerdo* y *tocino*; *sacerdote*, *Biblia*, *iglesia* y *rezar*. Aquí, el problema de la medición de distancia es aún más difícil.

- Tomar en cuenta las relaciones entre las palabras en la petición del usuario y en el documento: buscando *estudio de planes*, rechazar como no relevante *planes de estudio*.

  Para lograr este grado de calidad, se necesita reconocer (automáticamente) la estructura del texto y representarla en forma que permita la comparación necesaria, por ejemplo, en la forma de los grafos conceptuales.

Recientemente se adelantaron los desarrollos en una aproximación diferente al problema de búsqueda de información: generación automática de respuestas. La idea es la siguiente: en lugar de presentarle al usuario el documento completo donde probablemente se contiene la respuesta a su pregunta (por ejemplo, *¿cuándo fue la Revolución mexicana?*), simplemente darle la respuesta (en este caso, generar "En 1910-1917" basándose en la información encontrada en los textos).

Una de las técnicas más usadas para esto es la extracción de información: transformación de algunas partes de los textos libres en un formato de base de datos, por ejemplo: evento–fecha, artículo–lugar–precio, etc. Otra técnica posible es el razonamiento lógico sobre las relaciones encontradas en el texto.

Interfaces hombre máquina. Las computadoras están presentes en todos los aspectos de la vida cotidiana: en las oficinas, en las tiendas, en las escuelas, en los servicios públicos. Sin embargo, la gran mayoría de la gente no tiene la preparación adecuada para usarlas, económicamente es más ventajoso que las computadoras se adapten al modo de comunicación humano, que preparar a todas las personas para aprender cómo usar las máquinas, por ejemplo, que aprendan el SQL para formular con precisión sus preguntas.

De esto surge la idea ya muy conocida de las películas de ciencia ficción: las personas pueden comunicarse con las máquinas, dándoles órdenes en lenguaje natural y ellas, a su vez, son capaces de general respuestas en este mismo sentido.

Hablando de darles órdenes, no se trata de pronunciar los comandos especiales que generalmente se escogen de un menú: *abrir*, *edición*, *copiar*, *guardar*, *salir*. En lugar de esto, se trata de hablar a la máquina como se hablaría a un humano.

Un tipo específico de las interfaces en lenguaje natural consiste en la posibilidad de formular preguntas complejas a una base de datos, tomando como entrada una pregunta de un usuario, el sistema debe ser capaz de generar una sentencia SQL y generar la respuesta adecuada, sin que el cliente tenga conocimientos sobre el funcionamiento del sistema.

El problema más importante de este tipo de aplicaciones es que –a diferencia de las aplicaciones en la recuperación de información– se requiere entender exactamente la intención del usuario, ya que el costo de error puede ser muy alto. Realmente, si el robot entiende incorrectamente el co-

mando, pude hacer alguna acción destructiva o peligrosa. Si se malentiende la pregunta a la base de datos, la información proporcionada resultará incorrecta, lo que puede causar consecuencias graves.

Entonces, las interfaces en lenguaje natural en muchos casos requieren de las representaciones de información más detalladas y complejas, así como del análisis lingüístico más preciso y completo.

2.1.4 El estado de la investigación sobre la lengua española

Por razones históricas, la mayoría de la literatura sobre procesamiento de lenguaje natural no necesariamente está escrita en inglés, pero sí es el inglés el centro de tales estudios.

El número de hispanohablantes en el mundo supera los 400 millones, el español es una de las lenguas oficiales de la Organización de Naciones Unidas (ONU). Los métodos de enseñanza del español han sido bien descritos y la Real Academia de la Lengua Española constantemente financia investigaciones en ortografía, gramática y en la estandarización de la lengua. Existen también muchos buenos diccionarios de la lengua española de tipo académico.

Sin embargo, la investigación en lexicografía reflejada en tales diccionarios está muy orientada al ser humano. Junto con mucha información histórica, esos diccionarios proveen explicaciones semánticas, pero sin una descripción formal de las principales propiedades lingüísticas de lexemas, incluso en aspectos morfológicos y sintácticos.

La descripción formal y la algoritmización de un lenguaje es el objetivo de los equipos de investigación en lingüística computacional. Muchos equipos de investigación trabajan en Barcelona y Madrid. Sin embargo, incluso esto es bastante poco para un país de la Comunidad Europea, don-

de esfuerzos unilenguajes y multilenguajes son financiados por el gobierno y las agencias internacionales. Alguna investigación sobre el español se realiza en Estados Unidos.

En México – el país con mayor número de hablantes de la lengua española – la actividad en lingüística computacional ha sido bastante baja en las décadas pasadas. Actualmente, el equipo dirigido por el Prof. Luis Pineda Cortés en la Universidad Nacional Autónoma de México (UNAM) está trabajando en la muy dificultosa tarea de crear programas que sean capaces de llevar a cabo un diálogo en español con un humano. Otros equipos de investigadores trabajan en el Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) y en el Centro de Investigación en Computación del Instituto Politécnico Nacional.

2.2 Los niveles del lenguaje y la cadena de análisis

En el proceso de análisis típico para la comprensión del lenguaje natural, de forma general se distinguen los siguientes pasos[6]:

***Análisis morfológico*:** se analizan las cadenas de entrada y se les asigna uno o varios lemas (*lematización*). En este proceso se evalúan los signos de puntuación, y se les asigna una función determinada o son descartados.

***Análisis sintáctico*:** se transforman las secuencias lineales de palabras en estructuras que muestran la forma en que las palabras se relacionan entre sí.

***Análisis semántico*:** se asignan significados a las estructuras generadas por el analizador sintáctico, es decir, se establecen correspondencias entre las estructuras sintácticas y los objetos del dominio.

***Integración del discurso*:** el significado de una frase puede depender de las frases precedentes y también modificar el de las frases siguientes.

Análisis pragmático: la estructura de representación obtenida se interpreta para determinar su significado real y puntual dentro del contexto específico.

Este es el cuadro ideal de la cadena de análisis que, sin embargo, no se corresponde con la realidad. La mayoría de los sistemas no van más allá del análisis sintáctico. La dificultad que entraña el análisis de los niveles posteriores es evidente y además depende en gran medida del éxito obtenido en los análisis precedentes. Por otra parte, los límites entre los distintos niveles de análisis son muy difusos. En ocasiones las distintas fases son desarrolladas secuencialmente, mientras que en otras se hace de forma paralela. Un determinado analizador puede necesitar la ayuda de otro. De este modo, aunque por motivos de exposición suele ser útil distinguir estas fases de análisis, todas ellas interactúan de varias maneras, haciendo que sea imposible una completa separación. En definitiva, esta observación es válida no sólo para el análisis computacional, sino para el análisis lingüístico en general. En la figura 2.3 se muestra un diagrama de bloques de un analizador lingüístico típico, a continuación se detallan cada uno de sus niveles.

[6] Moreno Ortiz, Antonio *Estudios de Lingüística Española***,**

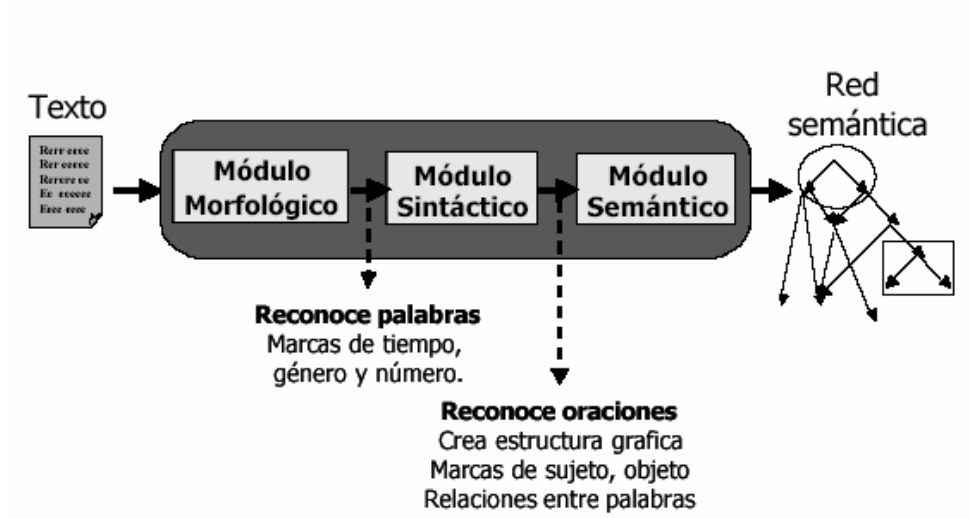


Fig. 2.3. Analizador lingüístico

2.2.1 Nivel morfológico

El análisis morfológico consiste en determinar la forma, clase o categoría gramatical de cada palabra de una oración.

La morfología abstrae las palabras de su contexto para clasificarlas en diferentes grupos según las funciones de que son capaces, estudia las diferentes formas que pueden adquirir para representar las categorías gramaticales y establece los medios que el idioma emplea para enriquecer su léxico formando nuevas palabras a base de las ya existentes.

Un ejemplo del resultado de realizar un análisis morfológico puede observarse en la siguiente tabla:

un	Artículo, singular, masculino
gato	Sustantivo, común, masculino, singular
negro	Adjetivo, singular, masculino
caza	Verbo *cazar*. Principal, indicativo, presente, tercera persona, singular
un	Artículo, singular, masculino
ratón	Sustantivo, común, masculino, singular
blanco	Adjetivo, singular, masculino

Tabla. 2.1 Un ejemplo de análisis morfológico de la oración

Un gato negro caza un ratón blanco.

2.2.2 Nivel sintáctico.

La tarea principal del nivel sintáctico es describir las relaciones entre las palabras de una oración y la función que cada palabra realiza, es decir, construir la *estructura de la oración*.

Las estructuras sintácticas se construyen con una *gramática*, la cual es una especificación mediante reglas de las estructuras permitidas en un determinado lenguaje. Las oraciones correctas son aquellas que obedecen las reglas gramaticales. El establecimiento de métodos para determinar únicamente las secuencias correctas es uno de los objetivos de los formalismos gramaticales en la lingüística computacional, se han considerado dos enfoques para describir formalmente la *gramaticalidad* de las oraciones: los constituyentes y las dependencias.[7]

[7] Galicia Haro, Sofía, Análisis Sintáctico conducido por un diccionario de patrones de manejo sintáctico del español Tesis Doctoral, CIC-IPN, México, 2000, pp. 14-18.

Enfoque de constituyentes

El enfoque de constituyentes consiste en analizar la oración mediante un proceso de segmentación y clasificación. La oración se segmenta en sus partes constituyentes, estas se clasifican como categorías gramaticales, se repite el proceso para cada parte, subdividiendo y clasificando, y así sucesivamente hasta que las partes constituyentes sean indivisibles, el resultado es un árbol como el que se muestra en la figura 2.3, donde los nodos terminales representan a las palabras que constituyen a la oración, y los nodos intermedios y la raíz representan las reglas de reescritura especificadas en la gramática.

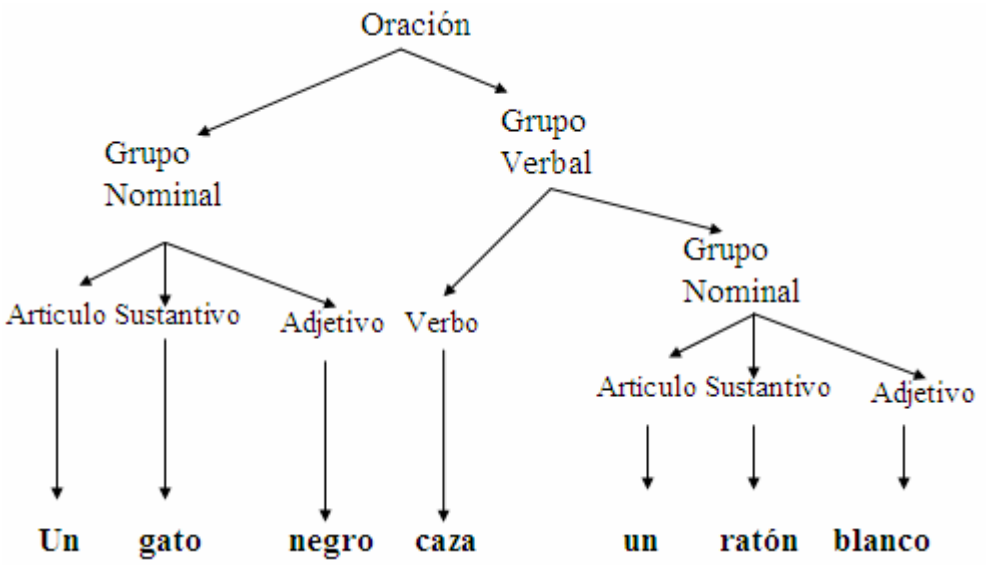


Fig. 2.4 Árbol de constituyentes de la oración ***un gato negro caza un ratón blanco.***

La línea de trabajo más importante e influyente respecto al enfoque de constituyentes corresponde al trabajo del matemático y lingüista Noam Chomsky.

Enfoque de dependencias

El enfoque de dependencias consiste en el establecimiento de la relación entre pares de palabras, una de ellas tiene el rol de *rectora* y la otra el rol de *dependiente* o subordinada. Si cada palabra de una oración tiene una palabra rectora, toda la oración se puede ver como una estructura jerárquica, el árbol de dependencias, donde la única palabra que no tiene rectora es la raíz del árbol.

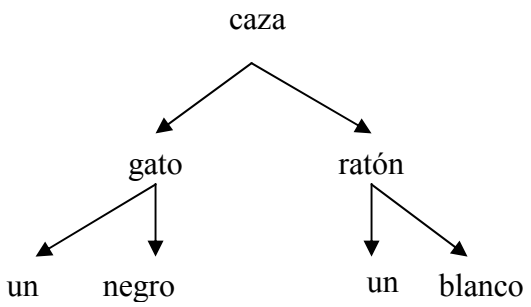


Fig. 2.5 Árbol de dependencias de la oración ***un gato negro caza un ratón blanco***.

El primer intento por formalizar una gramática de dependencias fue el trabajo desarrollado por Lucien Tesnière en 1959, actualmente la línea de trabajo más importante es la desarrollada por el investigador Igor Mel'cuk, la *Meaning* ⇔ *Text Theory* (MTT).

2.2.3 Nivel semántico

El propósito del análisis semántico es determinar el significado de las oraciones y representarlo de manera formal. Existen varias formas de representación formal semántica de las oraciones, tales como la lógica de primer orden y las redes semánticas. Los investigadores están más o menos de acuerdo en que los resultados del análisis semántico deben ser redes semánticas, donde se representan todos los conceptos y las relaciones entre ellos. Otra posible representación es algo muy parecido a las redes semánticas: los grafos conceptuales. Entonces, lo que se necesita saber

es cómo hacer la transformación de un árbol sintáctico a una red semántica. Ese problema todavía no tiene una solución general[8].

2.2.3.1 Redes semánticas

Las redes semánticas han sido ampliamente utilizadas en la inteligencia artificial como mecanismo de representación de conocimiento, por ello existe una gran diversidad de técnicas. Los elementos comunes en la mayoría de los esquemas de redes semánticas son:[9]

- Estructuras de datos en *nodos*, que representan conceptos, unidas por *arcos* que representan las relaciones entre los conceptos.
- Un conjunto de procedimientos de inferencia que operan sobre las estructuras de datos.

Se pueden distinguir tres categorías de redes semánticas:

- **Redes IS-A**
- **Redes de marcos**
- **Grafos conceptuales**

Redes IS-A

El tipo de red semántica por excelencia es el de las redes IS-A, tanto que suele utilizarse como sinónimo de red semántica. Una red IS-A es una jerarquía taxonómica cuya columna vertebral está constituida por un sistema de enlaces de herencia entre los objetos o conceptos de representación, conocidos como *nodos*. Estos enlaces o arcos pueden estar etiquetados "IS-A", también "SUPERC", "AKO", "SUBSET", etc.

Las redes IS-A son el resultado de la observación de que gran parte del conocimiento humano se basa en la adscripción de un subconjunto de elementos como parte de otro más general. Las

[8] Gelbukh, A y Grigori Sidorov, *Procesamiento automático del español con enfoque en recursos léxicos grandes,* IPN, 2006, p 68.
[9] Moreno Ortiz, Antonio, *Op. Cit.*

taxonomías clásicas naturales son un buen ejemplo: un perro es un cánido, un cánido es un mamífero, un mamífero es un animal. Obteniendo un número de proposiciones:

$\forall x$ (*perro* (*x*)) $\rightarrow$ *cánido* (*x*);
$\forall x$ (*cánido* (*x*)) $\rightarrow$ *mamífero* (*x*);
$\forall x$ (*mamífero* (*x*)) $\rightarrow$ *animal* (*x*);

La estructuración jerárquica facilita que la adscripción de propiedades a una determinada categoría se reduzca a aquellas que son específicas a la misma, heredando aquellas propiedades de las categorías superiores de la jerarquía.

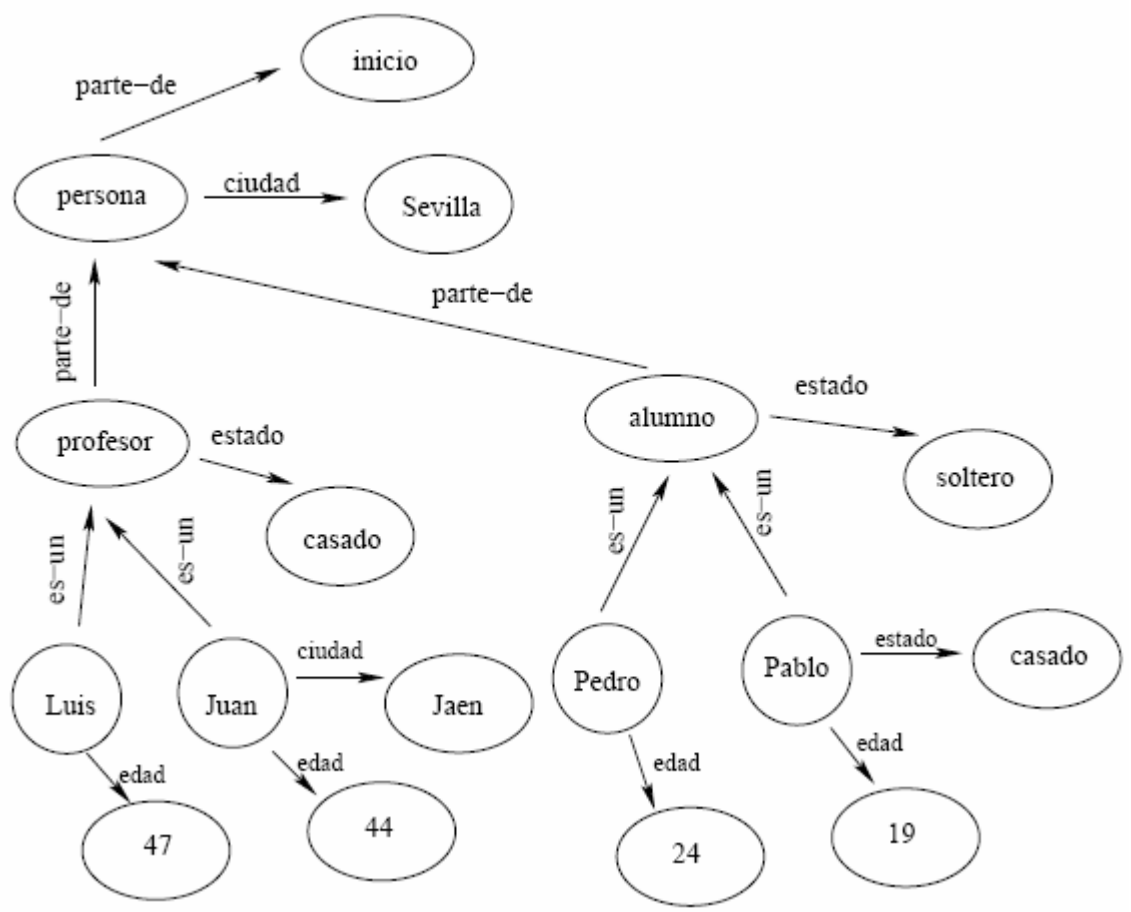


Figura 2.6 *Ejemplo de red IS-A*

Redes de marcos

El conocimiento taxonómico puede representarse en estructuras de datos denominadas marcos. Un marco tiene un conjunto de pares atributo-valor. El nombre de un marco se corresponde con el nodo de una red IS-A, los atributos se corresponden con los nombres de los arcos asociados a ese nodo, y los valores se relacionan con los nodos a los que apuntan dichos arcos. Los pares atributo-valor se suelen denominar ranuras (slots). Existen dos tipos de marcos: marcos de clase y marcos de instancia. Los marcos de clase representan conceptos y describen un conjunto de propiedades; los marcos de instancia representan objetos concretos y heredan propiedades de los marcos de clase.

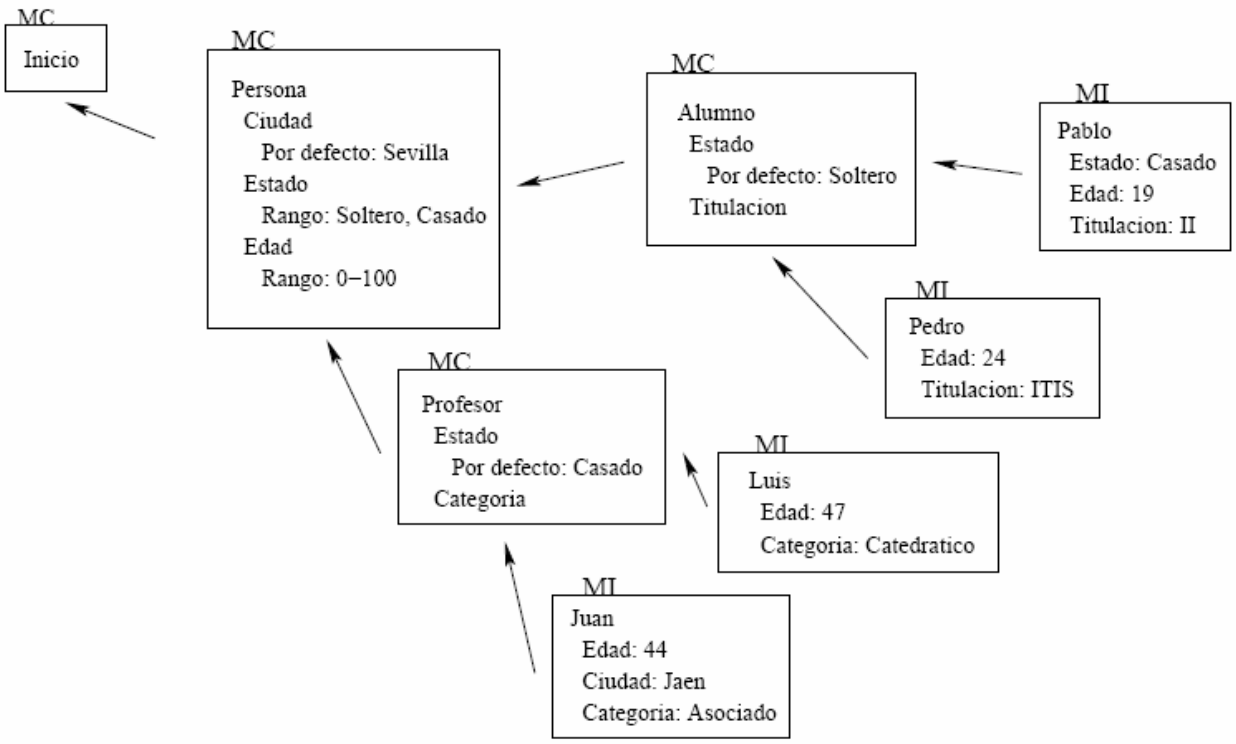


Figura 2.7 *Ejemplo de red de marcos equivalente a la red IS-A de la página anterior.*

Grafos conceptuales

Los grafos conceptuales tienen dos tipos de nodos: conceptos y relaciones, y cada arco une solamente a un concepto con una relación conceptual.

Figura 2.8 Un grafo conceptual.

Los grafos conceptuales tienen un gran potencial para representar, en forma simple y directa, algunos detalles finos del lenguaje natural que son difícilmente expresados por las redes semánticas o la lógica de predicados. El capítulo 4 está dedicado enteramente al formalismo de los grafos conceptuales.

2.3 Las aplicaciones de los grafos conceptuales

2.3.1 Minería de texto con grafos conceptuales

Actualmente, debido al gran valor del conocimiento y a la disponibilidad de grandes conjuntos de datos, muchas instituciones han identificado como una de sus necesidades prioritarias el diseño de los mecanismos que automaticen el análisis de datos, la extracción de información de éstos, y su conversión en conocimiento[10]. Uno de los esfuerzos más importantes en esta dirección son los sistemas de minería de texto. Estos sistemas permiten analizar grandes colecciones de textos y descubrir en ellos distintos tipos de patrones interesantes.

La minería de texto se realiza en dos fases, una de preprocesamiento, donde los textos son transformados a algún tipo de representación semiestructurada que permite el análisis automático, y una fase de descubrimiento, donde las representaciones intermedias son analizadas y son identificados algunos patrones interesantes, por ejemplo: agrupamientos, asociaciones, desviaciones y/o tendencias son posiblemente descubiertos.

La mayoría de los métodos actuales de minería de texto utilizan representaciones sencillas del contenido de textos, por ejemplo, listas de palabras clave. Por una parte, estas representaciones son construidas y analizadas fácilmente, pero por otra parte, estas representaciones limitan gradualmente los tipos de patrones descubiertos.

Recientemente, en muchas aplicaciones relacionadas con el análisis de texto existe la tendencia por empezar a usar representaciones del contenido de los textos mas completas que las palabras clave, es decir, representaciones que consideran más tipos de elementos textuales. En la minería de texto, por ejemplo, se cree que estas representaciones permitirán extender los tipos y mejorar

[10] Montes y Gómez, Manuel, *Minería de texto empleando la semejanza entre estructuras semánticas Tesis Doctoral*, CIC-IPN, 2002, pp. 18-20.

la expresividad de patrones descubiertos. Los grafos conceptuales son una opción para la representación del contenido de textos. En la última década se ha desarrollado un conjunto de métodos tales como la *comparación, agrupamiento conceptual*, *descubrimiento de asociaciones* y *detección de desviaciones* que han convertido a los grafos conceptuales en una herramienta de gran potencial para la minería de texto, por su capacidad para representar los detalles finos del lenguaje natural.

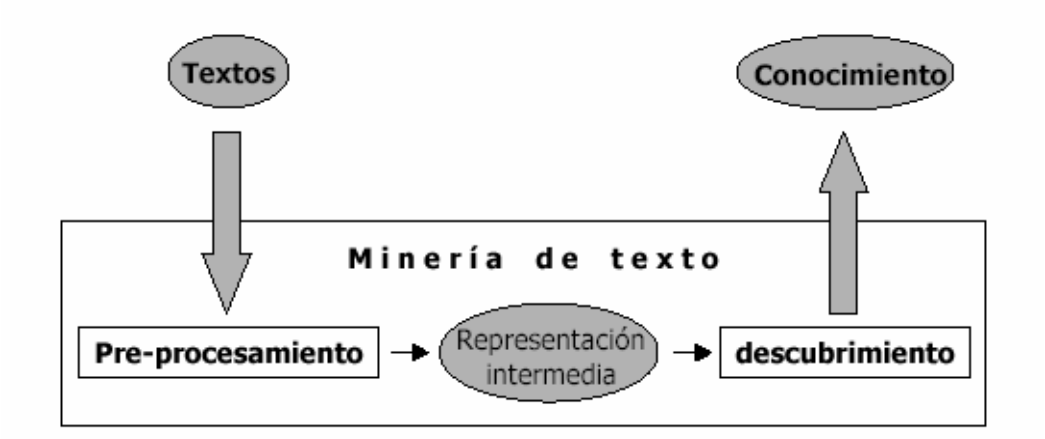


Fig. 2.9 Proceso de minería de texto.

2.3.2 Recuperación de información

El enfoque clásico de la recuperación de información es el modelo de espacios vectoriales, la caracterización formal planteada por el modelo de espacios vectoriales representa documentos de lenguaje natural a través de vectores en un espacio lineal multivectorial. Los documentos son representados como vectores de términos indexables o palabras clave. La valoración de la relevancia de documentos en una búsqueda por palabras clave se pueden calcular, teniendo en cuenta la teoría de las semejanzas del documento, comparando la desviación de los ángulos de cada vector del documento y el vector original de la pregunta, donde la pregunta del usuario se representa también como un vector del mismo tipo.

El enfoque de grafos conceptuales aplicados a la recuperación de información consiste en la indexación de los documentos no sólo mediante palabras clave sino también considerando las relaciones conceptuales entre ellas, esto es aplicable también a las preguntas del usuario, los grafos conceptuales ayudan a determinar la similaridad entre el requerimiento de usuario y el conjunto de documentos encontrados.

2.3.3 Traducción automática

Entre los métodos existentes para la traducción automática se encuentran los que están basados en interlingua. El proceso de traducción en estos sistemas ocurre en dos etapas. Se le asigna una estructura al texto de origen, usando para ello únicamente información de la lengua origen. Esta estructura es una oración en un lenguaje universal que representa el "significado" del texto. Partiendo de esta estructura, se puede generar el texto correspondiente de cualquier lengua meta sin importar cual fue la lengua de origen. Esta separación entre el conocimiento de la lengua de origen y la lengua meta es la principal motivación de la traducción automática de interlingua.

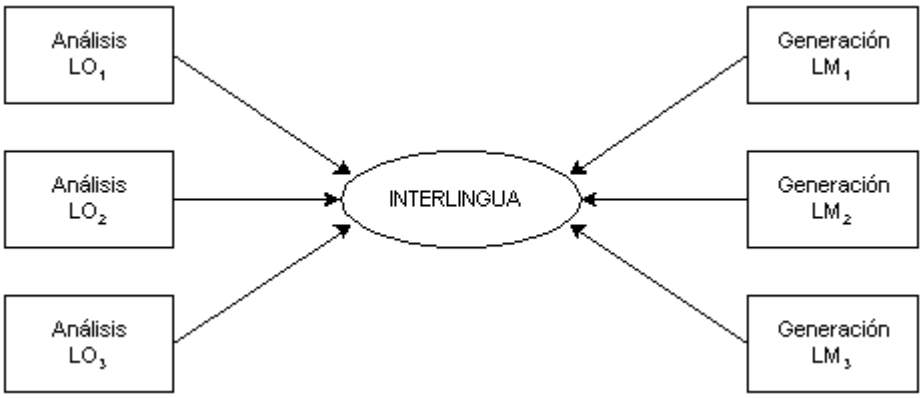


Fig. 2.10 Estructura de un sistema de traducción con interlingua.

Para la representación se usan estructuras formales de predicados lógicos o sus equivalentes, por ejemplo, redes semánticas o grafos conceptuales.

2.4 Algoritmos existentes para la obtención de grafos conceptuales

La mayor parte de los trabajos de generación de grafos conceptuales a partir de texto en lenguaje natural se dedica al idioma inglés, entre ellos podemos mencionar el desarrollado por Svetlana Hensman[11] (Universidad de Dublín). En la figura 2.11 se muestra la arquitectura general del sistema.

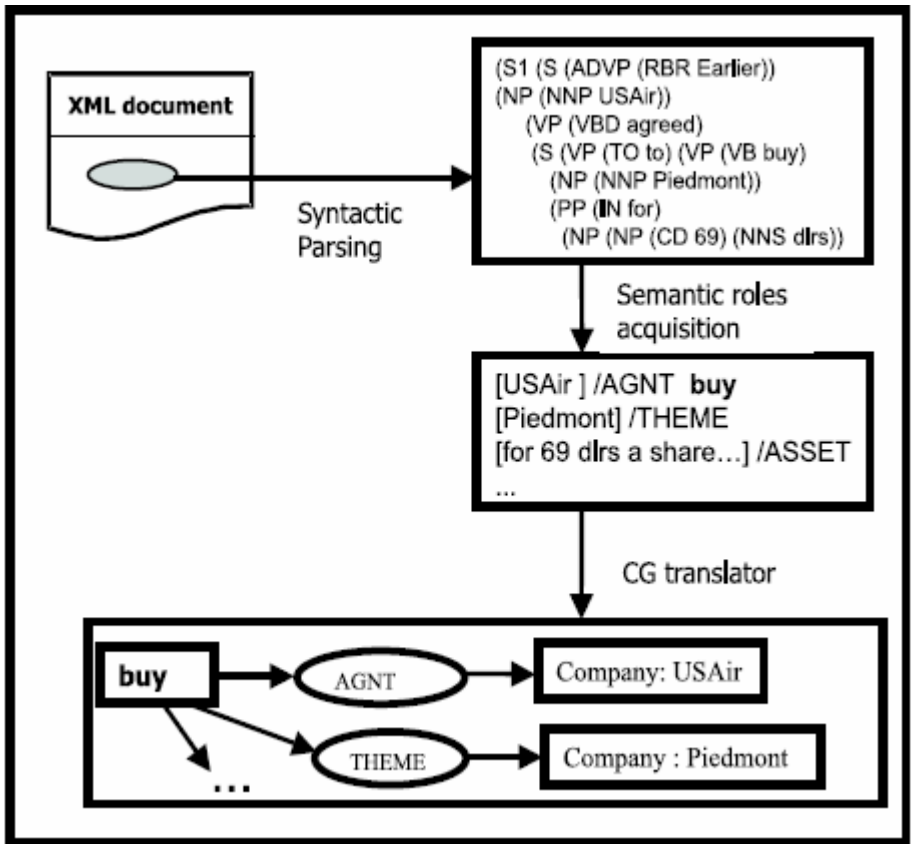


Fig. 2.11 Proceso de generación de grafos conceptuales de Svetlana Hensman

Este método consta de dos etapas, la primera consiste en la identificación de los roles semánticos en la oración mediante la utilización de WordNet y VerbNet, y en una segunda etapa utiliza los roles identificados y un conjunto de reglas sintáctico/semánticas para construir un grafo conceptual.

El proceso de construcción de un grafo conceptual consta de tres pasos:

- Paso 1. *Para cada constituyente de la oración se construye un grafo conceptual*. Cada parte de la oración debe ser representada por un grafo conceptual. Esto se hace recursivamente analizando la estructura sintáctica de la frase.

- Paso 2. *Ligar los grafos conceptuales que representan los constituyentes en un único grafo conceptual.* Todos los grafos conceptuales construidos en el paso 1 deben ser unidos al concepto que representa el verbo, y así crear un grafo conceptual que represente la oración completa.

- Paso 3. *Resolver las relaciones desconocidas*. En este paso se identifica todas las etiquetas asignadas en los pasos anteriores. Esto se realiza usando una lista de reglas de corrección.

Por otro lado, Lei Zhang y Yong Yu[12] (Universidad de Shangai) proponen un método para generar grafos conceptuales para textos de un dominio restringido, utilizando técnicas de aprendizaje automático y la gramática de ligas (*link grammar*). La gramática de ligas es un formalismo de sintaxis basado en constituyentes. Consiste en establecer arcos (*links*) etiquetados entre pares de

[11] Hensman, Svetlana, *Construction of Conceptual Graph representation of texts*.
[12] Zhang Lei y Yong Yu, *Learning to Generate Conceptual Graphs from Domain Specific Sentences*.

palabras, las ligas representan las dependencias sintácticas y semánticas, al conjunto de ligas de una oración se le llama *ligamiento*, estos deben satisfacer ciertas condiciones: *planaridad*, ningún arco puede cruzarse con otro y *conectividad*, las ligas deben ser suficientes para que todas las palabras estén conectadas. Un *link parser* debe buscar todos los posibles ligamientos para una sentencia dada.

El la figura 2.12 se muestra la correspondencia entre la gramática de ligas y los grafos conceptuales. A diferencia del árbol de constituyentes, la gramática de ligas captura también las relaciones semánticas entre las palabras.

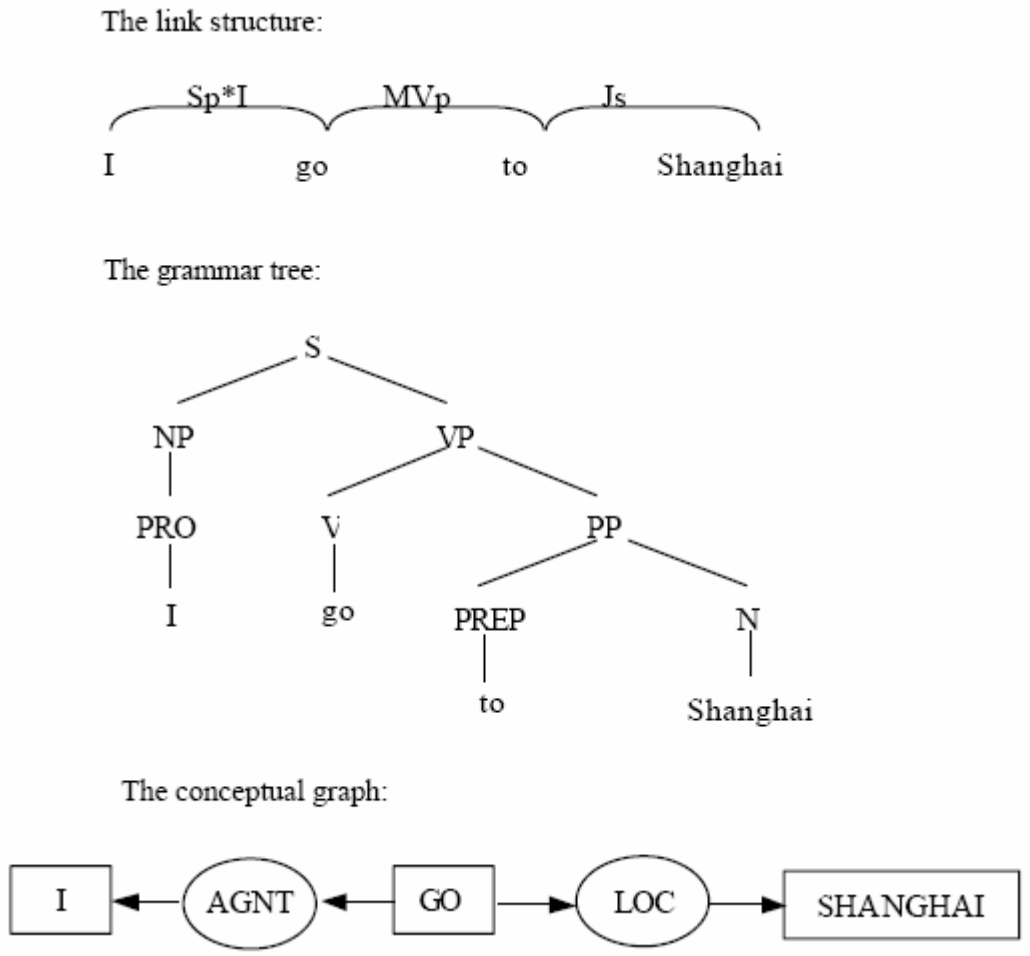


Fig. 2.12 *Link Grammar* y grafos conceptuales

2.5 Algunos analizadores de texto existentes y disponibles en el mercado

Existen un conjunto de herramientas comerciales y experimentales para el análisis de textos, la mayoría cubre la fase del análisis morfológico y análisis sintáctico. Algunas ofrecen la característica de generar una representación semántica de las oraciones. A continuación se describen algunas de estas herramientas.

2.5.1 Conexor Tools

Conexor es una empresa que ha desarrollado un conjunto de herramientas para procesamiento de lenguaje natural, cuya cobertura abarca desde al análisis morfológico hasta el análisis semántico,

Machinese Phrase Tagger. Es una herramienta de etiquetado morfológico, devuelve los lemas de las palabras de una oración y reconoce las clases de los constituyentes.

Text	Base-form	Phrase syntax and part-of-speech
un	uno	premodifier, determiner
gato	gato	nominal head, noun, noun phrase begins
negro	negro	postmodifier, adjective, noun phrase ends
caza	cazar	main verb, indicative present
un	uno	premodifier, determiner
raton	raton	nominal head, noun, noun phrase begins
blan-co	blanco	postmodifier, adjective, noun phrase ends, sentence boundary

Tabla 2.2. Resultados de Machinese Phrase Tagging para la oración *un gato negro caza un ratón blanco*

Machinese Sintax. Es un analizador sintáctico que devuelve la estructura de las oraciones, mostrando las dependencias sintácticas entre las palabras.

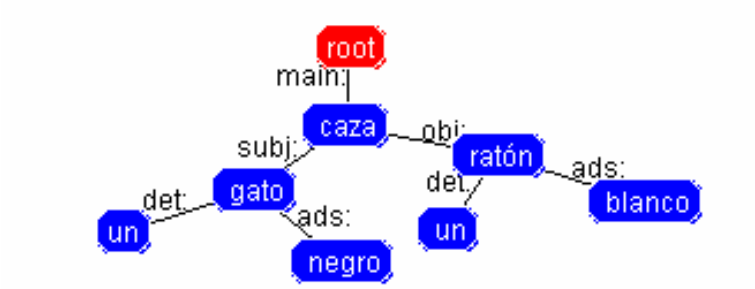


Fig. 2.13 Resultados de Machinese Sintax para la oración *un gato negro caza un ratón blanco*

Machinese Semantics. Es un analizador semántico que reconoce el rol semántico así como características gramaticales y léxicas. La salida del analizador es una representación de la estructura funcional de la oración.

2.5.2 ARIES Natural Language Tools

El conjunto de herramientas y recursos para lenguaje natural ARIES, desarrollado en la Universidad Politécnica de Madrid, conforman una plataforma léxica para la lengua española. Estas herramientas se pueden integrar en aplicaciones de procesamiento de lenguaje natural. Incluye: un lexicón español grande, herramientas para el mantenimiento y acceso al lexicón y un analizador/un generador morfológicos.

2.5.3 LBK

LBK es un entorno de desarrollo de analizadores léxicos y sintácticos basados en el formalismo de las gramáticas de unificación. Aunque no está restringido al formalismo HPSG (Head-Driven Phrase Structure Grammar), LBK implementa el formalismo de referencia DELPH-IN (Deep Linguistics Proccessing With HPSG**).**

2.6 Discusión

Existe un conjunto amplio de herramientas para procesamiento del lenguaje natural, la mayoría de ellas dedicadas a la fase morfológica y sintáctica de la cadena de análisis, un número pequeño de ellas llegan al nivel semántico y pocas utilizan grafos conceptuales como medio de representación semántica de las oraciones.

Asimismo existen varios métodos propuestos para generar grafos conceptuales a partir de texto, la mayoría dirigidos por sintaxis. Sin embargo, éstos sólo se han implementado para dominios restringidos y para el idioma inglés (no hemos encontrado sistemas existentes para ningún lenguaje distinto a inglés). Además, estos sistemas operan de una manera heurística, sin un sólido fundamento sintáctico, –es por eso que dependen del dominio específico para el cual se implementaron las heurísticas. Realizan todo el procesamiento dentro de un sólo programa, de tal manera que para mejorar el comportamiento de este programa se necesita reescribir su código o sus heurísticas.

En esta tesis, se propone un sistema que funciona sobre textos abiertos (no de un dominio restringido) y se aplica al lenguaje español. Nuestro programa toma como entrada la salida del analizador sintáctico de propósito general, basado en una gramática de gran cobertura. Con eso, se desacopla el desarrollo del convertidor del desarrollo del analizador sintáctico. Entre más se mejorará en el futuro el analizador sintáctico (o si se usará otro analizador mejor), mejores serán los resultados obtenidos por nuestro sistema, sin necesidad alguna de alterar su código (los analizadores sintácticos son desarrollados por otros equipos de investigadores independientes de nosotros y para diversos usos independientes de nuestro sistema).

III. Antecedentes

3.1 Introducción

La generación de grafos conceptuales está inscrito en la cadena de análisis lingüístico como una etapa posterior al análisis sintáctico. Para su funcionamiento, el programa propuesto requiere de una representación sintáctica de las oraciones, así como la información morfológica de cada uno de sus elementos. La herramienta que se utilizó para la generación de las estructuras sintácticas es el analizador sintáctico Parser 1.0, desarrollado en el Centro de Investigación en Computación del IPN.

3.2 El analizador sintáctico Parser 1.0

Dentro de las herramientas que el Laboratorio de Procesamiento Natural del Centro de Investigación en Computación ha desarrollado para el idioma español, está el analizador sintáctico Parser 1.0. Esta herramienta permite investigar la estructura sintáctica y morfológica de oraciones mediante un formalismo de gramática libre de contexto extendida. Es útil para el aprendizaje del formalismo de gramática libre de contexto extendida y para desarrollar y probar la gramática[13].

[13] Gelbukh A., Sidorov G, Galicia H., *Documentación de Parser 1.0.*

3.2.1 Interfaz del programa

El analizador PARSER 1.0 desarrollado en lenguaje C++, tiene una interfaz gráfica que facilita la visualización de los resultados del análisis de oraciones. La interfaz contempla la visualización de:

- Árbol de constituyentes, en forma gráfica o texto.
- Árbol de dependencias.
- Estructura morfológica.

A continuación se muestra un ejemplo del análisis realizado por el PARSER 1.0.

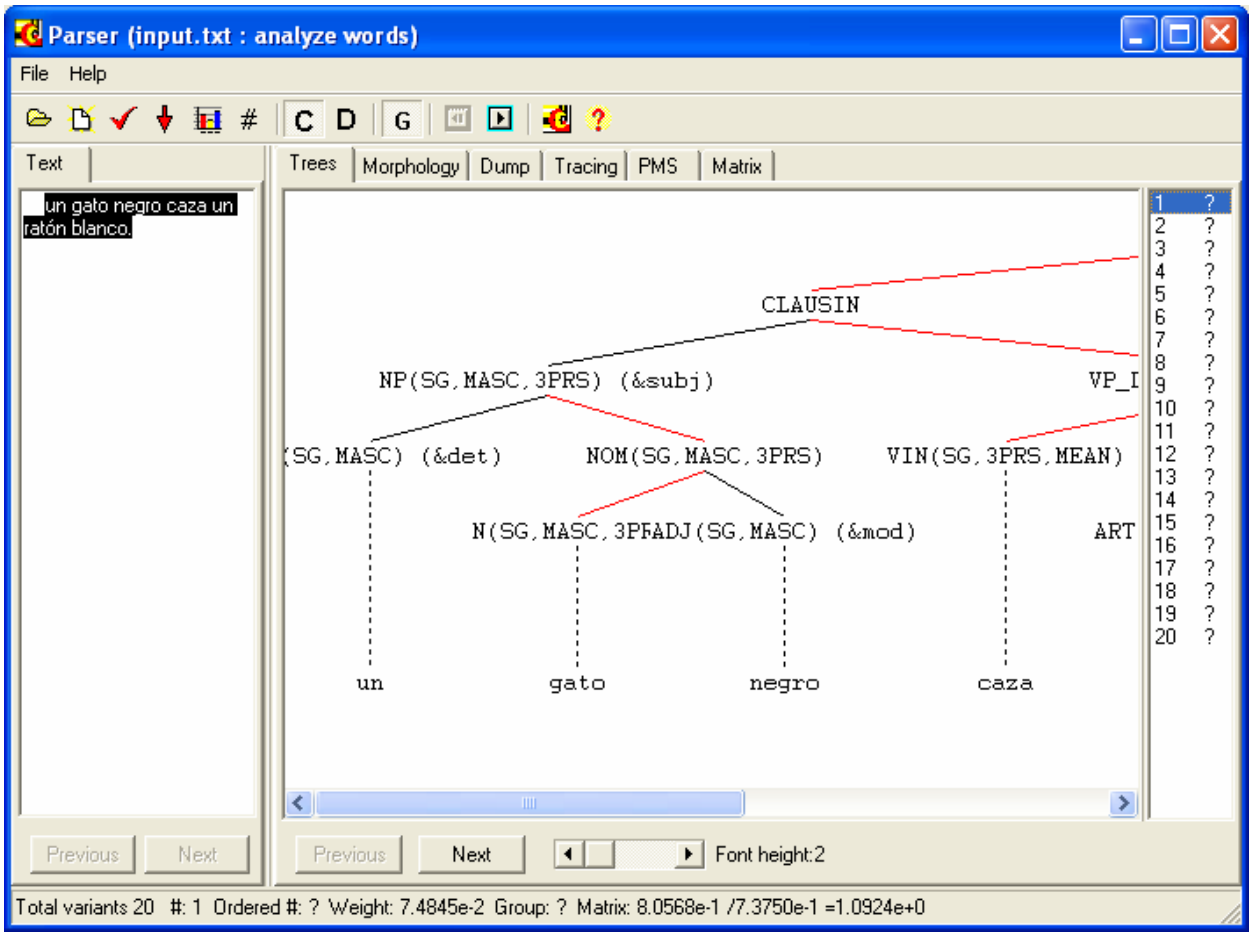


Fig. 3.1 La pantalla principal del PARSER 1.0 muestra gráficamente el árbol sintáctico de la oración *un gato negro caza un ratón blanco*.

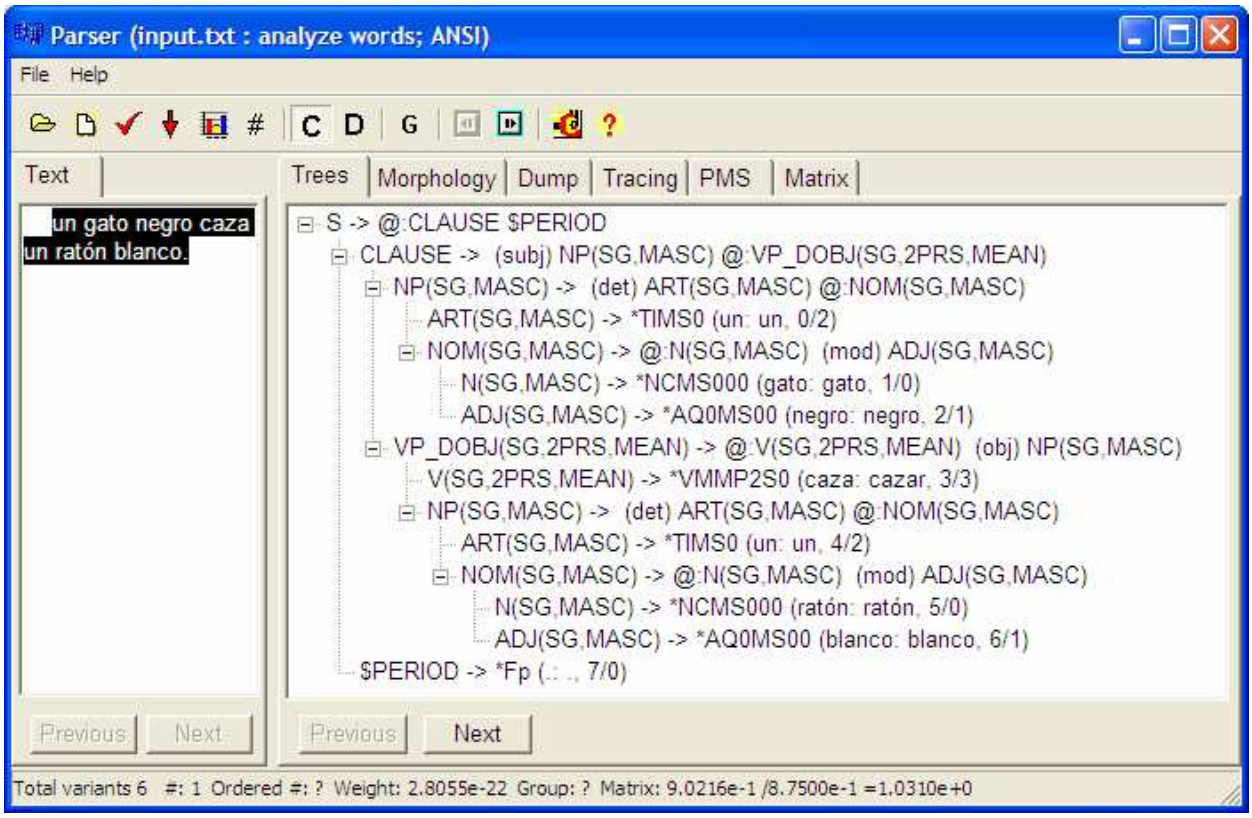


Fig. 3.2 La pantalla principal del PARSER 1.0 muestra el árbol de constituyentes de una oración en español.

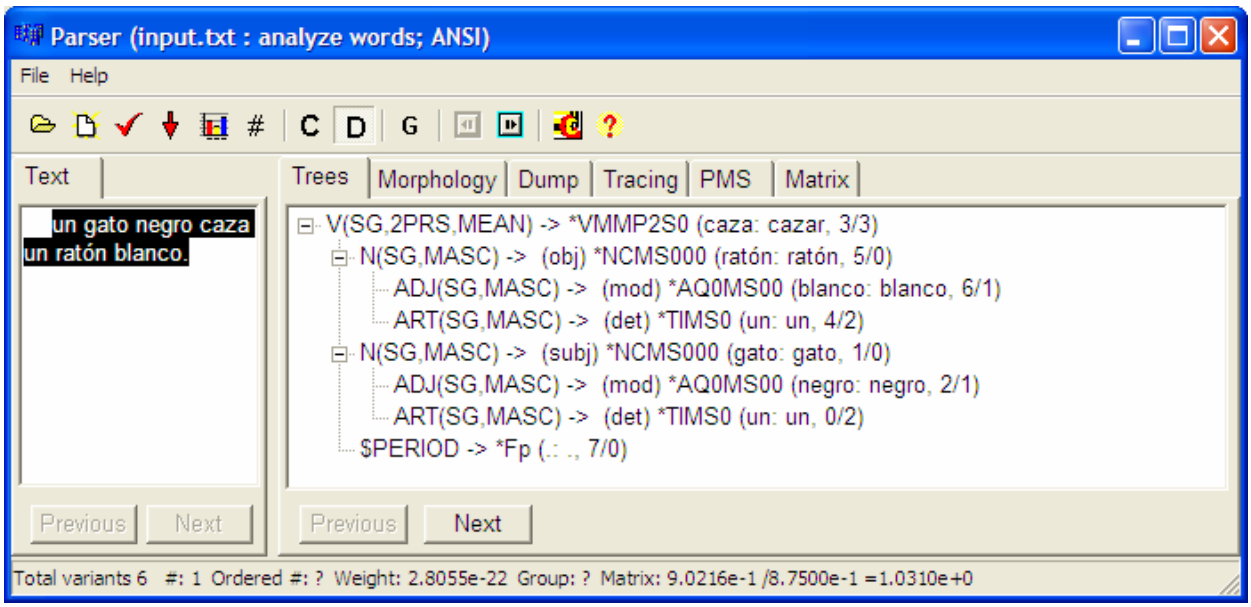


Fig. 3.3 La pantalla principal del PARSER 1.0 muestra el árbol de dependencias de una oración en español.

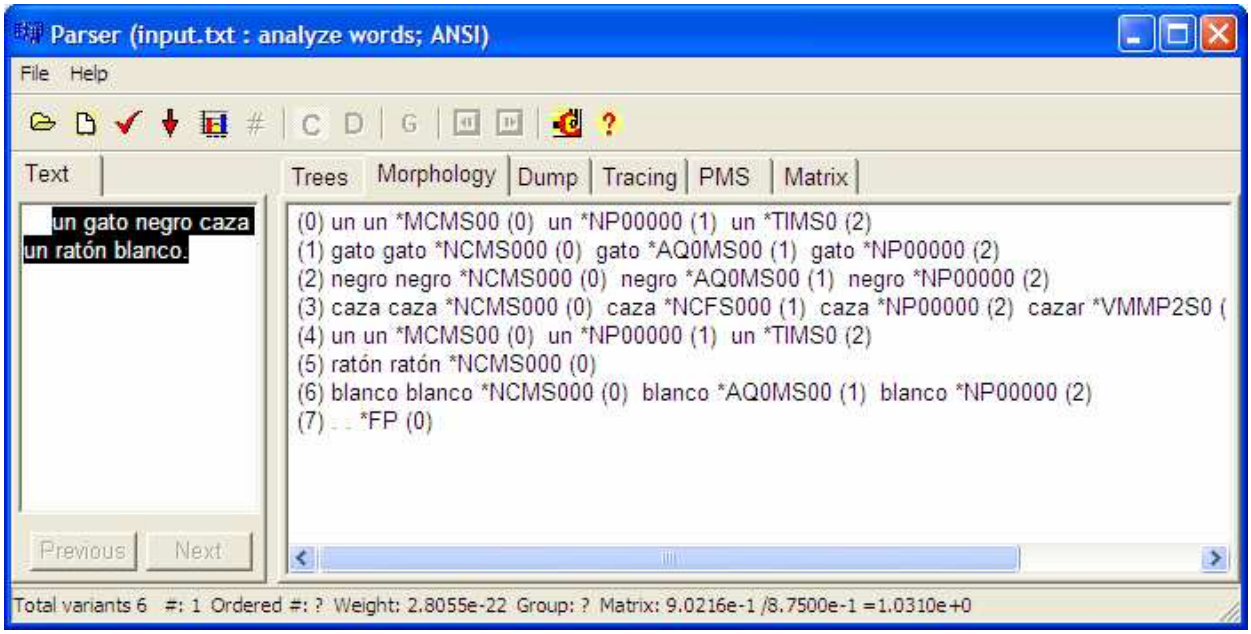


Fig. 3.4 La pantalla principal del PARSER 1.0 muestra la estructura morfológica de una oración en español.

3.3 Otras herramientas y recursos necesarios

3.3.1 Ontologías y taxonomías

Una *ontología* define los términos a utilizar para describir y representar un área de conocimiento. Las ontologías son utilizadas por las personas, las bases de datos y las aplicaciones que necesitan compartir un dominio de información (un dominio es simplemente un área de temática específica o un área de conocimiento, tales como medicina, fabricación de herramientas, bienes inmuebles, reparación automovilística, gestión financiera, etc.). Las ontologías incluyen definiciones de conceptos básicos del dominio, y las relaciones entre ellos[14]. Una *taxonomía* es una particular disposición de los objetos de una ontología con estructura de inclusión (hipérnimos → hipónimos) en forma de árbol.

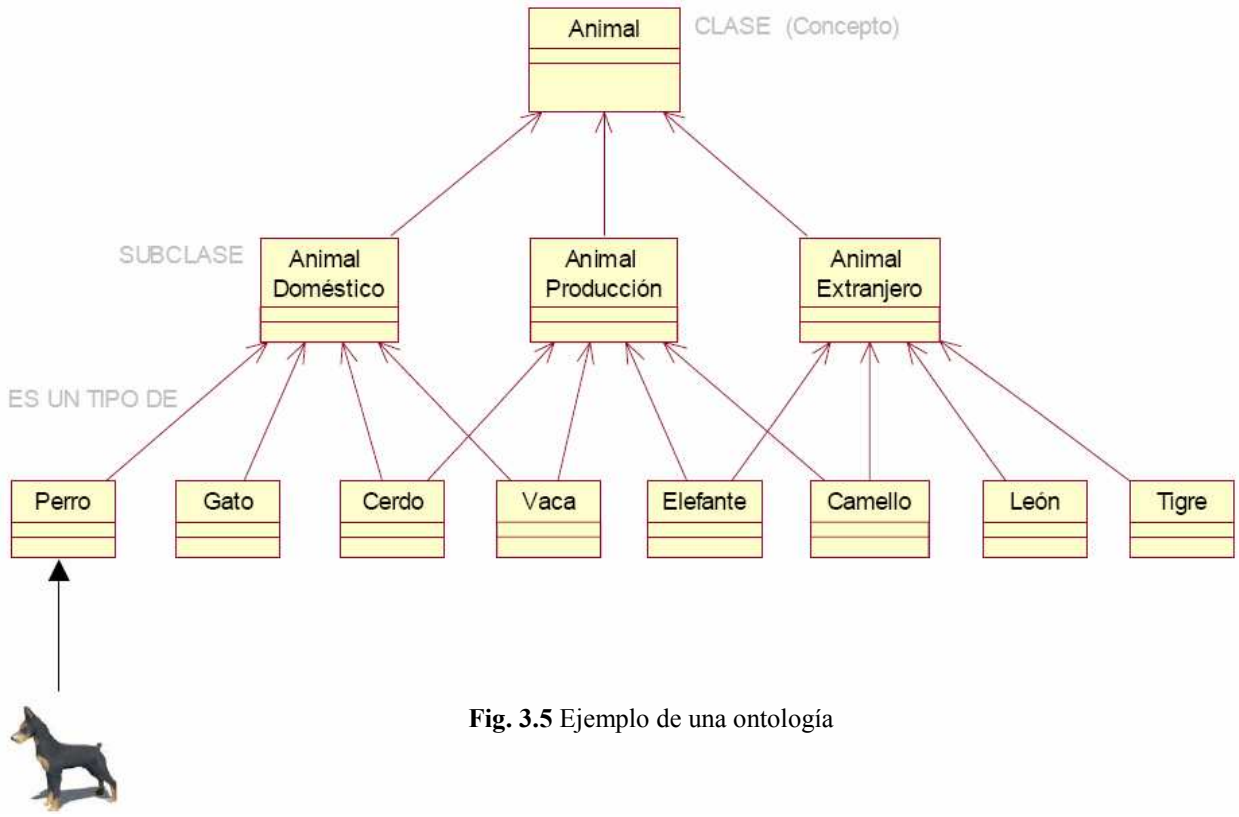


Fig. 3.5 Ejemplo de una ontología

3.3.2 WordNet

Wordnet es un recurso lingüístico desarrollado para su uso automático como apoyo, principalmente en tareas de análisis semántico. WordNet está organizada con base en relaciones. Los significados se representan mediante synsets. Las relaciones semánticas se representan mediante relaciones entre synsets. Las relaciones más importantes contempladas en WordNet son:

- Sinonimia.

[14] *OWL Web Ontology Language Use Cases and Requirements,* Word Wide Web Consortium (W3C), 2004.

- Antonimia.
- Hiponimia / hiperonimia. (relación es-un)
- Holominia / meronimia. (relación parte-de)

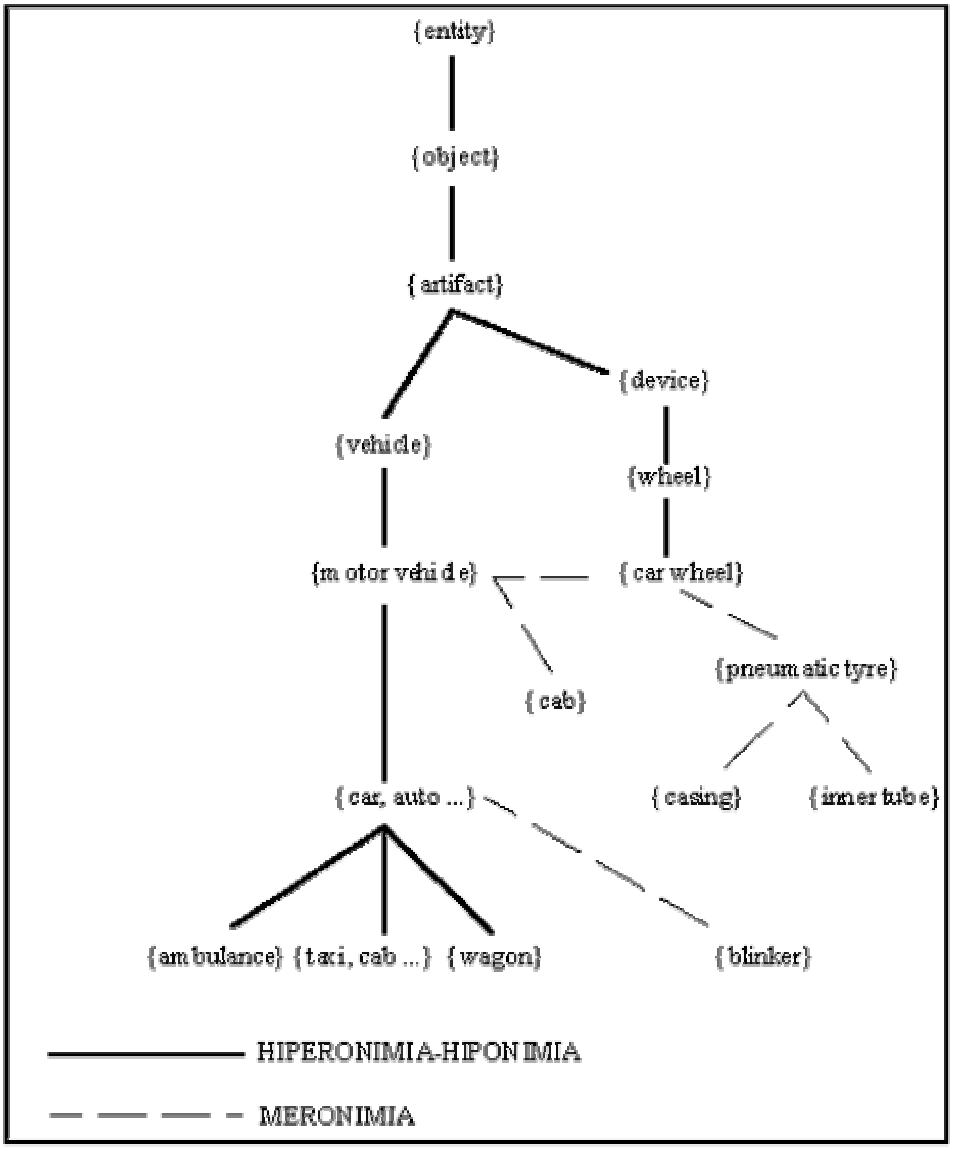


Fig. 3.6 Esquema parcial de relaciones en WordNet

3.3.3 Gramática de cláusulas definidas

El programa propuesto utiliza la extensión DCG (Definite Clause Grammar) de PROLOG como herramienta para la validación de los grafos generados. DCG es una extensión de las gramáticas libres de contexto que la mayoría de las implementaciones de PROLOG incorporan.

```
oracion --> sintagma_nominal, sintagma_verbal.
sintagma_nominal --> articulo, nombre, adjetivo.
sintagma_verbal --> verbo, sintagma_nominal.
articulo --> [un].
nombre --> [gato];[ratón].
verbo --> [caza].
adjetivo --> [negro];[blanco].
```

Fig. 3.5 Ejemplo de una gramática DCG.

3.3.4 TuPROLOG

TuPROLOG es una implementación de PROLOG basada en Java desarrollada por la Universidad de Boloña[15], se distribuye como un paquete de clases listo para usarse. TuPROLOG permite desarrollar programas que realicen llamadas a la máquina PROLOG, combinando las posibilidades de un lenguaje de propósito general como Java con las de un lenguaje orientado a la programación lógica. Asimismo, la integración entre Java y PROLOG se da en ambos sentidos, permitiendo incorporar objetos Java en predicados PROLOG.

[15] http://www.alice.unibo.it/tuProlog.

3.3.5 Herramientas para la minería de texto

Como parte de su trabajo de tesis doctoral, el Dr. Manuel Montes y Gómez del INAOE desarrolló un conjunto de herramientas para minería de texto basada en grafos conceptuales. Estas herramientas toman como entrada un conjunto de grafos conceptuales, básicamente realizan análisis de comparativo de grafos para descubrir similaridades, desviaciones, etc. Los grafos generados por el programa aquí propuesto pueden ser usados como entrada por tales herramientas ya que utilizan el formato CGIF[16].

[16] Ver en el capítulo IV el formato de los grafos conceptuales.

IV. Marco teórico: el formalismo de los grafos conceptuales

4.1 Introducción

Los grafos conceptuales son una forma de representación de conocimiento basado en la lingüística, la psicología y la filosofía. En un grafo conceptual, los nodos *concepto* representan entidades, atributos, estado y eventos, mientras que los nodos *relación* muestran cómo los nodos concepto se interconectan.

Según Sowa[17], la percepción es el proceso de construcción de un *modelo de trabajo* que representa e interpreta la entrada sensorial (*sensory input*). Este modelo tiene dos componentes: una parte sensorial formada por un conjunto de *perceptos*, cada uno de los cuales se corresponde con un aspecto de la entrada sensorial, y una parte más abstracta llamada *grafo conceptual*, el cual describe cómo los *perceptos* se acomodan para formar un mosaico. La percepción está basada en los siguientes mecanismos:

- La estimulación es registrada por una fracción de segundo en una forma llamada *icono sensorial*.
- El *comparador asociativo* busca en la memoria de largo plazo los conceptos que corresponden con todo o con una parte del icono.
- El *ensamblador* coloca los perceptos juntos en un modelo de trabajo que forma un aproximación cercana a la entrada sensorial. El registro del ensamblador se almacena como grafo conceptual.

[17] Sowa, J.F, *Conceptual Structures*, Addison-Wesley, 1984.

- Los mecanismos conceptuales procesan los conceptos concretos que tienen perceptos asociados y conceptos abstractos a los que no tienen asociado algún percepto.

Por ejemplo, cuando una persona ve un gato, las ondas de luz reflejadas por el animal son recibidas como un icono sensorial *s*. El comparador asociativo establece la correspondencia de *s* con un único percepto *p* o con una colección de perceptos, los cuales son combinados por el ensamblador en una imagen completa. Así como el ensamblador combina los perceptos, registra los perceptos y sus interconexiones en un grafo conceptual.

Gráficamente, los grafos conceptuales son dibujados como rectángulos y círculos ligados. Esas ligas representan las asociaciones lógicas en el cerebro, no las figuras actuales de las excitaciones neuronales.

El proceso de *percepción* consiste entonces en la generación de una estructura *u,* llamada grafo conceptual, en respuesta a una entidad externa o escena *e*, donde:

- La entidad *e* es percibida como un *icono sensorial s*.
- El *comparador asociativo* encuentra uno o más perceptos $p_1,...p_n$ que se corresponden con todo o una parte de *s*.
- El *ensamblado*r combina los perceptos $p_1...,p_n$ para formar un *modelo de trabajo* que se aproxima a *s*.
- Si tal modelo de trabajo se puede construir se dice que la entidad *e* es *reconocida* por los perceptos $p_1,...p_n$.
- Para cada percepto p_i en el modelo de trabajo, hay un concepto c_i llamado la interpretación de p_i.
- Los conceptos $c_1,...c_n$ son ligados por las relaciones conceptuales para formar el grafo conceptual *u*.

Los perceptos son fragmentos de imágenes que se ajustan unas con otras, como un rompecabezas. Un grafo conceptual describe de qué forma son ensamblados los perceptos. Las relaciones conceptuales especifican el *rol* de cada percepto: un percepto se corresponde con una parte de un icono a la izquierda o derecha de otro percepto; un percepto para un color puede ser combinado con un percepto de una figura para formar el grafo que representa una figura coloreada. Para los perceptos auditivos, un grafo puede especificar cómo los fonemas son ensamblados para formar sílabas y palabras. Tales grafos han sido usados para el lenguaje y la visión.

4.2 El formato de los grafos conceptuales

Gráficamente los conceptos en un grafo conceptual son representados por rectángulos y las relaciones por círculos.

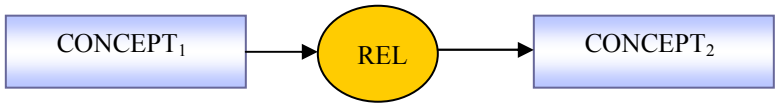


Fig. 4. 1 Grafo conceptual

En texto lineal, los rectángulos pueden ser abreviados por corchetes y los círculos por paréntesis.

$[CONCEPT_1] \rightarrow (REL) \rightarrow [CONCEPT_2]$

Para recordar la dirección de las flechas, el grafo anterior puede leerse como *La REL de* $CONCEPT_1$ *es* $CONCEPT_2$, por ejemplo:

[morder] → (AGNT) → [perro],

Que debe leerse *el AGENTE de MORDER es un PERRO.* La dirección contraria de la flechas produciría el absurdo *el AGENTE de PERRO es un MORDER.*

Las relaciones conceptuales pueden tener cualquier número de arcos; sin embargo, lo más común es que sean diádicas. Algunas, como la marca de tiempo pasado (PAST) o la negación (NEG), pueden son monádicas. Otros, como *entre* (BETW), son triádicos. La figura 4.2 muestra un grafo conceptual para la frase *un espacio está entre un ladrillo y un ladrillo*.

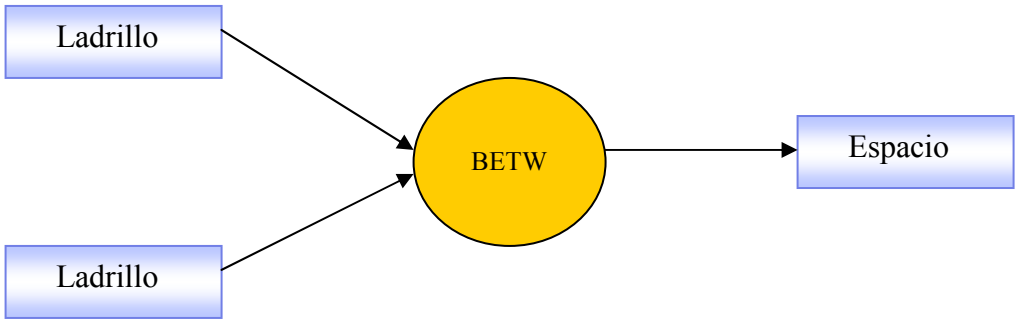


Fig. 4.2 Relación triádica

Definición. Un grafo conceptual es un grafo *bipartito*, esto es, que tiene dos tipos de nodos: conceptos y relaciones conceptuales, y cada arco une solamente a un concepto con una relación conceptual.

Aunque los diagramas ayudan a visualizar las relaciones, la teoría es independiente de cómo los grafos sean dibujados. La notación de los rectángulos y círculos es una conveniencia, pero no es fundamental, lo que es fundamental, es la base matemática, la cual es suficiente para representar un conjunto de relaciones entre entidades discretas. Para entidades concretas tales como gatos y tomates, el cerebro tiene perceptos para reconocer la entidad y pensar acerca de ellas; la gente puede pensar, por ejemplo, acerca de un restaurante como un lugar para comer sin imaginar deta-

lles acerca del mobiliario o decorado. Sin embargo, para entidades abstractas tales como JUSTICIA y SALUD, sólo están disponibles conceptos sin imágenes.

4.2.1 Conceptos

Los conceptos representan entidades, acciones y atributos, y tienen un tipo conceptual y un referente. El tipo conceptual representa la clase de elemento representado por el concepto, mientras que el referente indica la instancia específica referida por dicho concepto. Por ejemplo, el grafo de la Fig. 4.3, el concepto `[gato:Félix]` tiene el tipo gato y el referente Félix:

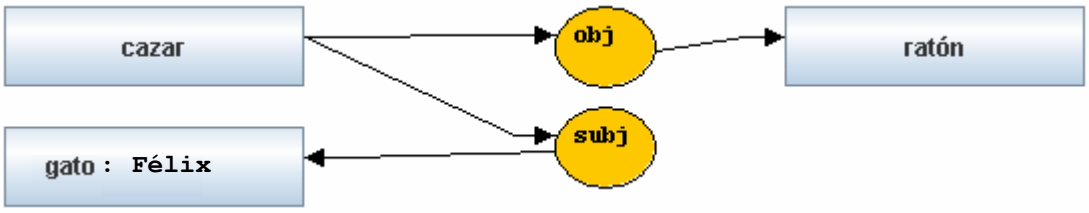


Fig. 4.3 Grafo conceptual para *El gato Félix caza un ratón*

Tipos conceptuales

Los tipos de conceptos se organizan en jerarquías de tipos (fig. 4.4), esta jerarquía es un ordenamiento parcialmente definido sobre el conjunto de tipos determinado por el símbolo $\leq$. Entonces, dada una jerarquía de tipos, considerando que *s*, *t* y *u* representan tres tipos conceptuales, se puede establecer lo siguiente[18]:

- Si $s \leq t$, entonces *s* es un *subtipo* de *t*, y *t* es un *supertipo* de *s*.
- Si $s \leq t$ y $s \neq t$, entonces *s* es un *subtipo propio* de *t*, se expresa como $s < t$ y *t* es un *supertipo propio* de *s*, expresado como $t > s$.

- Si *s es un subtipo de t y a la vez un subtipo de u* ($s \leq t$ y $s \leq u$), entonces *s* es un *subtipo común* de *t* y *u*.
- Si *s es un supertipo de t y a la vez un supertipo de u* ($t \leq s$ y $u \leq s$), entonces *s* es un *supertipo común* de *t* y *u*.

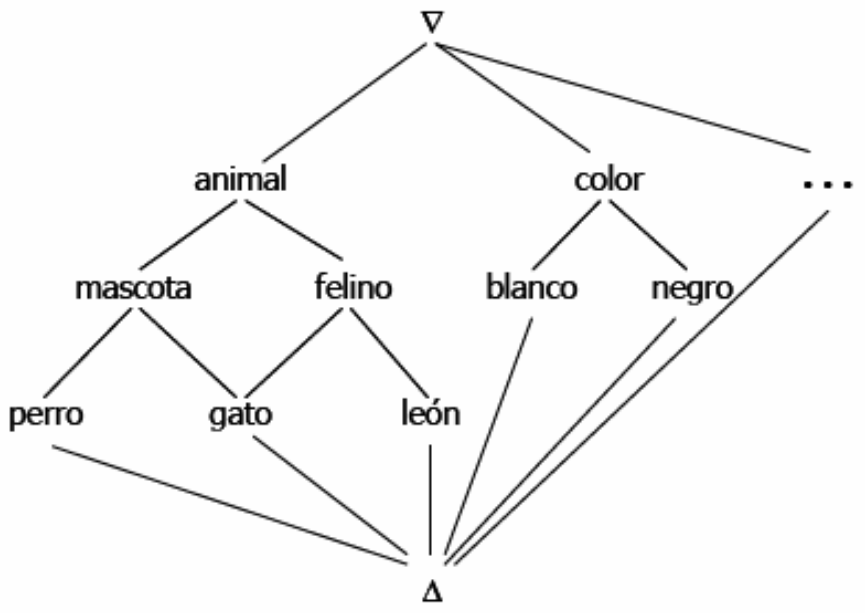


Fig. 4.4 Una jerarquía de tipos

Referentes

Los referentes pueden ser de dos clases: genéricos e individuales. Los genéricos se refieren a conceptos no especificados, por ejemplo, el concepto [`ratón`] de la figura 4.3 significa *un ratón*.

Los referentes individuales funcionan como sustitutos de elementos específicos del mundo real. Por ejemplo, el concepto [`gato:Félix`] de la figura 4.3 es un sustituto del gato Félix.

[18] Montes y Gómez, Manuel, *Op. Cit*.

Algunas notaciones estándares empleadas en los referentes se enlistan en la siguiente tabla:

`[gato]`	un gato
`[gato:*x]`	un gato x
`[gato:#]`	el gato
`[gato:Félix]`	un gato llamado Félix
`[periodo:@5min]`	un periodo de 5 minutos
`[gato:{*}]`	unos gatos
`[gato:{*}@3]`	tres gatos
`[gato: Félix, Garfield]`	unos gatos llamados Félix y Garfield

Tabla. 4.1 Notación de referentes

4.2.2 Relaciones conceptuales

Las relaciones conceptuales indican la manera en la que los conceptos se relacionan. Las relaciones conceptuales tienen un tipo relacional y una valencia. El tipo de relación indica el rol *semántico* que realizan los conceptos ligados a la relación, y la valencia indica el número de conceptos con los cuáles el nodo relación está ligado.

A continuación se enlistan algunas propiedades de las relaciones conceptuales:

- El número de arcos que pertenecen a una relación conceptual es igual a su valencia, una relación conceptual de valencia *n* es llamada *n*-aria, y sus arcos son numerados de 1 a *n*.

- Para cada relación conceptual *n*-aria existe una secuencia de *n*-tipos conceptuales denominada la *firma* de la relación. Esta firma restringe el tipo de conceptos que pueden conectarse a cada uno de los arcos de la relación conceptual.
- Todas las relaciones conceptuales del mismo tipo tienen la misma valencia y la misma firma.

En el grafo de la figura 4.3, la relación conceptual `(obj)` indica que es el ratón el que es cazado por el gato Félix, esta es una relación binaria con firma (entidad, entidad).

4.2.3 Grafos canónicos

Un grafo conceptual es una combinación de nodos concepto y nodos relación, donde cada arco de cada relación conceptual es ligada a un concepto; sin embargo, no todas las combinaciones tienen sentido, algunas de ellas incluyen combinaciones absurdas como:

`[soñar]` → `(agnt)` → `[idea]` → `(attr)` → `[verde]`

Lo cual podría leerse como *la idea verde sueña*. Para distinguir el significado de grafos que representan situaciones reales en el mundo externo, ciertos grafos son declarados como *canónicos*. A través de la experiencia, cada persona desarrolla una vista del mundo representada en grafos canónicos. Así, los grafos pueden ser *canonizados* por los siguientes tres procesos:

Percepción. Cualquier grafo conceptual construido en correspondencia con un icono sensorial es canónico.
Reglas de formación. Nuevos grafos canónicos pueden derivarse de otros grafos canónicos por las reglas *copia*, *restricción*, *fusión* y *simplificación*.
Interiorización. Grafos conceptuales arbitrarios pueden ser asumidos como canónicos.

Reglas de formación

Una *copia* exacta de un grafo canónico es también canónica. La regla de *restricción* reemplaza la etiqueta de tipo de concepto por un subtipo o por un concepto individual.

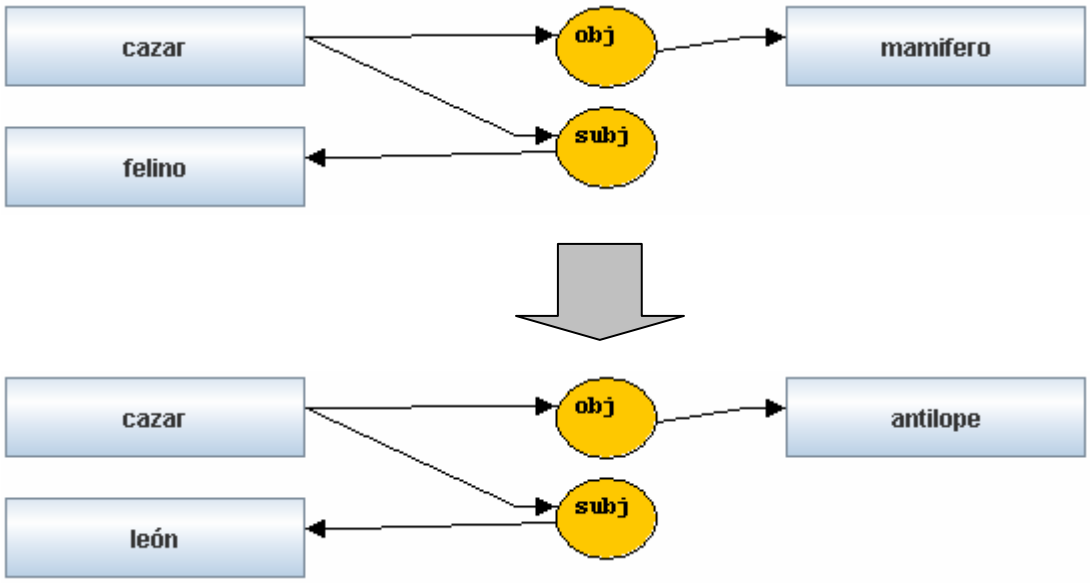


Fig. 4.5 Restricción de un grafo

Mediante las regla de fusión y simplificación, dos o más conceptos idénticos y comunes se fusionan en un solo concepto y las relaciones conceptuales se ligan a este concepto resultante. Cuando dos conceptos son fusionados, algunas relaciones en el grafo resultante pueden ser redundantes. Uno de cada par de los nodos redundantes puede ser borrado mediante la regla de simplificación, cuando dos relaciones del mismo tipo son ligadas al mismo concepto en el mismo orden, entonces ambas proporcionan la misma información, por lo que una de ellas pueda ser borrada. En la figura 4.6 se muestra la fusión de dos grafos conceptuales que representan las ora-

ciones: *un gato persigue rápidamente* y *un gato persigue un ratón*, *el resultado es un gato persigue rápidamente un ratón*. En el grafo resultante se elimina la duplicidad del concepto [perseguir] y el grafo resultante sigue siendo canónico.

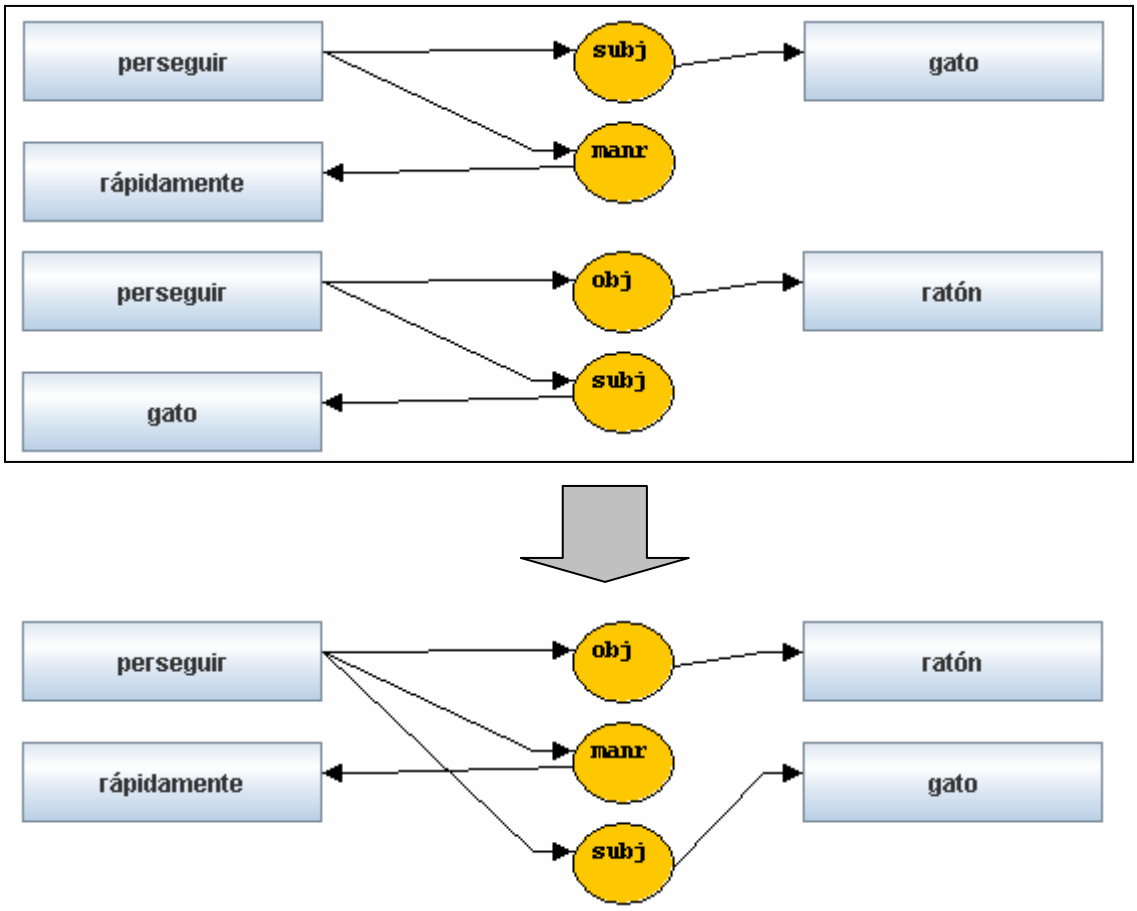


Fig. 5.6 Fusión y simplificación de dos grafos

Las reglas de formación son un tipo de gramática de grafos, además de la definición de la sintaxis, también implementan ciertas restricciones semánticas. En el apartado 4.4 se enlistan las relaciones conceptuales y el número y tipo de concepto a los cuales pueden estar ligadas.

Generalización

Las reglas de formación canónica son *reglas de especialización*. La restricción por ejemplo especializa el concepto [mamífero] a [antílope] o [vaca]. La generalización procede en sentido inverso. Mientras la especialización no preserva la verdad, la generalización sí lo hace. Por ejemplo, un grafo que represente *un felino caza un mamífero* podría ser especializado en un *gato caza una vaca*, en cambio, podría ser generalizado como, *un mamífero caza un mamífero*.

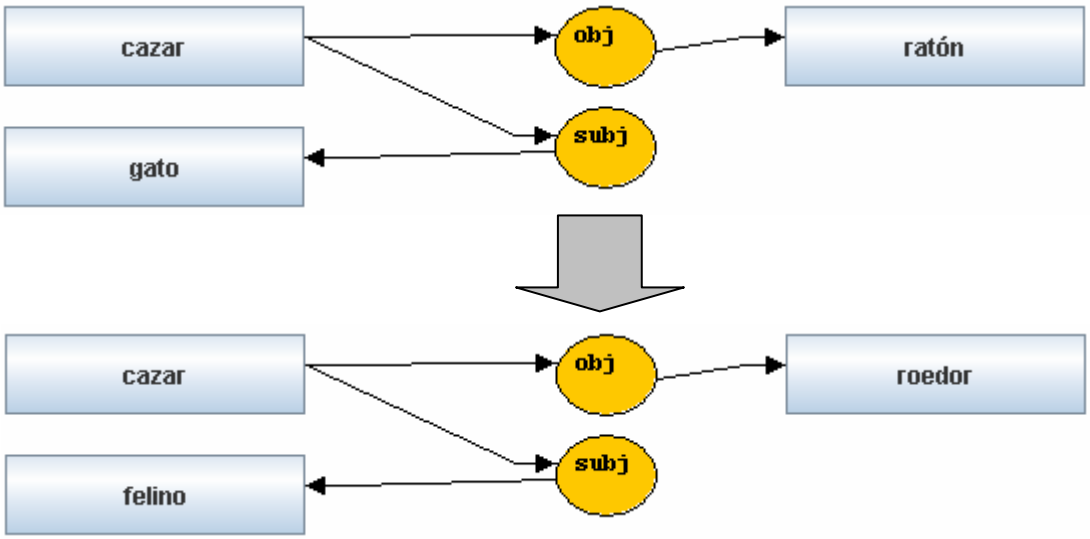


Fig. 4.7 Generalización de un grafo conceptual

Formalmente, la generalización define un ordenamiento parcial de grafos conceptuales llamado jerarquía de generalización; para cualesquiera grafos conceptuales *u*, *v*, y *w*, las siguientes propiedades son verdaderas:

- Reflexiva: $u \leq u$.
- Transitiva: si $u \leq v$ y $v \leq w$, entonces $u \leq w$.
- Antisimétrica: si $u \leq v$ y $v \leq u$ entonces $u = v$.
- Subgrafo: si v es un subgrafo de u entonces $u \leq v$.

- Subtipos: si *u* es idéntico a *v* excepto que uno o más tipos de *v* son restringidos a subtipos en *u*, entonces $u \leq v$.
- Individuales: Si *u* es idéntico a *v* excepto que uno o más conceptos genéricos de *v* son restringidos a conceptos individuales de ese mismo tipo, entonces $u \leq v$.
- Cima. El grafo [**T**] (*top*) es una generalización de todos los grafos.

4.3 Tipos de conceptos

En este apartado se muestra una lista parcial de tipos conceptuales. Para cada tipo, se muestran uno o más categorías de orden superior en la jerarquía de tipos (Fig. 4.8).

- **ACCION** < EVENTO. Una acción en un evento con un AGENTE animado.
  `[ACT]->(AGNT)->[ANIMADO]`

- **ANIMADO** < ENTIDAD. Los seres animados son los agentes de las acciones.

- **ANIMAL** < ANIMADO, ENTIDAD-MOVIL, OBJETO-FISICO.

- **ATRIBUTO** < T. Un atributo es una cualidad de una entidad.
  `[ENTIDAD]->(ATTR)->[ATRIBUTO]`

- **CIUDAD** < LUGAR. Una ciudad es un lugar.

- **ENTIDAD** < T. Una entidad incluye todos los objetos físicos y las abstracciones.

- **ENTIDAD-ESTACIONARIA** < ENTIDAD.

- **ENTIDAD-MOVIL** < ENTIDAD.

- **EVENTO** < T. Un evento incluye acciones de agentes animados así como sucesos, tales como explosiones, donde puede no estar presente un agente.

- **LUGAR** < ENTIDAD-ESTACIONARIA.

- **MEDIDA** < T. Las medidas no tienen otro supertipo que T.

- **OBJETO-FISICO** < ENTIDAD. Un objeto físico es un tipo de entidad.

- **PERSONA** < ANIMAL. Una persona es un tipo de animal.

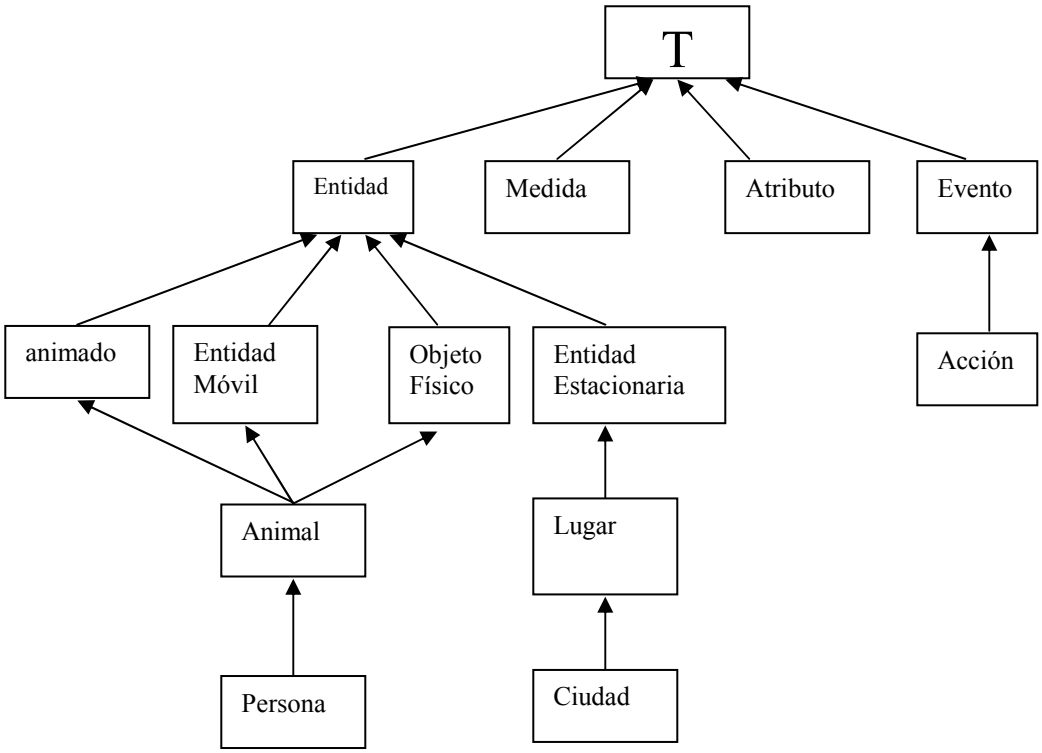


Fig. 4.8 Una jerarquía de tipos conceptuales.

4.4 Tipos de relaciones conceptuales

La siguiente tabla contiene una lista parcial de las relaciones conceptuales más frecuentes.

Tabla. 4.2 Relaciones conceptuales

(AGNT) Agente	Liga [ACCION] a [ANIMADO], donde ANIMADO es el actor de la acción. Ejemplo: *Eva muerde una manzana*. [MORDER]- (AGNT)->[PERSONA: Eva] (OBJ)->[MANZANA]
(ATTR) Atributo	Liga [ENTIDAD:*x] a [ENTIDAD:*y], donde *x tiene un atributo *y.. Ejemplo: *Un perro bravo.* [PERRO]->(ATTR)->[BRAVO]
(INST) Instrumento	Liga [ACCION] a [ENTIDAD]. *Ejemplo: La llave abrió la puerta.* [ABRIR]- (INST)->[LLAVE:#] (OBJ)->[PUERTA:#]
(LOC) Locación	Liga a [T] a [LUGAR]. Ejemplo: *Juan llegó a Tijuana*. [LLEGAR]- (AGNT)->[PERSONA:Juan] (LOC)->[CIUDAD: Tijuana]
(MANR) Manera	Liga [ACCION] a [ATRIBUTO], *Ejemplo: Juan llegó rápidamente*. [LLEGAR]- (AGNT)->[PERSONA:Juan] (MANR)->[RAPIDAMENTE]
(OBJ) Objeto	Liga [ACCION] a una [ENTIDAD] sobre la que se actúa. Ejemplo: *El gato caza un ratón*. [CAZAR]- (AGNT)->[GATO:#] (OBJ)->[RATON]
(PART) Parte	Liga una [ENTIDAD:*x] con una [ENTIDAD: *y] donde *y es parte de *x. Ejemplo: Un dedo es parte de una mano. [DEDO]->(PART)->[MANO]
(POSS)	Liga un [ANIMADO] con una [ENTIDAD], la cual es poseída por el ser anima-

Posesión	do. Ejemplo: *El reloj de Ana se detuvo.* [PERSONA:ANA]->(POSS)->[RELOJ:#]<-(OBJ)<-[DETENER]
(RCPT) Recipiente	Liga una [ACCION] a un [ANIMADO] el cual recibe o resultado de la acción. Ejemplo: *Un diamante fue regalado a Juana.* [REGALAR]- (OBJ)->[DIAMANTE] (RCPT)->[PERSONA:Juana]
(RSLT) Resultado	Liga una [ACCION] a una [ENTIDAD] que es generada por el acto. Ejemplo: *Pedro escribió un artículo.* [ESCRIBIR]- (AGNT)->[PERSONA: Juan] (RSLT)->[ARTICULO]

V. Generación de grafos conceptuales

5.1 Introducción

Los grafos conceptuales tienen un gran potencial para representar de forma simple y directa los detalles finos del lenguaje natural[19].

La regla de transformación básica es que palabras tales como sustantivos, verbos, adjetivos y adverbios, corresponden a nodos concepto, y las palabras funcionales tales como las preposiciones, conjunciones y verbos auxiliares, corresponden a nodos relación.

5.1.1 Correspondencias entre grafos conceptuales y la estructura sintáctica

A continuación se muestra la manera en que algunos elementos de oraciones en lenguaje natural, se representan en grafos conceptuales:

1. Los sustantivos, verbos y adjetivos y adverbios corresponden a los tipos de los nodos concepto:

   gato ⇨ [gato]
   cazar ⇨ [cazar]
   rápidamente ⇨ [rápidamente]

[19] Montes y Gómez, Manuel, *Op,. Cit.*

2. Los nombres propios se representan como el referente de los nodos concepto, el tipo indica la clase del objeto referenciado.

Félix ⟹ [gato: Félix]
Aguascalientes ⟹ [ciudad: Aguascalientes]

3. Las referencias definidas contextualmente corresponden a un referente con el símbolo #.

El gato ⟹ [gato: #]

El símbolo # seguido de una variable indica una co-referencia:

```
[animal: Félix *x] [gato: #*x]
```

4. Los sustantivos plurales se representan con el referente plural {*}, seguido de un indicador opcional de cantidad:

diez perros ⟹ [perro: {*}@9]

Los plurales específicos – no genéricos – y parcialmente especificados se representan de la siguiente manera.

Garfield y Félix ⟹ [gato: {Garfield, Félix}]
Garfield , Félix y otros gatos ⟹ [gato: {Garfield, Félix, *}]

Los prefijos Coll{*} y Dist{*} se usan para indicar una interpretación colectiva y distributiva de los sustantivos plurales.

Garfield y Félix maullan ⟹

[maullar]→(agnt) → [gato: Coll{Garfield, Félix}]

5. Los auxiliares como poder y deber corresponden a relaciones conceptuales como PSBL de posibilidad y OBLG de obligación. Estas relaciones afectan el contexto que encierra el grafo.

Posiblemente el gato cace un ratón

(PSBL) → [[cazar] → (agnt) → [ratón]]

6. Los tiempos de los verbos corresponden a relaciones conceptuales como (PASD) para pasado y futuro (FUTR), estas relaciones asimismo afectan el contexto que encierra el grafo.

El gato cazó un ratón

(PASD) → [[cazar] → (agnt) → [ratón]]

Puede haber varios niveles de anidamiento, por ejemplo:

El padre no debió golpear al niño

```
(PASD) →
     [(NO) →
          [(OBLG) →
          [[padre]←(agnt)←[golpear] → (ptnt) → [niño]]]]
```

7. El verbo tener, cuando es usado como verbo principal, puede corresponder a distintas relaciones conceptuales como (PART) de parte y (POSS) de posesión.

El gato tiene garras

[gato]→(poss)→[garra:{*}]]

Las terminaciones con en lenguajes con inflexiones, y el orden de las palabras en lenguajes sin inflexiones, corresponden a roles temáticos como (AGNT) agente, (PTNT) paciente, (INST) instrumento, (RCPT) recipiente, etc.

El niño partió una nuez con un martillo.

```
[PASD]→[partir]-
            (AGNT)→[niño: #]
            (PTNT)→[nuez]
            (INST)→[martillo]]
```

8. La información morfológica de las oraciones corresponde a los comentarios en los nodos concepto, Esta información es útil para tareas como traducción automática y generación de lenguaje.

5.2 Reglas para la formación de grafos

Las reglas para la composición de las relaciones conceptuales permitidas se representan mediante una gramática DCG (Gramática de Cláusulas Definidas). Esta gramática permite validar que las relaciones conceptuales establecidas sean aceptables, restringiendo las relaciones a los tipos de conceptos especificados en las reglas de formación, dependiendo de los tipos de conceptos en una estructura de jerárquica de conceptos, es decir, una ontología del dominio, representada mediante una red semántica en lenguaje Prolog. Tanto las reglas gramaticales como la ontología son extensibles y externas al programa de tal manera que pueden adecuarse al dominio que el usuario requiera.

```
% Reglas de producción

relacion --> rel_agente; rel_atributo; rel_objeto.

rel_agente      --> entidad,[agnt],animado.
rel_atributo    --> entidad,[attr],atributo.
rel_objeto      --> accion,[obj],entidad.

accion   -->[A],{word(A,accion)}.
animado  -->[A],{word(A,animado)}.
entidad  -->[E],{word(E,entidad)}.
atributo -->[A],{word(A,atributo)}.

% Reglas de inferencia para obtener los terminales

word(A,accion)  :- subc(A,accion).
word(A,animado) :- es(animado,A);subc(A,animado).
word(E,entidad) :- es(entidad,E);subc(E,entidad).
word(A,atributo):- es(atributo,E);subc(A,atributo).
```

Fig. 5.1 Muestra parcial de la gramática DCG para las relaciones permitidas.

```
% Ontología

subclase(animado,entidad).
subclase(persona,animado).
subclase(perro,animado).
subclase(bravo,atributo).
instancia('Eva',persona).
instancia('Fido',perro).
```

Fig. 5.2 Una pequeña ontología para un dominio restringido en implementada en Prolog.

La gramática aceptará la relación: [perro: Fido]→(ATRR)→[bravo] pero no la relación [persona: Eva]→(ATRR)→[persona].

5.3 El proceso de generación

El proceso genérico de transformación de un texto en lenguaje natural a grafos conceptuales consiste en analizar los árboles sintácticos de las oraciones, recorriendo el árbol de forma ascendente (*bottom-up*), identificando conceptos y estableciendo relaciones.

A continuación se muestra el proceso de transformación de una oración en un grafo conceptual.

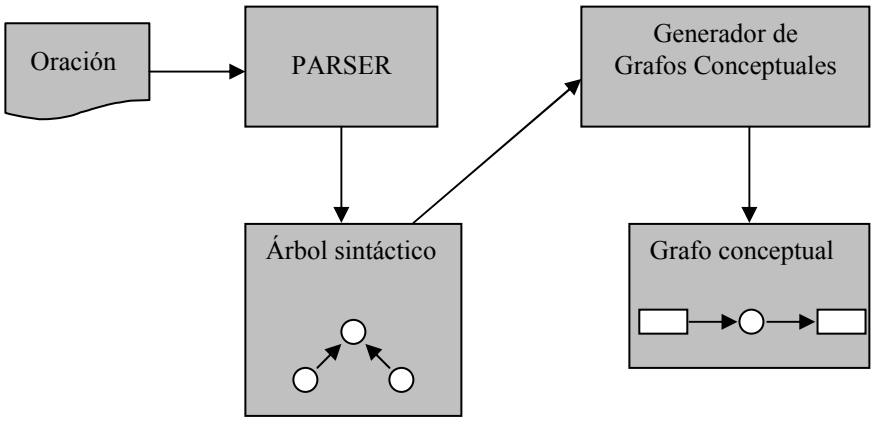


Fig. 5.3. El proceso de generación de un grafo conceptual a partir de una oración

Método de generación

El método propuesto de generación de los grafos conceptuales, está fundamentado en los árboles de dependencias. La estrategia de análisis consiste en el recorrido *descendente* del árbol dependencias proporcionado por el programa Parser 1.0. En la figura 5.4 se muestra el árbol de dependencias para la oración *un gato negro caza un ratón blanco.*

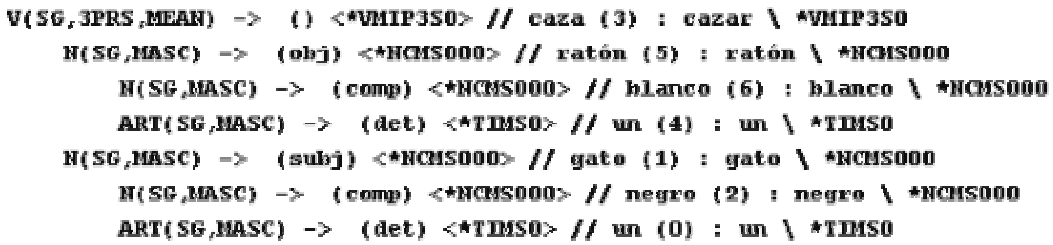


Fig. 5.4. Árbol de dependencias para la oración: *un gato negro caza un ratón blanco.*

El método básico de generación es el siguiente:

- Se inicia por la raíz del árbol (que en el caso del árbol de dependencias corresponde al verbo principal), como la cabeza del grafo conceptual. La información morfológica de este permiten establecer relaciones del tipo (PAST) o (FUTR) para el grafo conceptual.

- Se analiza cada uno de los hijos y se identifica la relación sintáctica entre el nodo analizado y su padre, existen las siguientes relaciones para la palabra dependiente:

  - dobj (objeto directo),
  - subj (sujeto),
  - obj (objeto indirecto),
  - det (modificador que es un artículo o un pronombre),
  - adver (adverbial),
  - cir (circunstancial),
  - prep (preposicional),
  - mod (modificador que no es un artículo o un pronombre),
  - subord (subordinativa),
  - coord (coordinativa).

- Para cada relación detectada se realiza la validación mediante la gramática de relaciones permitidas.

El resultado de este proceso se muestra en la figura 5.5.

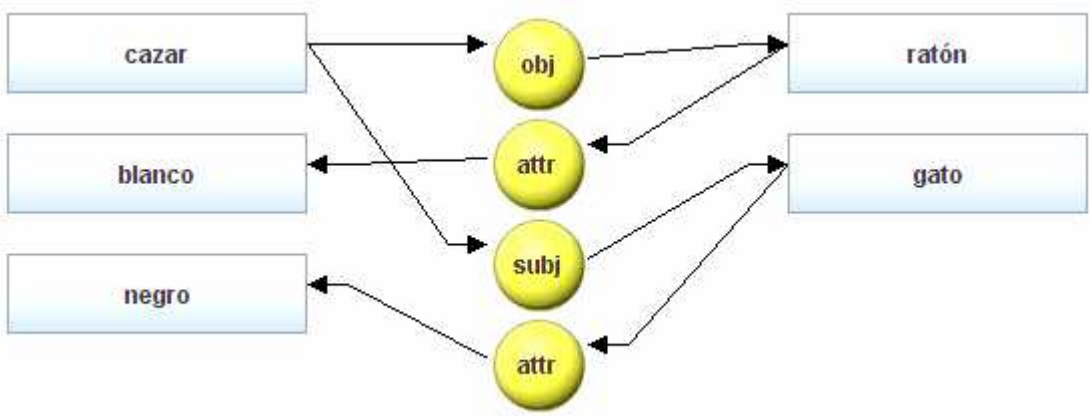


Fig. 5.5 Grafo conceptual generado para la oración: *un gato negro caza un ratón blanco*

```
[cazar: *1;v][ratón: *2;n][blanco: *3;a][gato: *4;n][negro: *5;a]
(obj ?1?2)(attr ?2?3)(subj ?1?4)(attr ?4?5)
```

Fig. 5.6 Grafo conceptual generado en formato CGIF para la oración: *un gato negro caza un ratón blanc*

VI. Implementación

6.1 Introducción

La aplicación fue implementada en lenguaje Java, permitiendo una integración sencilla con herramientas preexistentes cono TuProlog, así como la posibilidad de extender la aplicación en un futuro, utilizando las API desarrolladas por la comunidad de los grafos conceptuales, tales como Notio y Prolog+CG.

6.2 Estructura de la aplicación

La estructura de la aplicación de muestra en la figura 6.1.

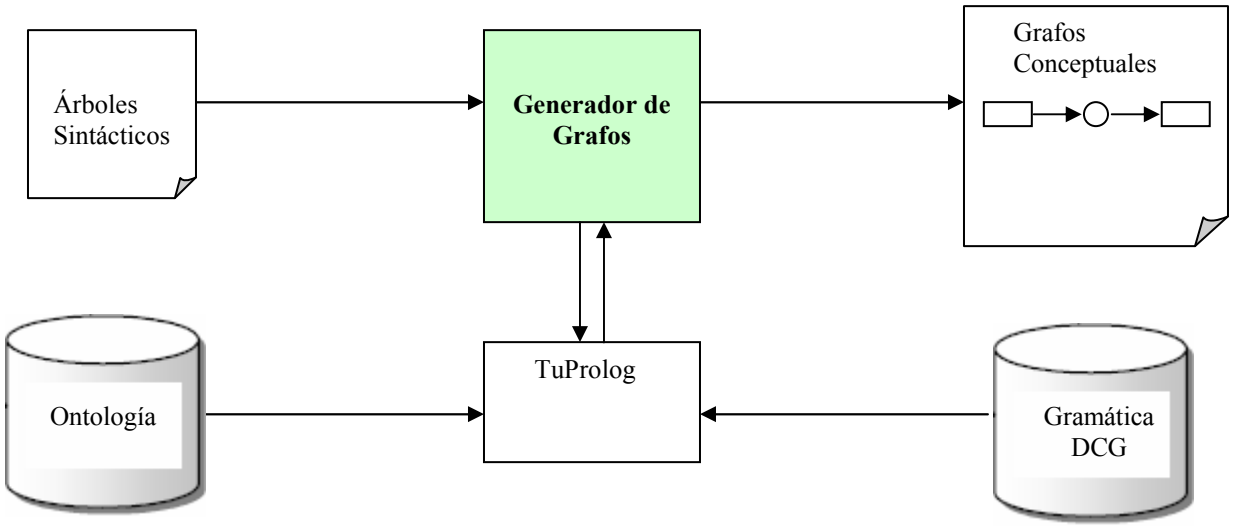


Fig. 6.1 Estructura de la aplicación.

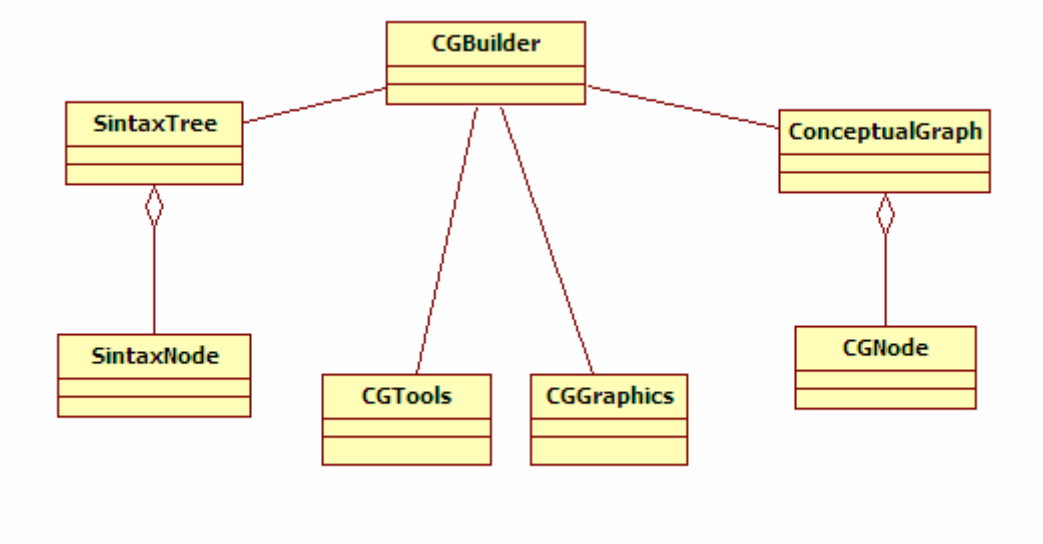


Fig. 6.2 Diagrama de clases

6.2.1 Principales clases

CGBuilder. Es la clase principal de la aplicación, su responsabilidad es la interacción con el usuario a través de la interfaz gráfica y la generación de los grafos conceptuales.

SintaxTree. Modela el árbol sintáctico de entrada para su análisis, está compuesto por varios *SintaxNode*.

ConceptualGraph. Modela el grafo conceptual generado, el cual está a su vez compuesto por varias instancias de la clase *CGNode*.

CGTools. Clase que presta servicios a CGBuilder, incluye métodos estáticos para recuperar los archivos y guardar los resultados, así como para trasformar la entrada en formato de texto plano a su correspondiente representación interna mediante la clase *SintaxTree*.

CGGraphics. Esta clase tiene la responsabilidad de mostrar la salida gráfica al grafo conceptual.

El la siguiente puede verse la ventana principal de la aplicación, en ella se muestra la oración a analizar, el árbol sintáctico de entrada en el formato original entregado por el Parser 1.0, y los resultados de la generación, en formato CGIF y en formato gráfico.

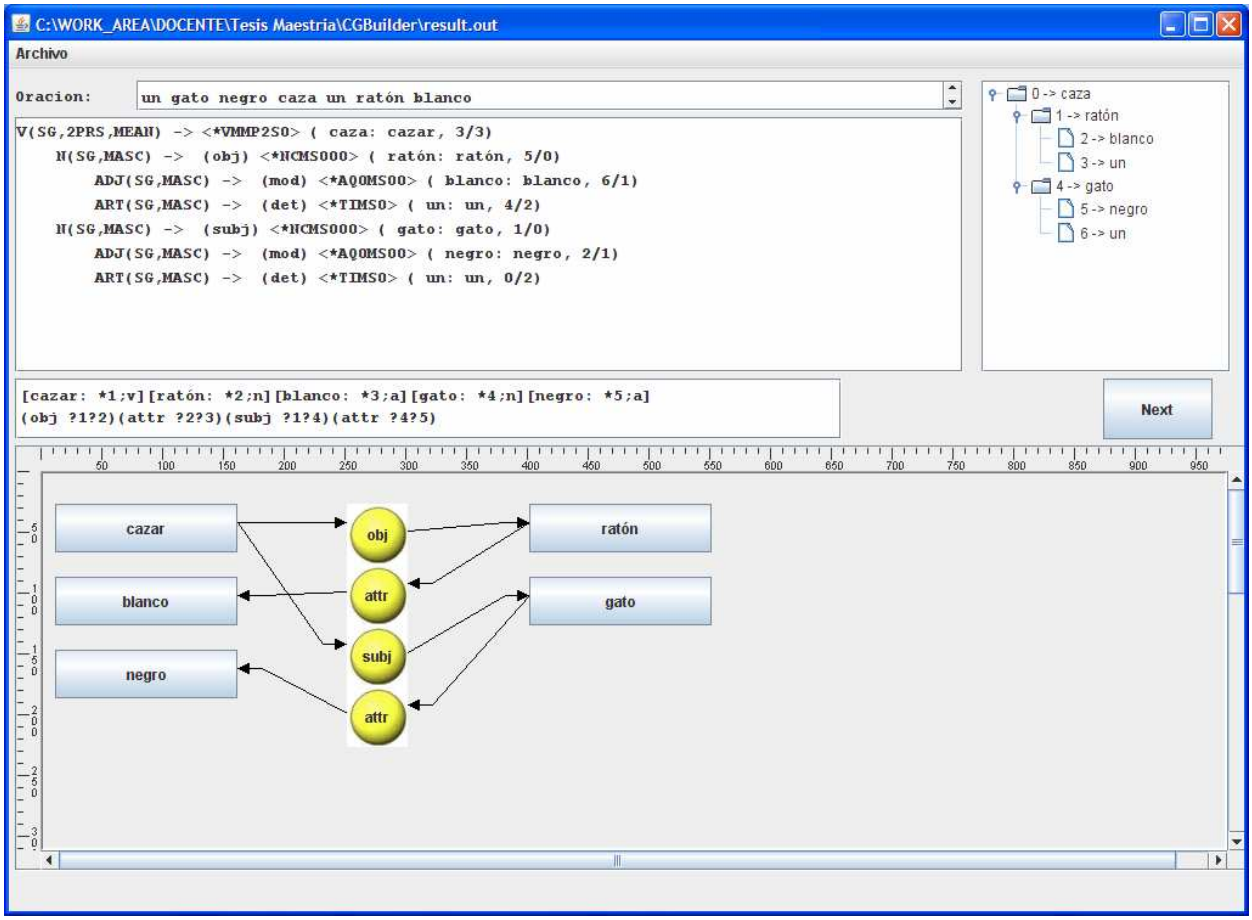


Fig. 6.3 Ventana principal de la aplicación

6.3 Formatos

6.3.1 Formato del archivo de entrada

El analizador sintáctico Parser 1.0 entrega como salida un archivo en texto plano que tiene la siguiente estructura.

```
+-------------------------------------------------------------------
|un gato negro caza un ratón blanco .
+-------------------------------------------------------------------
un (0) un (*MCMS00)  (0) un (*NP00000)  (1) un (*TIMS0)  (2)
gato (1) gato (*NCMS000)  (0) gato (*AQ0MS00)  (1) gato (*NP00000)  (2)
negro (2) negro (*NCMS000)  (0) negro (*AQ0MS00)  (1) negro (*NP00000)
(2)
caza (3) caza (*NCMS000)  (0) caza (*NCFS000)  (1) caza (*NP00000)  (2)
cazar (*VMMP2S0)  (3) cazar (*VMIP3S0)  (4)
un (4) un (*MCMS00)  (0) un (*NP00000)  (1) un (*TIMS0)  (2)
ratón (5) ratón (*NCMS000)  (0)
blanco (6) blanco (*NCMS000)  (0) blanco (*AQ0MS00)  (1) blanco
(*NP00000)  (2)
. (7) . (*FP)  (0)

Parsing  8 words, total variants: 6

1:
329  V(SG,2PRS,MEAN) -> <*VMMP2S0> ( caza: cazar, 3/3)
147      N(SG,MASC) ->  (obj) <*NCMS000> ( ratón: ratón, 5/0)
43           ADJ(SG,MASC) ->  (mod) <*AQ0MS00> ( blanco: blanco, 6/1)
239          ART(SG,MASC) ->  (det) <*TIMS0> ( un: un, 4/2)
147      N(SG,MASC) ->  (subj) <*NCMS000> ( gato: gato, 1/0)
43           ADJ(SG,MASC) ->  (mod) <*AQ0MS00> ( negro: negro, 2/1)
239          ART(SG,MASC) ->  (det) <*TIMS0> ( un: un, 0/2)
113      $PERIOD -> <*Fp> ( .: ., 7/0)
```

Fig. 6.4 Formato del archivo de entrada.

En la primera parte, en el encabezado está la oración analizada, la segunda parte es la sección de información morfológica y por último la sección de árboles de dependencia, en esta parte se enlistan todas las variantes que el analizador sintáctico encontró para la oración. Al cargar un archivo, el generador de grafos analiza el primer árbol que se encuentre.

6.3.2 Formatos del archivo de salida

Los grafos conceptuales están definidos en una sintaxis abstracta que es independiente de cualquier notación, pero el formalismo puede ser representado en muchas formas concretas diferentes. En el capítulo IV se mostró la notación gráfica (*display form*) y la notación lineal *(linear form*). En este apartado se mostrará la notación CGIF (*Conceptual Graph Interchange Form)* que es la que se utiliza como salida del programa que se propone en este trabajo.

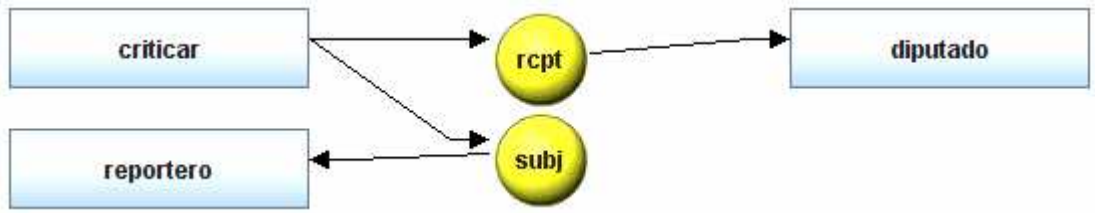


Fig. 6.5 Un grafo conceptual en forma gráfica

```
[criticar: *1;v][diputado: *2;n][reportero: *3;n]
(rcpt ?1?2)(subj ?1?3)
```

Fig 6.6 Grafo conceptual equivalente al anterior en formato CGIF.

El formato CGIF es muy simple, representa los nodos concepto con corchetes, incluyendo un índice e información morfológica, este índice se anota en los nodos relación para indicar qué con conceptos está ligada la relación expresada mediante paréntesis. Este formato se eligió por su simplicidad y por ser el formato que utilizan las herramientas de minería de texto desarrolladas en el laboratorio de Laboratorio de Lenguaje Natural por el Dr. Manuel Montes y Gómez.

6.3.3 Formato de la gramática DCG

```
% Reglas de producción

relacion        --> rel_agente; rel_atributo; rel_objeto.
rel_agente      --> entidad,[agnt],animado.
rel_atributo    --> entidad,[attr],atributo.

% Terminales

accion   -->[A],{word(A,accion)}.
animado  -->[B],{word(B,animado)}.
atributo -->[C],{word(C,atributo)}.

% Reglas de inferencia

word(A,accion)  :- subc(A,accion).
word(B,animado) :- es(animado,B);subc(B,animado).
word(A,atributo):- es(atributo,C);subc(C,atributo).
```

Fig 6.7 Formato de la gramática DCG

Como ya se mencionó, DCG es una extensión de las gramáticas libres de contexto implementadas en PROLOG. DCG extiende la sintaxis de PROLOG introduciendo el operador `-->` para escribir las reglas de producción. Los terminales de la gramática se escriben con símbolos entre corchetes [ ], asimismo, se permite utilizar predicados para la obtención de los símbolos terminales. En la gramática de la figura 4.10 puede observarse que el símbolo no terminal *acción* genera un terminal variable [**A**], el cual será obtenido como resultado de unificar tal variable en el predicado `word,` para ello se utilizan las reglas de inferencia que se muestran en la parte inferior de la figura. Por ejemplo, para unificar una palabra **A** como *acción* es necesario que **A** se encuentre en la ontología como una subclase de *acción* y para unificar la palabra **A** como animado es necesario que **A** se encuentre en la ontología como una subclase de *animado* o bien sea una instancia de *animado*.

6.3.4 Formato de la ontología

Para representar la ontología se utilizan 2 tipos de predicado PROLOG:

- `subclase(hiponimo,hiperonimo)` para indicar una relación "parte-de".
- `es(instancia, clase) para` expresar una relacion "es-un".

```
% Ontología

% Relaciones semánticas

subclase(animado,entidad).
subclase(persona,animado).
subclase(perro,animado).
subclase(bravo,atributo).
instancia('Eva',persona).
instancia('Fido',perro).

% Reglas de inferencia

es(Clase,Obj):- instancia(Obj,Clase).
es(Clase,Obj):- instancia(Obj,Clasep),subc(Clasep,Clase).

subc(C1,C2):- subclase(C1,C2).
subc(C1,C2):- subclase(C1,C3),subc(C3,C2).
```

Fig 6.8 Ontología

En la figura se muestran también las reglas de inferencia para restreas la clase de una instancia o bien la superclase de una clase dada.

6.4 Integración de las herramientas

6.4.1 TuProlog

TuProlog implementa una máquina PROLOG en una clase Java. Esto posibilita que cualquier clase en Java pueda utilizar los servicios de una máquina PROLOG. TuProlog incluye la clase Theory para almacenar los un conjunto de predicados y reglas, la teoría, esta clase puede ser creada a partir de un archivo de codigo PROLOG preexistente o bien crearse a partir de una biblioteca de teorías, tal como DCG.

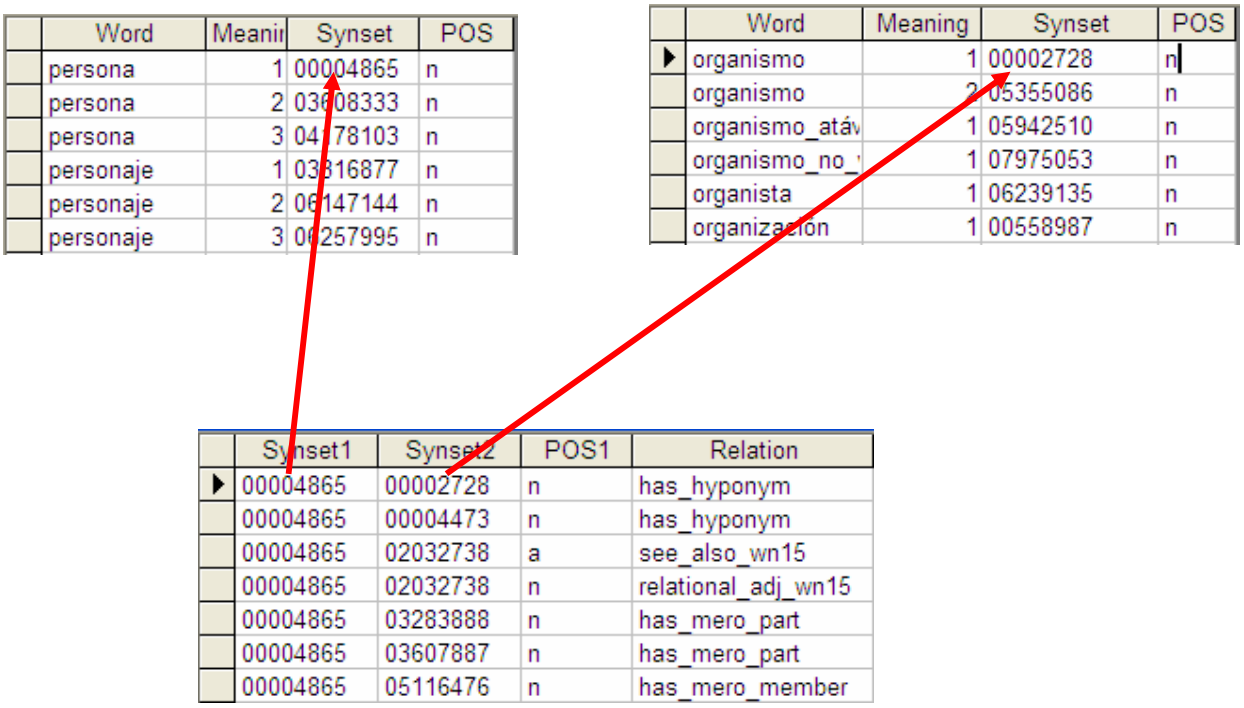
```
Prolog engine   = new Prolog();
DCGLibrary dcg  = new DCGLibrary(); // Carga DCG
Theory tdcg     = new Theory(dcg.getTheory());
Theory tgram    = new Theory(new FileInputStream("gram.pl"));
tdcg.append(tgram);
engine.setTheory(tw);
String question  = "phrase(relacion(R),[perro, attr, bravo]).";
SolveInfo answer = engine.solve(question);

if(answer.isSuccess())  // la frase puede ser generada por DCG
{
     // Hacer algo
}
```

Fig. 6.9 Llamando a TuProlog

Para utilizar una gramática DCG mediante TuProlog, basta con construir el texto de la consulta y enviarsela a la máquina PROLOG, utilizando el predicado `phrase`, predifinido en DCG. Bajo este esquema, se mantienen tanto la gramática DCG como la ontología externas a la aplicación principal, permitiendo la flexibilidad de adaptarse a un dominio específico.

6.4.2 WordNet

WordNet se utiliza como base para la construcción de la ontología en PROLOG. Básicamente se tranforma el formato de WordNet al formato de la ontología, restringiendose a las relaciones de tipo hiperónimo/hipónimo.



Word	Meanir	Synset	POS
persona	1	00004865	n
persona	2	03608333	n
persona	3	04178103	n
personaje	1	03816877	n
personaje	2	06147144	n
personaje	3	00257995	n

Word	Meaning	Synset	POS
organismo	1	00002728	n
organismo	2	05355086	n
organismo_atáv	1	05942510	n
organismo_no_'	1	07975053	n
organista	1	06239135	n
organización	1	00558987	n

Synset1	Synset2	POS1	Relation
00004865	00002728	n	has_hyponym
00004865	00004473	n	has_hyponym
00004865	02032738	a	see_also_wn15
00004865	02032738	n	relational_adj_wn15
00004865	03283888	n	has_mero_part
00004865	03607887	n	has_mero_part
00004865	05116476	n	has_mero_member

Fig. 6.10 Relaciones en WordNet

La relación de WordNet indicada en la figura se expresara en la ontología mediante el predicado:

```
subclase(persona,organismo).
```

VII. Resultados obtenidos

7.1 Ejemplos de aplicación del algoritmo

7.1.1 Muestras de grafos conceptuales generados

A continuación se muestran algunos ejemplos de los grafos conceptuales generados por la aplicación.

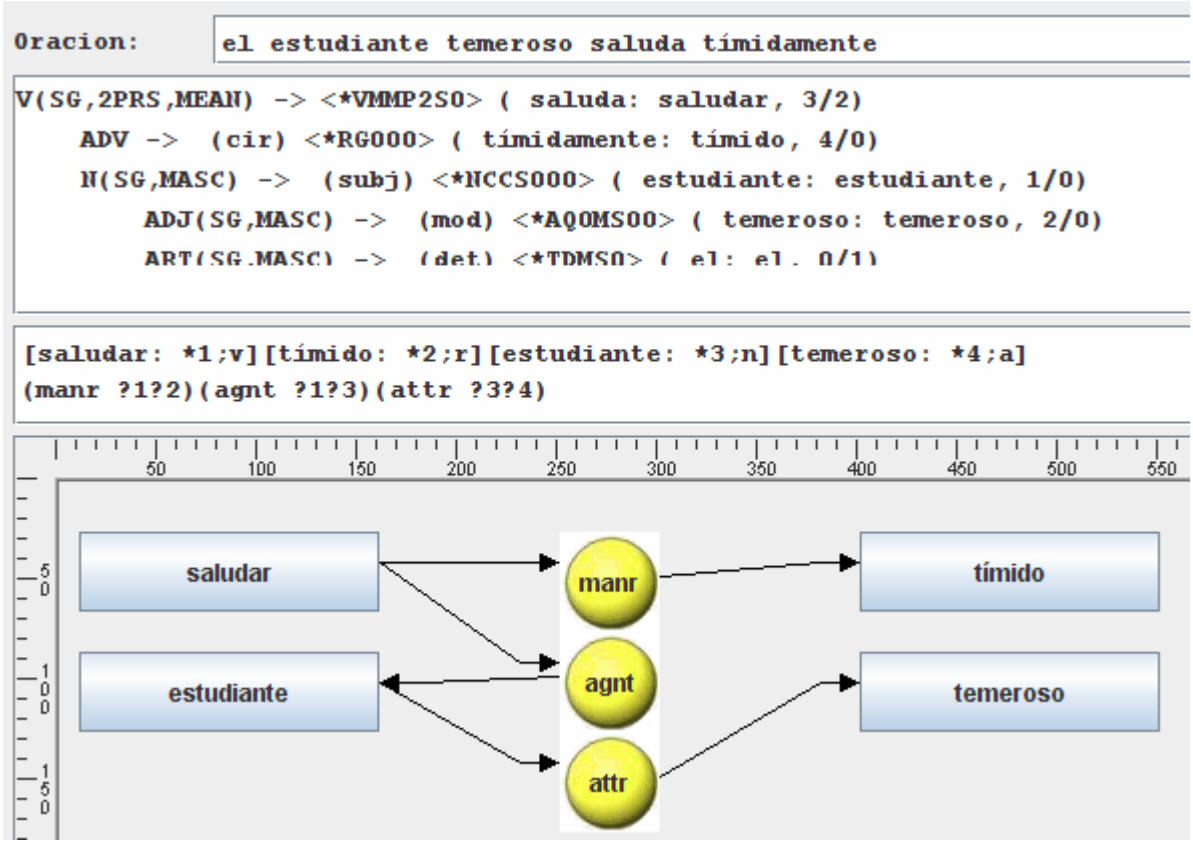


Fig. 7.1 Grafo conceptual para una oración.

El grafo mostrado en la figura 7.1 representa los conceptos y las relaciones detectadas en la oración *el estudiante temeros saluda tímidamente*, la cabeza del grafo es la acción *saludar*, el agente que lleva a cabo la acción es el estudiante, la acción se realiza de manera *tímida*, el grafo también indica que el estudiante tiene el atributo *temeroso*. Los adjetivos fueron mapeados como atributos de la palabra de la que dependen, y el adverbio es representado en al grafo conceptual como la manera en que la acción fue llevada a cabo.

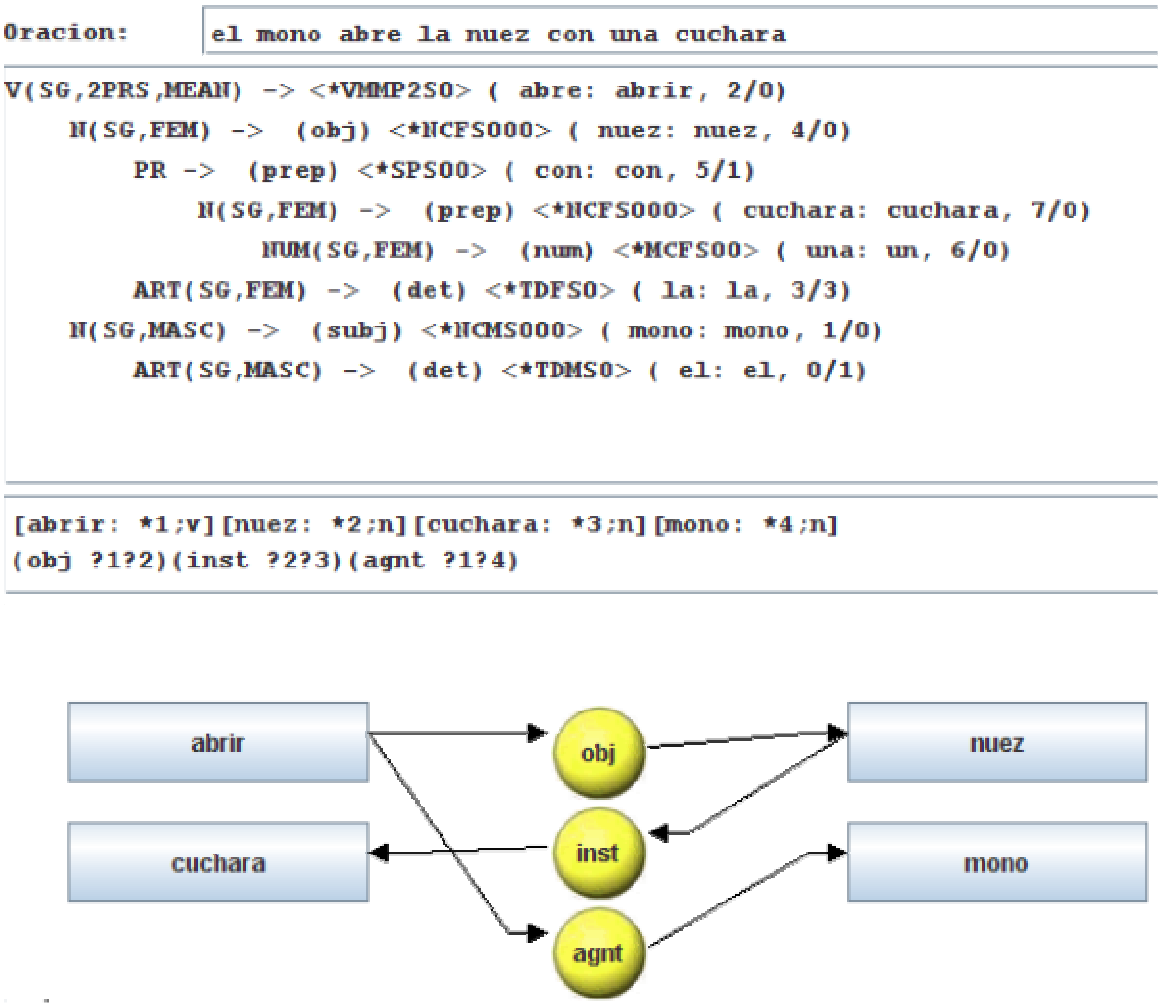


Fig. 7.2 Grafo conceptual para una oración.

En el segundo ejemplo, se muestra el grafo de la oración *el mono abre la nuez con una cuchara*, la cabeza del grafo es la acción a*brir*, el agente es *el mono*, el objeto es la *nuez* y el instrumento es una *cuchara.*

7.1.2 Prueba con el programa de comparación de grafos conceptuales para minería de texto

Los grafos mostrados a continuación se utilizaron como entrada del programa de comparación de grafos conceptuales cuyo objetivo es comparar grafos encontrando la medida de similitud.

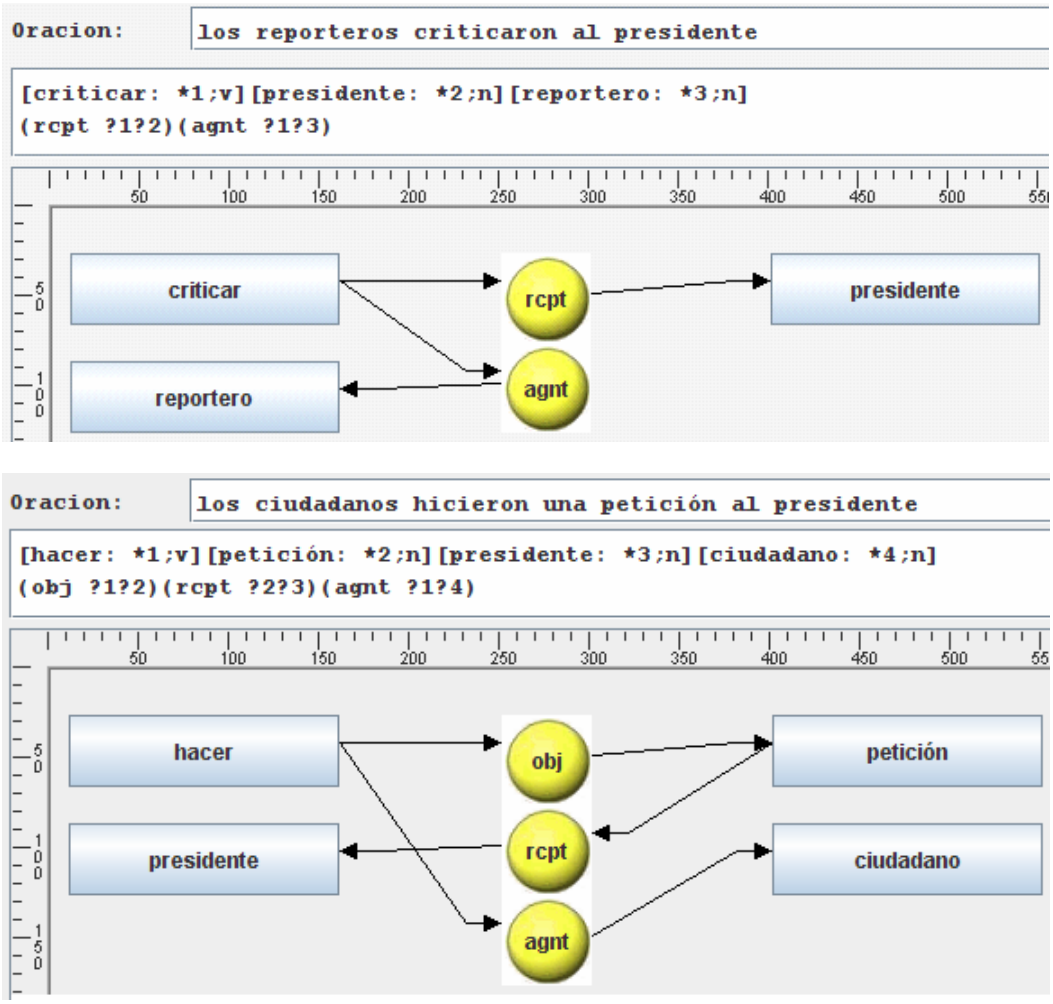


Fig. 7.3 Dos grafos conceptuales generados.

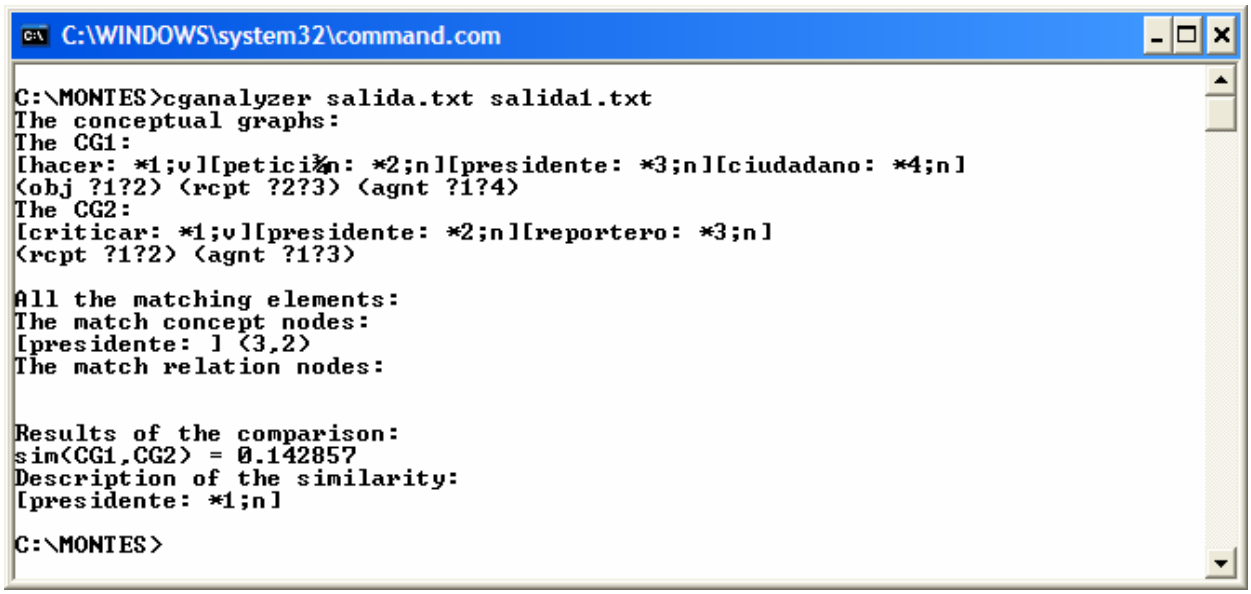


Fig. 7.4 Resultado de comparación de los grafos de la figura **7.3**

7.2 Discusión de los resultados.

Una inspección de los resultados mostrados arriba demuestra que el programa construye los grafos correctamente, dando los resultados que concuerdan con la intuición humana. Sin embargo, la calidad de los resultados que arroja la aplicación estará estrechamente ligada a la de los árboles que le entrega el analizador sintáctico. Lo consideramos más bien como una ventaja de nuestra aproximación, ya que nuestro convertidor podrá ser compatible con las futuras versiones de los analizadores sintácticos y así dar aún mejores resultados.

El Parser puede obtener más de un árbol sintáctico para una oración. El generador de grafos puede entonces generar el grafo correspondiente a cada variante del árbol sintáctico que el Parser genere para cada oración. En este modo muestro programa podrá ser usado para una mejor desambiguación de la estructura sintáctica, con un fundamento semántico más sólido que los métodos puramente estadísticos existentes.

VIII. Conclusiones y trabajo futuro

8.1 Conclusiones

Se desarrolló un programa que genera grafos conceptuales tomando como entrada árboles sintácticos. Esto permite aplicarlos a tareas que aprovechen la riqueza expresiva de los grafos conceptuales como representación del contenido de textos. Aplicaciones como la minería de texto y la recuperación de información pueden hacer uso de esta representación. Asimismo este trabajo es un paso a procesamiento semántico del texto en español y probablemente permitirá un avance en muchas otras aplicaciones que requieren de un tratamiento semántico y no puramente estadístico.

Las **aportaciones** específicas de este trabajo incluyen:

- El desarrollo del método para obtener la representación semántica (grafos conceptuales) del texto en español.
- Demostración de la utilidad de este método para las aplicaciones, usando como ejemplo la minería de texto.
- Uso de gramáticas de dependencia como la base sintáctica para el método, lo que permitirá alcanzar mejor calidad de los resultados según el desarrollo de los analizadores sintácticos de dependencia (tales como por ejemplo DILUCT).
- Desarrollo de una arquitectura desacoplada del convertidor semántico compatible, en principio, con diferentes analizadores sintácticos, existentes o futuros.

8.2 Trabajo Futuro

Como un trabajo futuro para mejorar el método presentado aquí, se planean las siguientes **acciones inmediatas**:

– Evaluación más rigurosa: hacer manualmente un corpus de grafos conceptuales y comparar cuantitativamente la salida de nuestro programa con lo hecho a mano;

– Resolución de correferencia y unión de los grafos de diferentes oraciones: identificar los nodos (palabras o frases) que refieren a las mismas entidades o acciones; estos nodos se representan con un solo nodo de la estructura semántica resultante;

– Aplicar los resultados del programa a otros tipos de tareas y sistemas. Específicamente, integrar el programa con el Prolog+CG, para poder hacer inferencias lógicas sobre los grafos.

A más largo plazo, de esta tesis emanarían las siguientes **líneas de investigación**:

– Desarrollar las aplicaciones clásicas de procesamiento de lenguaje natural (tales como la recuperación de información, respuesta a preguntas, traducción automática, etc.) basadas en la representación semántica y simbólica del texto, y no puramente estadística;

– Aplicar el método a las situaciones menos tradicionales, tales como el diálogo con el usuario en la elicitación de requerimientos de software;

– Estudiar los efectos de las tareas del procesamiento lingüístico al comportamiento del convertidor –por ejemplo, la calidad de la resolución de anáfora, desambiguación sintáctica, etc.

– Aplicar el razonamiento sobre los grafos conceptuales a las tareas propias del procesamiento de texto, tales como resolución de ambigüedad (sintáctica, de referencia, del sentido de las palabras, etc.).

Referencias

1. Abney, S. P. *Parsing by chunks*. In R. C. Berwick, S. P. Abney, and C. Tenny (Eds.) Principle-Based Parsing: Computation and Psycholinguistics. Kluwer, Dordrecht, 257-278, 1991.
2. Allen, J. F. *Natural Language Understanding.* Benjamin Cummings, 1995.
3. Bolshakov, Igor., Gelbukh, Alexander. *Computational Linguistics. Models, Resources, Applications*. IPN. México, 2004.
4. Bresnan, J. W., editor. *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA. 1982.
5. Chomsky, N. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, MA. 1965.
6. Chomsky, N. *Syntactic Structures*. The Hague: Mouton & Co, 1957.
7. Church, K. and Patil, R. *Coping with syntactic ambiguity or how to put the block in the box on the table.* Computational Linguistics 8, 139-149, 1982.
8. Collins, M. *Head-driven Statistical Models for Natural language parsing.* Ph.D. Thesis University of Pennsylvania.
9. *DG Website Dependency-Based Approaches to Natural Language Syntax.* ufal.mff.cuni.cz/dg/%20dgmain.html 1999.
10. Fraser, N. *Dependency parsing,* PhD thesis, UCL, London, 1994.
11. Galicia-Haro Sofía N., *Análisis sintáctico conducido por un diccionario de patrones de manejo sintáctico para lenguaje español.* Tesis doctoral, CIC, IPN, México, 2000.
12. Galicia-Haro Sofía N., Bolshakov I. A. y Gelbukh A. F. *Un modelo de descripción de la estructura de las valencias de verbos españoles para el análisis automático de textos.*1999
13. Galicia-Haro Sofía N., Gelbukh A. F. y Bolshakov I. A. *Una aproximación para resolución de ambigüedad estructural empleando tres mecanismos diferentes.* J. Procesamiento de Lenguaje Natural, No 27, September 2001. SEPLN, Spain, 55-64, 2001.

14. Gelbukh, A y Grigori Sidorov, *Procesamiento automático del español con enfoque en recursos léxicos grandes,* IPN, 2006
15. Gelbukh, Alexander. *Computational Processing of Natural Language: Tasks, Problems and Solutions.* Congreso Internacional de Computación en México D.F., Nov 15-17, 2000.
16. Gelbukh, Alexander. *Using a semantic network for lexical and syntactical disambiguation.* CIC-97, nuevas aplicaciones e Innovaciones Tecnológicas en Computación, Simposio Internacional de Computación, Mexico City, Mexico, pp. 352-366, 1997.
17. Grihsman R, *Computacional Linguistics*, Cambrigde University Press, 1986.
18. Hensman, Svetlana, Construction of Conceptual Graph representation of texts.
19. Hirst, Graeme. *Semantic interpretation and the resolution of ambiguity.* Studies in Natural Language Processing. Cambridge University Press, Cambridge, United Kingdom, 263. 1987.
20. Hudson, R. A. (eds.) *Dependency and Valency.* An International Handbook of Contemporarary Research. Berlin: Walter de Gruyter. www.phon.ucl.ac.uk/home/dick/wg.htm 1998.
21. Lombardi, V., L. Lesmo. *Formal Aspects and Parsing Issues of dependency theory.* In Proceedings International Conference COLING-ACL'98. August 10-14 Quebec, Canada, 787-793, 1998.
22. Mel'cuk, I. A. and A. K. Zolkovsky. *Towards a functioning meaning-text model of language.* Linguistics 57: 10- 47, 1970.
23. Mel'cuk, I. A. *Dependency Syntax.* In P. T. Roberge (ed.) Studies in Dependency Syntax. Ann Arbor: Karoma 23-90, 1979.
24. Mel'cuk, I. *Dependency Syntax: Theory and Practice*. New York: State University of New York Press, 1988.
25. Montes y Gómez, Manuel, Minería de texto empleando la semejanza entre estructuras semánticas Tesis Doctoral, CIC-IPN, 2002
26. Moreno Ortiz, Antonio, *Estudios de Lingüística Española***.**
27. Real Academia Española, *Esbozo de una Nueva Gramática de la Lengua Española*, Madrid, 1973.

28. Resnik, P. and Hearst, M. *Syntactic ambiguity and conceptual relations.* In: K. Church(ed.) Proceedings of the ACL Workshop on Very Large Corpora, 58-64, 1993.
29. Saussure Ferdinand, *Curso de lingüística general,* Fontamara*, 1980.*
30. Sells, P. *Lectures on Contemporary Syntactic Theories.* CSLI Lecture Notes, Stanford, CA. Number 3, 1985.
31. Sowa, J.F, *Conceptual Structures*, Addison-Wesley, 1984.
32. Steele, J. *Meaning - Text Theory.* Linguistics, Lexicography, and Implications. James Steele, editor. University of Ottawa press, 1990.
33. Tapanainen, P., Järvinen, T., Heikkilä, J., Voutilainen. A. *Functional Dependency Grammar*. www.ling.helsinki.fi/~tapanain/dg/ 1997.
34. W3C*, OWL Web Ontology Language Use Cases and Requirements,* Word Wide Web Consortium (W3C), 2004.
35. Yuret, D. *Discovery of Linguistic Relations Using Lexical Attraction.* Ph. D. thesis. Massachusetts Institute of Technology, 1998.
36. *Zhang Lei y Yong Yu, Learning to Generate Conceptual Graphs from Domain Specific Sentences.*