INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Laboratorio de Lenguaje Natural
y Procesamiento de Texto

DESAMBIGUACIÓN DE SENTIDOS DE PALABRAS USANDO RELACIONES SINTÁCTICAS COMO CONTEXTO LOCAL

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

ING. JAVIER TEJADA CÁRCAMO

DIRECTOR: DR. ALEXANDER GELBUKH

México, D.F.
Mayo, 2006

SIP-14

INSTITUTO POLITECNICO NACIONAL
SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 17:00 horas del día 15 del mes de Diciembre de 2005 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:

"DESAMBIGUACIÓN DE SENTIDOS DE PALABRAS USANDO RELACIONES SINTACTICAS COMO CONTEXTO LOCAL"

TEJADA	CÁRCAMO	JAVIER
Apellido paterno	materno	nombre(s)

Con registro: B 0 3 1 2 2 2

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. Grigori Sidorov

Primer vocal
(Director de Tesis)

Dr. Alexander Guelboukh

Segundo vocal

Dra. Sofía Natalia Galicia Haro

Tercer vocal

M. en C. Alejandro Bolello Castillo

Suplente

M. en C. Miguel Jesús Torres Ruiz

INSTITUTO POLITECNICO NACIONAL SECRETARIA DE INVESTIGACION Y POSGRADO 30 ENE 2006 RECIBIDO DIRECCION DE POSGRADO

EL PRESIDENTE DEL COLEGIO

Dr. Hugo César Coyote Estrada

INSTITUTO POLITECNICO NACIONAL CENTRO DE INVESTIGACION EN COMPUTACION DIRECCION

A mis padres Mary y Mateo,
que aunque lejos los llevo en el corazón.

A mi esposa Margaret y mi hija Fernanda,
fuente inagotable de inspiración
en mi diario quehacer.

Agradecimientos

Es difícil asimilar lo rápido que pasa el tiempo. A veces creo que sólo han pasado unos meses desde que salí de mi país, Perú, y empecé esta maestría; sin embargo, ya han transcurrido casi tres años. Comparo al ingeniero de ayer con el Maestro en Ciencias de hoy, y estoy seguro que valió la pena tanto esfuerzo y sacrificio. Esta sensación, la debo en gran parte a muchas personas e instituciones. De manera muy especial quiero agradecer a las siguientes:

A mi asesor, el *Dr. Alexander Gelbukh*, por su apoyo incondicional, sus múltiples consejos y sobretodo sus valiosas enseñanzas.

A los doctores *Grigori Sidorov*, *Igor Bolshakov*, *Mikhail Alexandrov* del Centro de Investigación en Computación, *Ernesto Cuadros* de la Universidad San Pablo de Perú y a la doctora *Sofía Galicia* de la Universidad Nacional Autónoma de México, por sus consejos, críticas y apoyo moral que me ofrecieron en todo momento.

A mi amada esposa, *Margaret*, y mi linda hija, *Fernanda*. Creo que el amor que me brindaron fue el arma principal que me permitió salir adelante, superando cuantos obstáculos se me presentaron en el camino.

A mis queridos padres, *Mary* y *Mateo*, por sus enseñanzas, su amor, su dedicación y fundamentalmente, por esa férrea disciplina y responsabilidad que me inculcaron desde pequeño. Espero algún día poder educar a mis hijos como ellos lo hicieron conmigo.

A mis hermanos, *Wily* y *Claudia*, por todo el cariño y apoyo, que pese a la distancia, me dieron. Siempre recuerdo los buenos momentos que pasamos juntos.

A mis compañeros y amigos mexicanos, peruanos, cubanos y españoles, que de una u otra manera influyeron directamente en mi formación profesional.

Al *Centro de Investigación en Computación*, al *PIFI* del Instituto Politécnico Nacional y al *CONACYT*, por el apoyo económico que me otorgaron durante estos tres años. Sin su ayuda hubiera sido imposible concluir mis estudios de maestría.

A este gran país, México, por la oportunidad que me dio no sólo de hacer la maestría, sino también de conocer sus costumbres y tradiciones.

Resumen

La desambiguación de sentidos de palabras consiste en determinar el significado de una palabra ambigua dentro de un contexto específico. Éste es un problema muy complejo, el cual tiene que ser resuelto para satisfacer las necesidades de otras áreas del procesamiento del lenguaje natural. Desde el principio de los tiempos, algoritmos supervisados y no supervisados han intentado solucionar este problema siendo los primeros los que han logrado mejores resultados. Sin embargo, las grandes cantidades de información requeridas para desambiguar un vocablo superan la capacidad de procesamiento de los sistemas supervisados. Ante dicha situación surgen los algoritmos no supervisados como una alternativa para etiquetar semánticamente un vocablo ambiguo. En esta tesis, se presenta un algoritmo no supervisado que se basa en la similitud de sentidos y la semejanza de contextos. El algoritmo confía en la intuición: dos palabras diferentes expresan significados semejantes si ocurren en contextos similares. Con la ayuda de un analizador sintáctico y un corpus se crea un recurso de vocablos relacionados sintácticamente, los cuales se comparan con las características sintácticas existentes en el contexto local de un vocablo ambiguo, obteniendo términos similares a éste, de tal manera que la similitud entre dichos términos y el vocablo ambiguo es evaluada usando diferentes métricas de similitud. Finalmente, un algoritmo que maximiza dicha similitud y que acumula puntos para cada sentido del vocablo ambiguo, elige el que obtuvo mayor puntaje como etiqueta semántica de dicho vocablo.

Abstract

Word Sense Disambiguation (WSD) is the task of determining the meaning of an ambiguous word in a given context. It is an open problem that has to be solved in order to meet the needs of other natural language processing tasks. Supervised and unsupervised algorithms have been tried throughout the WSD research history. Until recently, supervised systems have been achieving the best accuracy. However, these systems are limited by the need in very expensive manual data preparation. In order to advance in WSD, benefits of unsupervised systems should be examined. In this thesis, an unsupervised algorithm based on sense similarity and syntactic context is presented. The algorithm relies on the intuition that two different words are likely to have similar meanings if they occur in similar local contexts. With the help of a principle-based broad coverage parser, a one-million-word training corpus is parsed and local context features are extracted based on some rules. Similarity values between the ambiguous word and the words that occurred in a similar local context as the ambiguous word are evaluated. Based on a similarity maximization algorithm, polysemous words are disambiguated.

Contenido

Vista general de la tesis

Índice detallado de la tesis

Índice de figuras

Índice de tablas

Capítulo 1
Introducción

El lenguaje natural es parte integral de nuestras vidas, siendo éste el principal vehículo usado por los seres humanos para comunicarse e intercambiar información. Tiene el potencial de expresar una gran cantidad de ideas; incluso elaborar y comprender pensamientos muy complejos. La lingüística computacional tiene por objetivo capturar este poder, suministrando la funcionalidad necesaria a computadoras para que éstas puedan analizar y procesar lenguaje natural, y de manera análoga, intenta comprender cómo las personas lo hacen. Actualmente, existen varias tareas lingüísticas automatizadas; tales como la traducción de texto, la corrección tipográfica, ortográfica, gramatical y de estilo, la sintetización del habla, los resúmenes de documentos, la clasificación de textos, etc.

La lingüística computacional es la ciencia encargada de implementar, investigar y perfeccionar dichas aplicaciones. Actualmente, se han obtenido algunos resultados favorables; sin embargo, un número considerable de problemas aún no han sido resueltos, entre los cuales destaca la ambigüedad de sentidos de palabras, la cual se ha transformado en un gran reto para esta ciencia. En términos generales, es posible afirmar que los seres humanos no se percatan de la existencia de ambigüedades en el lenguaje, las cuales resuelven casi inconcientemente usando el contexto, el conocimiento que poseen y la realidad en la que viven. Desafortunadamente, las computadoras no poseen dicha información, y consecuentemente no realizan una buena labor de desambiguación, quizás por el mal uso que posiblemente se hace de él, o quizás porque éste no es suficiente para solucionar dicho problema.

Se dice que una estructura gramatical (un texto, una oración, una palabra) es ambigua, cuando ésta puede ser entendida de dos o más formas, es decir que expresa más de un significado. Si la ambigüedad corresponde a la interpretación de una oración o un fragmento es llamada *estructural o sintáctica*, y si ésta se presenta en un vocablo es llamada *léxica*. Por ejemplo, la oración *the man saw the girl with the telescope*, presenta ambigüedad estructural, ya que podría ser interpretada de dos maneras: el hombre vio a una niña que tenía un telescopio o el hombre usó el telescopio para ver a una niña. Ahora, la oración *the man saw the girl with the red hat*, expresa un significado no ambiguo para un ser humano, ya que éste sabe que un sombrero no es utilizado para ver; sin embargo, sigue siendo ambigua para una computadora, porque ésta no sabe que un sombrero no se usa para ver.

La ambigüedad léxica, ocurre cuando un vocablo expresa múltiples sentidos, como por ejemplo *party*, el cual posee cinco sentidos diferentes si se toma como referencia el diccionario computacional WordNet 2.0, tal como se muestra en la figura 1.

An organization to gain political power.
An occasion on which people can assemble for social interaction and entertainment.
A band of people associated temporarily in some activity.
A group of people gathered together for pleasure.
A person involved in legal proceedings.

Figura 1. Definiciones de "party" según WordNet 2.0

Todos los sentidos de party podrían ser agrupados como un sentido genérico, tal como *group of people*; sin embargo algunas aplicaciones de recuperación de información o traducción automática tienen la necesidad de distinguir el sentido que expresa una palabra ambigua en un contexto. Por ejemplo, un sistema de traducción automática utiliza vocablos diferentes para definir cada sentido de un término ambiguo.

La ambigüedad léxica puede reflejarse como homonimia y polisemia. La homonimia se refiere a aquellas palabras que se escriben igual, pero que expresan diferentes significados, los cuales no guardan ninguna relación entre sí. Por ejemplo, *bank* expresa sentidos que hacen referencia a *river bank* y *financial institution*. La polisemia agrupa aquellas palabras que expresan significados relacionados como en el caso del vocablo *party*. Estas características dificultan la automatización de dicha tarea.

Ahora bien, si el número de términos ambiguos en una oración es superior a uno, las interpretaciones de la misma se incrementan dramáticamente, ya que una proposición puede expresar tantos significados como el producto de todos los posibles sentidos que expresan cada uno de los vocablos que forman parte de ésta. En el ejemplo anterior, *party* tiene 5 sentidos, *take* tiene 42, *vote* tiene 5, *last* tiene 10 y *election* tiene 4; por ende, existen 42,000 posibles interpretaciones para dicha proposición.

El método propuesto en esta tesis está enfocado a la resolución de la ambigüedad léxica para el idioma inglés. Se eligió este idioma debido a la existencia de recursos léxicos confiables, lo cual no sucede para el español. La expresión *recursos léxicos confiables* hace referencia a recursos y herramientas lingüísticas, tales como analizadores sintácticos, morfológicos, diccionarios electrónicos, ontologías, corpus etiquetados semántica o sintácticamente, corpus de texto, etc.

1.1 Hipótesis

Esta tesis se basa en la premisa: dos vocablos diferentes tienen significados similares si éstos ocurren en contextos parecidos. Basándose en dicha afirmación, es posible obtener términos similares a un vocablo ambiguo extrayendo las relaciones sintácticas existentes en su contexto local para luego compararlas con otras previamente almacenadas en una base de datos de recursos sintácticos. Los términos obtenidos permitirán seleccionar el sentido que expresa dicho vocablo en un contexto específico, aplicando un algoritmo de comparación de sentidos y ciertas medidas de similitud computadas sobre un espacio semántico.

1.2 Objetivo general

Implementar un método no supervisado basado en conocimiento, capaz de seleccionar el sentido que expresa un vocablo ambiguo, tarea conocida como resolución de sentidos de palabras (WSD por sus siglas en inglés), en un contexto dado explotando al máximo sus relaciones sintácticas. Dicho método debe ser capaz de integrar varios procesos y recursos lingüísticos, tales como analizadores sintácticos, morfológicos, medidas de similitud semántica, diccionarios computacionales, recursos sintácticos y bases de datos de valores que reflejen la similitud entre sentidos.

1.3 Objetivos específicos

- Definir los criterios lingüísticos que justifiquen el uso de ciertas relaciones de dependencia sintáctica, de tal manera que sea posible obtener la máxima cantidad de términos relacionados con el vocablo ambiguo, aprovechando al máximo el árbol sintáctico de la oración.

- Aplicar el modelo de espacio vectorial, tradicionalmente usado en sistemas de clasificación de textos, a las diferentes relaciones sintácticas extraídas de un corpus de texto, de tal manera que a cada una de éstas se le asocie un peso que refleje la correspondencia semántica que existe entre ambos miembros de la relación.

- Implementar un algoritmo que obtenga un conjunto de términos con cierta similitud semántica a un vocablo específico. Para ello, es necesario extraer las relaciones de dependencia existentes en el contexto sintáctico de dicho vocablo y luego compararlas con un conjunto de relaciones existentes en una base de datos, basándose en un esquema de vectores multidimensionales.

– Implementar un algoritmo de comparación basándose en el propuesto por McCarthy *et al*. [54], quien lo utilizó para obtener el sentido predominante de un término ambiguo. Sin embargo, el método propuesto será usado para seleccionar el sentido que expresa un vocablo en un contexto específico.

– Analizar algunas variables relevantes en el proceso de desambiguación de sentidos basado en el contexto local, tales como el número de vocablos similares necesarios para desambiguar una palabra exitosamente y el impacto de diferentes medidas de similitud y relación semántica en WSD.

1.4 Importancia y relevancia

La ambigüedad de sentidos de palabras es un problema que se presenta en cualquier lenguaje humano. Las palabras suelen expresar diferentes ideas dependiendo del contexto, la temática y el mundo pragmático en el que se presentan. La importancia de su resolución puede verse reflejada en muchas áreas y aplicaciones que dependen del procesamiento de lenguaje natural, tales como: traducción automática, recuperación de información, extracción de información, categorización de textos, etc. La desambiguación por sí misma no tiene sentido; sin embargo; su aplicación correcta en otras áreas permite obtener mejores resultados.

El hecho de que el método propuesto en este trabajo haya sido aplicado a la resolución de la ambigüedad en el idioma inglés, no implica que éste no pueda ser aplicado al español. La carencia de algunos recursos, tales como diccionarios computacionales organizados coherentemente para el español (WordNet para el inglés es más confiable que WordNet para el español), analizadores morfológicos y sintácticos de alto rendimiento y corpus textuales etiquetados con sentidos, son algunos de los problemas que tienen que ser resueltos para aplicar este método de manera confiable.

Finalmente, es necesario mencionar que este trabajo surgió como un proyecto piloto por parte del Laboratorio de Lenguaje Natural perteneciente al Centro de Investigación en Computación del Instituto Politécnico Nacional ubicado en la ciudad de México. En este trabajo nos concentramos sobre el inglés como lenguaje objeto del estudio; pero por la naturaleza de los métodos desarrollados, éstos pueden ser aplicados al idioma español. El Laboratorio de Lenguaje Natural cuenta con algunos recursos léxicos para el español, tales como un analizador morfológico, un analizador sintáctico y varios corpus de texto, los cuales podrían ser utilizados en la ejecución de este proyecto. La creación de un método de desambiguación de sentidos de palabras para el español beneficiaría el desarrollo de muchas aplicaciones, por ejemplo la traducción automática entre el español y los diferentes

idiomas hablados en México, la recuperación de información clasificada sobre los diferentes fenómenos naturales que se presentan en las costas de México, entre otros. Cabe resaltar que para maximizar el rendimiento de dichas tareas, definitivamente es necesario contar con un método confiable que sea capaz de desambiguar sentidos exitosamente.

1.5 Novedad científica

McCarthy *et al*. [54] propusieron un método en el cual cada término ambiguo es etiquetado semánticamente con su sentido más predominante sin importar su contexto. Para ello, aplicaron un algoritmo de comparación entre dicho término y un conjunto de vocablos similares a éste, los que a su vez son proporcionados por el tesauro de Lin [51]. Por ejemplo, el sentido más preponderante de *bank* hace referencia a *institución financiera*; de tal manera que si éste expresase el sentido *banco de peces* en un texto, el método planteado siempre le asignará como etiqueta semántica el sentido *institución financiera*. Este método ha logrado mejores resultados que cualquiera de los algoritmos no supervisados existentes, los cuales se basan en la información que aporta el contexto del vocablo ambiguo y no en su sentido predominante.

En esta tesis se aplica el algoritmo de etiquetado de sentidos propuesto por McCarthy *et al*. , para la obtención del sentido que expresa un vocablo ambiguo en un contexto específico. Los términos similares necesarios para este proceso son proporcionados por una base de datos de relaciones sintácticas, contra las que se comparan las relaciones de dependencia existentes en el contexto local del vocablo ambiguo. El método propuesto ha logrado mejor precisión que el mejor método no supervisado conocido hasta el momento.

Otra de las novedades científicas presentadas en este trabajo, es el uso de un contexto sintáctico más amplio que el comúnmente empleado en los trabajos de WSD. Dicho contexto está conformado por el conjunto de flechas salientes del vocablo (*camisa* $\rightarrow$ *roja*); y también por la flecha entrante a éste. Asimismo, las ramas del tipo *palabra_1* $\rightarrow$ *preposición* $\rightarrow$ *palabra_2,* son consideradas como una relación de dependencia entre *palabra_1* y *palabra_2*.

1.6 Aportaciones

La desambiguación de sentidos de palabras es un proceso que requiere de muchos recursos y herramientas lingüísticas, tales como corpus de texto, diccionarios computacionales, corpus etiquetados semánticamente, analizadores morfológicos y

sintácticos. También, requiere de otros procesos más específicos como la implementación de medidas de similitud y relación semántica, sistemas de recuperación de información, entre otros. Finalmente, las aportaciones de este trabajo, las cuales se listan a continuación, caen en dichas categorías.

- Método para desambiguar sentidos de palabras.
- Analizador sintáctico de dependencias basado en MINIPAR.
- Programa para la extracción de tripletas de dependencia sintáctica.
- Sistema de recuperación de palabras similares.
- Programa para el etiquetado semántico de un vocablo.
- Base de datos de valores de similitud.
- Base de datos de vocablos relacionados sintácticamente.

A continuación, se explica brevemente la funcionalidad y los beneficios que ofrece cada una de estas aportaciones.

1.6.1 Método para desambiguar sentidos de palabras

El principal aporte de esta tesis es el método propuesto para la desambiguación de sentidos. Éste es un método no supervisado basado en conocimiento, el cual ha tomado como referencia algunas técnicas y trabajos previos, siendo los más relevantes el algoritmo planteado por McCarthy *et al*. [54], usado en este trabajo para obtener el sentido que expresa un término ambiguo en un contexto dado, el modelo de espacio vectorial o el esquema TF-IDF (por sus siglas en inglés *term frecuency-inverse document frecuency*), usado para organizar el recurso sintáctico, el uso de analizadores morfológicos y sintácticos para obtener categorías gramaticales y árboles de dependencia sintáctica y la aplicación de medidas de similitud y relación semántica sobre WordNet.

El método planteado integra cada una de estas herramientas como módulos individuales que interactúan unos con otros dependiendo de las necesidades de la aplicación. La funcionalidad que provee cada módulo puede ser reutilizada por cualquier otra aplicación concerniente al procesamiento de lenguaje natural.

1.6.2 Analizador sintáctico de dependencias basado en MINIPAR

Una de las necesidades del método de desambiguación presentado, es obtener las relaciones sintácticas del contexto local de una palabra. Éstas serán utilizadas para crear una base de datos de pares de vocablos unidos bajo cierta relación de dependencia y además, permitirán seleccionar aquellos términos que presenten alguna similitud semántica

con el vocablo ambiguo, ya que la obtención de términos similares se basa en la comparación de contextos sintácticos.

MINIPAR [50] es un conjunto de librerías implementadas por Dekang Lin, las cuales han sido desarrolladas en lenguaje C. Éstas proporcionan la funcionalidad necesaria para analizar sintáctica y morfológicamente los vocablos de una oración. De esta manera, el analizador sintáctico implementado utiliza dichas librerías para generar árboles de dependencia sintáctica para todas las oraciones de un corpus de texto.

1.6.3 Programa para la extracción de tripletas de dependencia

Una tripleta de dependencia sintáctica es una estructura gramatical que se encuentra conformada por un par de vocablos que interactúan bajo cierta relación de dependencia. Las relaciones *convencionales* utilizadas por la lingüística computacional no identifican plenamente el contexto local de un vocablo; es por ello, que se han creado relaciones de dependencia *sin preposición*, y *especial*. En la primera, la preposición es excluida cuando se presenta como modificador de algún vocablo, de tal manera que el término dependiente de una preposición, pasa a ser el modificador de dicho vocablo. En las relaciones de dependencia *especial* se incluye como parte del conjunto de los modificadores de un vocablo, el término al que modifica dicho vocablo. Éstas serán explicadas ampliamente en el capítulo 4, específicamente en el módulo encargado de obtener tripletas normalizadas.

La extracción de tripletas de dependencia del árbol sintáctico, es una tarea obligatoria para poder crear una base de datos de términos relacionados y obtener las características sintácticas del contexto local de un término ambiguo. El programa implementado permite extraer relaciones de dependencia *convencional*, *sin preposición*, y *especial*, para lo cual utiliza como fuente de información el árbol proporcionado por el analizador sintáctico.

1.6.4 Programa para la recuperación de palabras similares

El método de desambiguación propuesto asigna un sentido a un vocablo basándose en un conjunto de términos que tengan cierta similitud semántica a éste. Una de las maneras de obtener tales términos, es comparar el contexto del vocablo a desambiguar con el contexto de otros. De esta manera, al obtener los contextos más parecidos se obtendrán los términos más similares.

El programa que implementa la recuperación de palabra similares se basa en el modelo de espacio vectorial o el esquema TF-IDF (por sus siglas en inglés *term frecuency-inverse document frecuency*), el cual generalmente es aplicado a tareas de clasificación y

similitud de documentos. Cada contexto sintáctico es representado por un vector multidimensional, el cual es comparado con más de 50,000 vectores previamente compilados en una base de datos.

Finalmente, el programa proporciona un conjunto de términos, cada uno con un peso que refleja la similitud semántica con respecto al vocablo ambiguo.

1.6.5 Programa para el etiquetado semántico de un vocablo

Este programa permite elegir el sentido de un vocablo tomando en cuenta sus términos similares; de tal manera que se computa la proximidad semántica existente entre los sentidos del vocablo ambiguo y cada una de las glosas de los términos similares. Para ello, se usan métricas de similitud y WordNet como espacio semántico.

El programa de etiquetado semántico se basa en el algoritmo propuesto por McCarthy *et al*. [54], con la diferencia que los términos similares utilizados son calculados dinámicamente utilizando una base de datos de recursos sintácticos. Asimismo, las medidas de similitud implementadas son las propuestas por Jiang–Conrath [34], Lin [51], y la métrica de relación propuesta por Banerjee–Pedersen, más conocida como medida de Lesk adaptada [6].

1.6.6 Base de datos de valores de similitud

Como ya se mencionó, WordNet es el diccionario computacional usado por las métricas de similitud para computar un valor de proximidad semántica. Son dos las mediciones requeridas por el algoritmo de etiquetado semántico:

- Medida que cuantifique la proximidad semántica entre dos sentidos basándose en alguna métrica de similitud.
- Medida que permita elegir el sentido de un vocablo que más se asemeje a otro sentido.

El costo computacional que implica calcular dicha información es alto. Esto se debe a que WordNet está organizado como un conjunto de archivos planos y cada vez que es necesario realizar dichos cálculos, es posible que se tenga que recorrer ciertas jerarquías de información dependiendo del tipo de métrica que se utiliza. Por ende, el tiempo de cómputo y acceso a su información es elevado.

Ésta es la razón principal por la que se ha creado una base de datos con más de cien mil valores que hacen referencia a dicha información y que usan tres medidas de similitud,

específicamente las propuestas por Jiang–Conrath [34], Lin [51] y Lesk [46]. Además de acelerar el proceso de desambiguación, este recurso puede ser utilizado por cualquier aplicación que procese lenguaje natural. La tabla 1 muestra el peso de similitud entre dos sentidos tomando en cuenta la medida de Jiang–Conrath, tal como se encuentra almacendo en la base de datos. Cada sentido usa el formato *vocablo#categoría gramatical#número de sentido*, tal como se muestra en la tabla 1.

Tabla 1. Ejemplo de valores almacenados en la base de datos de sentidos

Sentido 1	**Sentido 2**	**Similitud según medida JCN**
Atlanta#n#1	*today#n#1*	0.525
Atlanta#n#1	*today#n#2*	0.048
city#n#1	*cast#n#1*	0.061
city#n#1	*cast#n#2*	0.061
appointment#n#1	*indefinity#n#1*	0.043
appointment#n#2	*indefinity#n#1*	0.048
administration#n#1	*law_of_nature#n#1*	0.057
administration#n#2	*law_of_nature#n#1*	0.061
couple#n#1	*marriage_bed#n#1*	0.050
couple#n#2	*marriage_bed#n#1*	0.049
candidate#n#1	*mathematician#n#1*	0.063
candidate#n#2	*mathematician#n#1*	0.059
audience#n#1	*oracle#n#3*	0.000
audience#n#2	*oracle#n#1*	0.046
audience#n#2	*oracle#n#2*	0.000
assistant#n#1	*Napoleon_Bonaparte#n#1*	0.110
assistant#n#1	*Norma#n#1*	0.000
assistant#n#1	*opera_bouffe#n#1*	0.052
assistant#n#1	*Paper_electrophoresis#n#1*	0.050
construction#n#1	*law_degree#n#1*	0.053
construction#n#2	*law_degree#n#1*	0.046
construction#n#3	*law_degree#n#1*	0.060
construction#n#4	*law_degree#n#1*	0.063
construction#n#5	*law_degree#n#1*	0.042
construction#n#6	*law_degree#n#1*	0.056
construction#n#7	*law_degree#n#1*	0.041

1.6.7 Base de datos de vocablos relacionados sintácticamente

Para obtener los términos similares a un vocablo ambiguo, es necesario comparar el contexto sintáctico de dicho término con otros previamente recopilados, los cuales han sido

obtenidos del corpus SemCor y están conformados por tripletas de dependencia sintáctica. Dependiendo del tipo de relación de dependencia usada, se han creado tres bases de datos:

- Base de datos de relaciones de dependencia sintáctica *convencional*.
- Base de datos de relaciones de dependencia sintáctica *sin preposición* y *convencional*.
- Base de datos de relaciones de dependencia sintáctica *convencional*, *especial* y *sin preposición*.

Cada una de éstas está conformada por alrededor de medio millón de pares de vocablos bajo cierta relación de dependencia. Además, cada tripleta cuenta con un valor que especifica el grado de relación semántica existente entre ambos términos; así como la información sintáctica correspondiente también conocida como POS (*part of speech*). La tabla 2 presenta varias tripletas almacenadas en este recurso; véase la sección 4.1.5 para la definición del peso de relación.

Tabla 2. Relaciones de dependencia sintáctica almacenadas en la base de datos

Vocablo 1	**POS**	**Vocablo 2**	**Peso de relación**
accessory	N	*ceramic*	3.913
accessory	N	*hand-crafted*	4.758
accessory	N	*lady*	3.115
accident	N	*fewer*	1.718
accident	N	*war*	0.923
accident	N	*grievance*	2.228
accident	N	*murder*	1.567
accident	N	*automobile*	3.227
bastard	N	*unbroken*	0.442
bastard	N	*mean*	0.366
bastard	N	*reactionary*	0.856
bastard	N	*crummy*	1.057
bastard	N	*stupid*	0.413
bastard	N	*poor*	0.670
zone	N	*fact*	0.677
zone	N	*strike*	0.813
zone	N	*sector*	0.903
zone	N	*sleeping*	0.939
worried	V	*friend*	1.052
worried	V	*health*	1.052
worried	V	*happening*	1.226
world	N	*impious*	0.237
world	N	*possible*	0.244
world	N	*outside*	0.247

1.7 Publicaciones generadas

Hasta el momento se generaron las siguientes publicaciones; otras están en proceso de preparación.

- J. Tejada-Cárcamo, A. Gelbukh, H. Calvo. Desambiguación de sentidos de palabras usando relaciones sintácticas como contexto local. Tutorials and Workshops of Fourth Mexican International Conference on Artificial Intelligence, ISBN 968-891-094-5, 2005.

- J. Tejada-Cárcamo. El impacto de relaciones sintácticas y similitud semántica en la desambiguación de sentidos de palabras (preparado).

Capítulo 2
Procesamiento automático de texto y resolución de la ambigüedad

A lo largo de este capítulo se presentan cuatro tópicos. Primero, se describe brevemente el concepto de lingüística computacional, así como sus aplicaciones y productos, proporcionando un breve panorama sobre esta área. En el segundo punto, se presenta el rol de la ambigüedad en la lingüística computacional, los diversos tipos de ésta y las aplicaciones que la requieren para incrementar su rendimiento. En el tercer punto, se detallan los métodos existentes para la resolución de este problema (métodos basados en conocimiento o basados en corpus) y la importancia del contexto (micro-contexto y macro-contexto) en la obtención del sentido correcto de un vocablo ambiguo. Finalmente, en la última sección, se definen algunas medidas de similitud y relación semántica utilizadas para comparar la proximidad o lejanía semántica entre dos conceptos, tomando como referencia un espacio semántico.

2.1 Lingüística computacional

La lingüística computacional puede considerarse una disciplina de la lingüística aplicada y la inteligencia artificial. Tiene como objetivo la creación e implementación de programas computacionales que permitan la comunicación entre el hombre y la computadora, ya sea mediante texto o voz [25]. La lingüística computacional, también es llamada procesamiento del lenguaje natural (PLN). Algunos ejemplos de aplicaciones de PLN son los programas reconocedores de habla, traductores automáticos, correctores ortográficos, etc.

Pese a los avances tecnológicos, poco es lo que se sabe acerca de cómo el ser humano procesa el lenguaje natural. Los lingüistas llevan décadas intentando descifrar cómo funciona esta capacidad única de la especie humana. Muchos animales tienen formas complejas de comunicación; sin embargo, ninguno de estos pseudo-lenguajes cumple la característica más significativa del lenguaje humano: la *infinitud discreta*.

La *infinitud discreta* quiere decir que el lenguaje humano es discreto en cuanto a sus unidades, pero infinito en cuanto a las combinaciones que pueden hacerse con éstas. Por ejemplo, las palabras son unidades discretas y finitas de la lengua; sin embargo, combinando un número limitado de palabras es posible construir frases infinitas. Ésta es la razón principal por la que un niño o un adulto construyen continuamente frases que no han escuchado jamás, tomando como base aquellos vocablos escuchados, memorizados y

comprendidos con anterioridad. Basándose en esta característica, es posible afirmar que hablar es inventar constantemente nuevas combinaciones de palabras.

2.2 Aplicaciones de la lingüística computacional

En esta sección se presentan algunas aplicaciones o productos que son desarrollados actualmente por esta disciplina, específicamente aquellas que tienen que ver con el procesamiento y generación de texto, diálogo con computadoras y comprensión del lenguaje. Algunas de estas tareas han sido implementadas exitosamente para el inglés, mientras que para otros lenguajes, aún se encuentran en plena investigación o implementación.

2.2.1 División automática de palabras

La división automáticade palabras, que en inglés se denomina *automatic hyphenation*, consiste en dividir automáticamente una palabra cuando ésta llega al final de un renglón, de tal manera que una parte es movida al siguiente renglón. La división del vocablo debe realizarse en posiciones específicas del mismo, las cuales generalmente coinciden con límites silábicos. Por ejemplo, las siguientes divisiones de palabras son correctas para el idioma español: *re-ci-bo*, *re-u-nir-se*, *dia-blo*, *ca-rre-te-ra*, *mu-cha-chas*; y sería incorrecto dividirlas en otras posiciones, tales como: *recib-o*, *di-ablo*, *muc-hac-has*.

La división automáticade palabras, mejora la apariencia de los textos elaborados por computadora, ajustando sus márgenes derechos, ahorrando papel y al mismo tiempo mejorando la legibilidad de un texto en comparación con aquellos que no fueron construidos usando esta característica. La información lingüística que utilizan los programas computacionales, a veces suele ser muy limitada, ya que sólo consideran las vocales o consonantes y las combinaciones inseparables de letras (como parejas de consonantes *ll, rr, ch* o diptongos *io, ue, ai* en español). Sin embargo, para lograr una mejor calidad, se requiere información más detallada sobre cada vocablo.

La división automáticade palabras depende directamente de la estructura morfológica de cada una, por ejemplo: *sub-ur-ba-no* y *su-bir* para el español. Los programas que se basan en diccionarios pueden tomar en cuenta todas estas consideraciones.

2.2.2 Corrección tipográfica y ortográfica

Esta aplicación, que en inglés se denomina *spell checking*, consiste en detectar y corregir errores ortográficos y tipográficos en el texto a nivel palabra cuando ésta es considerada fuera de su contexto. Los errores tipográficos se presentan cuando un usuario presiona una tecla que no corresponde a lo que realmente quería digitar, y los errores ortográficos se presentan cotidianamente en la ortografía de las palabras, y más aún si el lenguaje que se está usando no es el nativo.

La cantidad de información lingüística necesaria para construir un corrector ortográfico, es mayor que la usada para la división automática de palabras. Cualquier corrector empleará técnicas basadas en listas o diccionarios que contengan todos los vocablos válidos para un lenguaje específico. Asimismo, es necesario contar con ciertos criterios sobre similitud semántica, algunas presuposiciones acerca de los errores tipográficos y ortográficos más frecuentes y conocimientos detallados de morfología para poder crear un diccionario compacto y eficiente.

2.2.3 Corrección de estilo

Los errores de estilo son aquellos que hacen uso incorrecto de las palabras o sus combinaciones, ya sea desde una perspectiva general o específica a un género literario. Esta aplicación puede ser análoga a los manuales hechos para personas, los cuales hablan sobre normas y estilos gramaticales; por ende, los correctores de estilo juegan un rol didáctico y prescriptivo en la redacción de textos. Por ejemplo, no es recomendable usar palabras vulgares en construcciones coloquiales cuando se escriben documentos oficiales. Algunas propiedades formales para textos en español mencionan que los párrafos serios, normalmente no deben tener más de diez veces la preposición *de*, ni tampoco exceder veinte renglones de escritura. Asimismo, el uso de anglicismos está penalizado en la redacción de este tipo de textos.

Los correctores de estilos deben de usar un diccionario de palabras con sinónimos, información acerca del uso de las preposiciones, compatibilidad con otros vocablos, marcas de uso dependiendo del género literario, etc. También deben de usar analizadores sintácticos para detectar construcciones sintácticas incorrectas. Algunos correctores de estilo comerciales, calculan la longitud promedio de los vocablos, el número de letras de cada vocablo, la longitud de las oraciones, la longitud de los párrafos y otras estadísticas que resultan de combinar las anteriores, para de esta manera solucionar alguno de los problemas referentes a estilo.

La referencia a un vocablo o a una combinación de vocablos, tiene por objetivo ayudar al autor a redactar su texto, proporcionándole mayor flexibilidad, corrección, vocabulario y expresiones idiomáticas. Sorpresivamente, sólo una insignificante cantidad de todas las posibles combinaciones de palabras, son realmente permitidas en un lenguaje. El conocimiento de estas posibles combinaciones podría llegar a ser un recurso importante para el autor. Por ejemplo, a un extranjero de habla inglesa, le serviría mucho conocer los verbos que comúnmente se usan con el sustantivo *ayuda*, tales como *prestar* o *pedir*, o con el sustantivo *atención*, tales como *dedicar* o *prestar*, para evitar la creación de combinaciones incorrectas, como la estructura sintáctica *pagar atención*, la cual sería una traducción vocablo a vocablo de la frase inglesa *pay attention*.

2.2.4 Corrección de errores gramaticales

La detección y corrección de errores gramaticales es una tarea más delicada que la corrección de errores ortográficos; ya que ésta no sólo debe tomar en cuenta los vocablos adyacentes en una oración, sino debe de considerar toda la oración. Los errores gramaticales, por lo general, son aquellos que violan las reglas sintácticas o las relacionadas con la estructura de la oración, por ejemplo en el idioma español es necesario que exista una correspondencia entre el género y el número de un sustantivo y un adjetivo. Otro error gramatical es el incorrecto uso de preposiciones, por ejemplo *casarse a María* o *debajo la puerta*. Existen ciertos errores gramaticales que ni siquiera son evidentes para los hablantes nativos de un idioma.

Estos errores podrían solucionarse con un análisis sintáctico completo del texto en mención; pero debido a la falta de herramientas sintácticas totalmente confiables, los correctores gramaticales, si bien han logrado avances significativos, aún no han conseguido cubrir la totalidad de errores gramaticales que pueden presentarse en un idioma.

Algunos correctores comerciales se basan en técnicas simplistas para la detección de errores gramaticales. Las consecuencias de usar dichos correctores simplistas son las posibles falsas alarmas que suelen mostrar al usuario. Por ejemplo, en la proposición *las pruebas de evaluación numerosas*, la correspondencia gramatical entre los vocablos *evaluación* y *numerosas* es correcta; sin embargo, un programa simplista mencionaría la existencia de un posible error en cuanto a la correspondencia en *número* entre ambos vocablos. Finalmente, el autor del texto es quién tiene la última palabra para aceptar la corrección o rechazarla.

2.2.5 Recuperación de información

La recuperación de información, que en inglés se denomina *information retrieval* (IR), es la ciencia encargada de buscar información en archivos de diversos tipos, en meta-datos y en bases de datos textuales, de imágenes o de sonidos. La plataforma sobre la cual es posible realizar dichas búsquedas se extiende desde computadoras de escritorio, redes de computadoras privadas o públicas hasta intranets e internet.

La recuperación de información comprende un estudio interdisciplinario. Esto genera normalmente un conocimiento parcial desde diferentes perspectivas, lo cual se convierte en una desventaja cuando no se cuenta con un grupo humano capaz de satisfacer dichos requerimientos. Algunas de las disciplinas utilizadas por esta área son la psicología cognitiva, la arquitectura y diseño de la información, la lingüística, la semiótica, el comportamiento humano orientado a la información, la informática y la biblioteconomía. Los buscadores, tales como Google, Altavista, Lycos son algunas de las aplicaciones más populares de esta área.

2.2.6 Traducción automática

La traducción automática es una actividad tan antigua como la informática. Esta actividad consiste en convertir el texto de un idioma a otro automáticamente usando un programa computacional. Se trata de una disciplina que ha contribuido de manera determinante al desarrollo de la lingüística computacional. Es una de las aplicaciones informáticas que mayores recursos humanos y económicos ha recibido. El mercado ofrece en la actualidad un amplio abanico de productos y es difícil para el usuario elegir el más adecuado a sus necesidades. Es importante saber que un texto producido por un sistema de traducción automática debe ser revisado con cuidado antes de darlo por válido y publicarlo. Sin embargo, existen casos en los que no es necesario obtener resultados de calidad y basta con una aproximación al contenido esencial del texto, si lo que se desea es detectar información relevante o crítica.

Desde el punto de vista de diseño de sistemas de traducción automática, existen tres enfoques diferentes de esta disciplina: los sistemas directos, los de transferencia y los sistemas basados en interlingua. Cada uno de estos, permite que las traducciones se desarrollen de una u otra forma, variando el resultado obtenido.

Los sistemas de traducción directa podrían compararse con grandes diccionarios, los cuales traducen un texto usando una técnica muy simplista; es decir, vocablo a vocablo. La información sintáctica que poseen estos sistemas es mínima; por ende, los resultados que ofrecen suelen ser bastante pobres.

Los sistemas de transferencia contienen además de grandes léxicos bilingües y multilingües, un amplio conocimiento sintáctico y semántico de las lenguas tratadas. Esto permite traducir vocablos de una lengua a otra teniendo en cuenta el contexto morfológico, sintáctico y semántico. Estos sistemas son capaces de lograr cierta transferencia estructural automática; es decir, realizan cambios en el orden de constituyentes y en la estructura de la frase adecuándola a la lengua destino.

En los sistemas interlingua se lleva a cabo un análisis mucho más profundo del texto. La idea de este enfoque consiste en crear un lenguaje artificial conocido como *interlingua*, el cual comparte y distingue una considerable cantidad de características comunes a varios idiomas. De esta manera, para traducir un texto de un idioma a otro, se necesita un analizador que convierta el texto del idioma fuente al interlingua, y un generador que convierta el interlingua al texto del idioma destino.

Hay que tener en cuenta que no existen sistemas puros de traducción directa, de transferencia o interlingua, sino sistemas que se aproximan más a un enfoque determinado. Asimismo, es posible que un sistema pueda utilizar características específicas de cada enfoque.

2.2.7 Generación del habla

La generación del habla, también llamada generación de voz, es la disciplina encargada de producir discurso humano sin utilizar directamente la voz humana. En general, se denomina sintetizador del habla a cualquier tipo de sistema (cualquier aplicación informática implementada sobre una computadora o algún tipo de dispositivo electrónico) capaz de producir habla artificialmente.

En particular, los sistemas que son capaces de generar habla sintética usando como fuente de información grafías convencionales agrupadas en un texto, se denominan sistemas generadores de voz, que en inglés se conocen como *text-to-speech* (TTS). Es necesario tener en cuenta que las representaciones gráficas usadas por los lenguajes naturales no permiten establecer una correspondencia biunívoca entre los grafemas y los sonidos que representan. Tomando en cuenta esta característica, existen idiomas más regulares que otros.

Los sistemas generadores de voz se componen de dos fases. En la primera, se toma una entrada en forma de texto y se genera una salida usando alguna representación lingüística simbólica. En la segunda, se toma como entrada dicha representación lingüística y se obtiene como resultado el habla sintética. En cuanto a las cualidades de la voz sintética generada, se habla de su naturalidad (similitud entre la salida sonora y la voz de un

auténtico humano) y de su inteligibilidad (claridad y facilidad para entender la salida del sistema).

2.2.8 Resumen de documentos

En muchos casos es necesario determinar automáticamente los tópicos de un escrito, de tal manera que dicha información sea utilizada para clasificar documentos del mismo tema. Los tópicos obtenidos también pueden ser usados por sistemas de recuperación de información o simplemente para orientar a los seres humanos sobre la temática de un documento antes de que empiecen a leerlo.

Existen varias ideas a tomar en cuenta para lograr este objetivo, tales como neutralizar las variantes morfológicas para reducir las palabras encontradas en un texto, por ejemplo oraciones $\rightarrow$ oración, regímenes $\rightarrow$ régimen y a la vez usar un diccionario electrónico o tesauro para poder clasificar un vocablo dentro de un tópico más especializado, por ejemplo el vocablo *oración* pertenecería al tópico *lingüística*, el cual a su vez pertenece a *ciencias sociales* que se encuentra inmerso dentro del tópico global *ciencias*.

Hay otras técnicas que se basan en una lista de *palabras de parada*, que en inglés se denomina *stop-words*, en las que se incluyen preposiciones, artículos y aquellos términos que no sean considerados como relevantes en un lenguaje, para que de esta manera se puedan seleccionar sólo las vocablos importantes de un texto, y después proceder a encontrar los vocablos relacionados a éstos (dos términos son considerados relacionados si la distancia entre ambos es muy corta en la misma oración). Finalmente, dichos vocablos son usados para comprimir el texto de un documento, eliminando oraciones y párrafos que contienen poca o ninguna coincidencia con este conjunto de términos.

2.3 La ambigüedad y la lingüística computacional

La ambigüedad surge en el lenguaje natural cuando una estructura gramatical puee ser interpretada de varias maneras. La desambiguación de sentidos de palabras, que en inglés se denomina *word sense desambiguation* (WSD), consiste en identificar el sentido de un vocablo ambiguo en un determinado contexto usando un conjunto de candidatos establecidos, por ejemplo, *bat* podría ser un pequeño animal nocturno o una pieza de madera para hacer deporte y *bank* puede hacer referencia a un banco de peces o a una institución financiera. Cuando se usa un diccionario para encontrar la definición de una palabra, es posible verificar los diversos sentidos que ésta presenta, los cuales pueden ser

totalmente diferentes, por ejemplo los sentidos del vocablo *consult* son *pedir un consejo* y *dar un consejo*.

2.3.1 Aplicaciones que requieren resolver la ambigüedad

La desambiguación no es un fin en sí misma, sino un proceso intermedio muy necesario para algunas tareas del procesamiento del lenguaje natural, tales como traducción automática, recuperación de información, extracción de información, categorización de textos, etc.

La traducción automática requiere al menos dos etapas: entender el significado de lo que se desea traducir, y una vez comprendido, generar las oraciones correctas al idioma destino. WSD es requerido en ambas etapas, ya que un vocablo puede tener más de una posible traducción en el lenguaje destino. Por ejemplo, la palabra inglesa *drug* puede ser traducida al turco como *ilac*, término que hace referencia al sentido de medicina o como *uyusturucu* para su correspondiente sentido de *dope* dependiendo del contexto donde ésta haya aparecido.

El área de recuperación de información también se beneficia de WSD, ya que la existencia de vocablos ambiguos en las consultas, es uno de los problemas principales en los sistemas de recuperación de información. Dichos sistemas necesitan módulos de WSD para no mostrar aquellos documentos cuyos sentidos no son relevantes para la consulta.

Para los sistemas de procesamiento de habla, es importante determinar la correcta pronunciación de las palabras para generar sonidos naturales. Este proceso es muy difícil, ya que existen vocablos que toman una entonación diferente teniendo en cuenta el sentido que desean expresar. En el artículo escrito por Stevenson [75] se menciona un ejemplo al respecto, en el cual se afirma que la palabra *lead* es pronunciada de una manera distinta cuando se refiere al sentido *be in fron*t que cuando se refiere al sentido *a type of metal*. WSD podría ayudar a identificar el correcto sentido de una palabra en el texto para generar una pronunciación correcta. El problema inverso podría ocurrir en el reconocimiento de palabras homófonas; es decir, vocablos que son diferentes textualmente; pero se pronuncian de la misma manera.

Para analizar el contenido y temática de un texto, es necesario considerar la distribución de las categorías predefinidas de las palabras, ya que existen vocablos que por su propia naturaleza expresan algún concepto, idea o tema específico. WSD es útil cuando se desea encontrar vocablos similares al sentido específico de una palabra [42].

2.3.2 Tipos de ambigüedad

Tradicionalmente, se distinguen tres tipos de ambigüedad: ambigüedad léxica, semántica y estructural o sintáctica [30]. La ambigüedad léxica se encarga de procesar aquellos vocablos que pueden pertenecer a diferentes categorías gramaticales, por ejemplo *para* puede desempeñarse como preposición o como alguna conjugación del verbo *parar* e incluso del verbo *parir*.

La ambigüedad semántica procesa aquellos vocablos que tienen múltiples significados, por ejemplo *banco* puede significar *banco de peces*, *banco para tomar asiento* o *institución financiera*. Asimismo, es posible que una misma estructura sintáctica exprese diferentes significados, como la oración *todos los estudiantes de la escuela hablan dos lenguas*, la cual podría significar que cada estudiante habla dos lenguas o que en la escuela sólo se hablan dos lenguas determinadas. Otro ejemplo de este tipo de ambigüedad se presenta en la oración *vi a tu hermano volando hacia París*, la cual no deja claro si la persona que vio al hermano y el hermano iban en el mismo avión o en caso contrario el hermano vuela.

La ambigüedad sintáctica, también conocida como ambigüedad estructural procesa aquellas oraciones que pueden tener más de una estructura sintáctica. Por ejemplo, en la oración *Juan vio a un hombre con un telescopio*, es posible extraer al menos dos árboles de constituyentes, tal como se muestra en la figura 2.

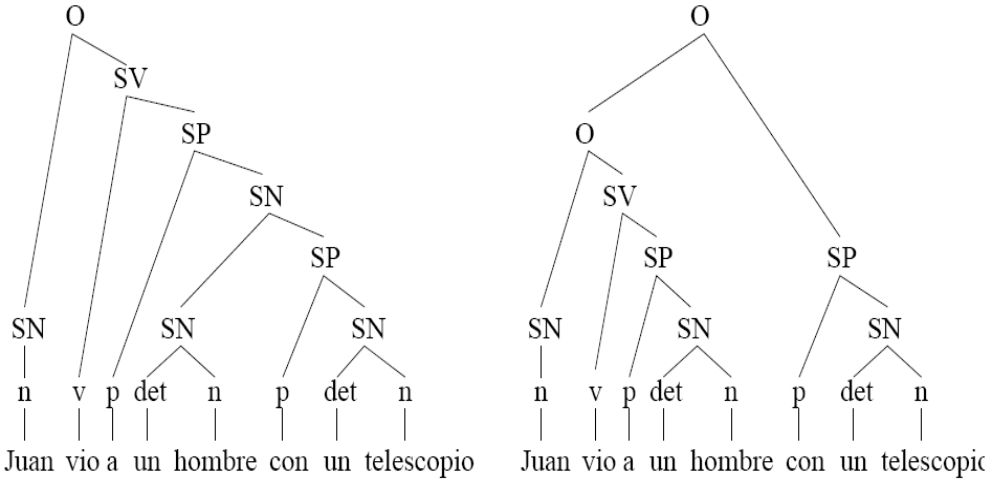


Figura 2. Ejemplo de ambigüedad sintáctica en una oración

2.3.3 Ambigüedad de sentidos de palabras

En la práctica resulta muy difícil distinguir el tipo de ambigüedad [37], debido a la estrecha relación de dependencia entre los niveles clásicos del análisis lingüístico (morfológico, sintáctico, semántico y pragmático). Por ejemplo, el vocablo *traje* posee ambigüedad léxica, ya que puede ser verbo o sustantivo, y también ambigüedad semántica, ya que expresa diferentes ideas en cada caso.

Los trabajos realizados en esta tesis están orientados a solucionar el problema de la ambigüedad de sentidos de palabras o ambigüedad semántica a nivel de palabra. Esto significa etiquetar un vocablo ambiguo con el sentido que expresa en un contexto determinado. Es necesario tener en cuenta que el significado o idea específica de *sentido de palabra*, aún no se encuentra claramente definido por la comunidad científica, lo cual ha hecho que se desarrollen diferentes trabajos en WSD, cada uno tomando su propia versión acerca de dicha definición. Lo que si parece claro, es la posibilidad de distinguir entre desambiguación morfo-sintáctica y desambiguación de sentidos de palabras [37].

El método propuesto intenta etiquetar semánticamente un vocablo ambiguo con el sentido (tomando como referencia WordNet 2.0) que expresa en un contexto específico. Para ello, toma en cuenta las relaciones de dependencia existentes en su micro-contexto y la información proporcionada por diversos recursos externos.

2.4 Resolución de ambigüedad de sentidos de palabras

Todos los trabajos de desambiguación de sentidos de palabras asocian el contexto (ya sea micro o macro-contexto) de un vocablo ambiguo con la información de un recurso de conocimiento externo (métodos basados en conocimiento), o con el contexto de instancias del mismo vocablo previamente desambiguado, las cuales son obtenidas de un corpus en una fase previa de entrenamiento (métodos basados en corpus). Cualquiera de estos métodos puede ser utilizado para asignar un sentido a cada ocurrencia de un vocablo ambiguo. La siguiente sección describe los recursos y técnicas empleadas por ambos métodos.

2.4.1 Métodos basados en conocimiento

En la década de los 70, diversas técnicas de inteligencia artificial eran usadas para desambiguar sentidos de palabras; sin embargo, esos trabajos se vieron limitados por la

falta de recursos de información, surgiendo un gran problema, denominado *cuello de botella para la adquisición de conocimiento* [24].

Fue en los 80, cuando surgió una diversidad de materiales léxicos de gran escala, como diccionarios electrónicos, tesauros y corpus, los cuales dieron inicio a la extracción de información automática. Fue en esta época, que disminuyó el uso de teorías lingüísticas, las cuales fueron sustituidas por heurísticas para solucionar el problema de WSD. A continuación se explican los recursos más importantes que actualmente se usan por los métodos basados en conocimiento.

a. Uso de diccionarios electrónicos

Un diccionario electrónico surge de convertir un diccionario normal, creado exclusivamente para el uso humano, a formato electrónico. Estos diccionarios proveen información sobre sentidos de vocablos ambiguos, lo cual es explotado por el área de WSD.

El primer investigador que usó dichos diccionarios fue Lesk [46], quien creó una base de conocimiento, asociando a cada sentido una lista de palabras obtenidas de su definición. El proceso de desambiguación compara el contexto del vocablo ambiguo con la lista de cada uno de sus sentidos. Otra variante, es comparar las listas de los vocablos del contexto con las del término ambiguo. Lesk logró una eficiencia de 50-70%; aunque el problema con este método es su *sensibilidad*; es decir, que la presencia o ausencia de algún vocablo contextual afecta radicalmente los resultados. Sin embargo, ésta técnica ha servido como base para muchos sistemas de WSD que usan diccionarios electrónicos como recurso de información.

Wilks [81] intentó mejorar el conocimiento asociado a la definición de cada sentido, calculando la frecuencia de co-ocurrencia entre los vocablos miembros de cada lista usando para ello un corpus de información. De esta manera, obtuvo diferentes medidas en cuanto al grado de relación semántica entre ellos. Finalmente, usó un método basado en vectores, el cual relaciona cada vocablo con su contexto y la métrica obtenida. Al experimentar con un solo término ambiguo, específicamente el vocablo *bank*, logró una eficiencia del 45% en etiquetado de sentidos y 90% en la detección de homógrafos.

Debido a que estos diccionarios fueron creados para personas y no para computadoras, se encuentran ciertas inconsistencias. Dichos recursos proporcionan información detallada a nivel léxico; sin embargo, omiten la información pragmática necesaria para determinar un sentido [32]. Por ejemplo, la relación entre *ash* y *tobacco*, *cigarette* y *tray* no sería directa en una red semántica; sin embargo en el *Brown Corpus*, *ash* co-ocurre muy frecuentemente con ambos vocablos.

b. Uso de tesauros

El tesauro es un sistema que organiza el conocimiento basado en conceptos que muestran relaciones entre vocablos. Las relaciones expresadas comúnmente incluyen jerarquía, equivalencia y asociación (o relación). Los tesauros también proporcionan información como sinonimia, antonimia, homonimia, etc.

El tesauro de Roget denominado en ingles *Rogets´s Internacional Thesaurus*, fue convertido a versión electrónica en 1950, y ha sido el más usado en una amplia variedad de aplicaciones, tales como traducción automática [53], recuperación de información [73] y análisis de contenido [71]. Dicho tesauro suministra una jerarquía de conceptos de ocho niveles. Típicamente, la ocurrencia de un mismo vocablo bajo diferentes categorías, representa un sentido diferente, de tal manera que un conjunto de términos asociados en una misma categoría se encuentran relacionados semánticamente.

La hipótesis básica para la desambiguación basada en tesauros, establece que cada una de las categorías semánticas de los vocablos que forman el contexto del término ambiguo, determinan la categoría semántica de todo el contexto. Finalmente, esta categoría es la que elige el sentido del vocablo ambiguo. Patrick [59] usó este tesauro para discriminar sentidos de verbos, examinando los grupos semánticos formados por cadenas derivadas del mismo tesauro [12][13]. Dicho método es capaz de discriminar el sentido correcto de verbos como *inspire* (*to raise the spirits vs. to inhale, breathe in, sniff*, etc.), *question* (*to doubt vs. to ask a question*) con un alto grado de confiabilidad.

Yarowsky [82] formó clases de palabras tomando como iniciales las definidas en las categorías del tesauro de Roget. Luego, para cada integrante de las clases formadas, obtuvo un contexto de cien palabras extraídas de la enciclopedia de Grolier, que en inglés se denomina *Grolier's Encyclopedia*, y usando técnicas estadísticas basadas en información mutua, identificó aquellas que co-ocurren con los miembros de las clases formadas. Los grupos resultantes son usados para desambiguar nuevas ocurrencias de vocablos polisémicos, comparando un contexto de cien vecinos del término ambiguo, con los miembros de cada grupo. Finalmente, usó la regla de Bayes para escoger un grupo, y como cada uno de ellos está enlazado a un sentido, etiquetó semánticamente la palabra polisémica. Este método reportó 92% de seguridad al realizar pruebas con vocablos ambiguos que expresan tres sentidos.

Al igual que los diccionarios electrónicos, un tesauro es un recurso creado para humanos, por consiguiente no contiene información correcta acerca de las relaciones entre palabras. Es ampliamente conocido que en los niveles más altos de la jerarquía de conceptos existe cierta contrariedad (auque esto es cierto para cualquier jerarquía de

conceptos), debido a que los conceptos de dichos niveles son muy amplios como para poder establecer categorías semánticas. Pese a estos problemas, los tesauros proporcionan una red muy rica en cuanto a asociaciones de palabras y un conjunto de categorías semánticas potencialmente importantes para el procesamiento del lenguaje natural.

c. Uso de diccionarios orientados a la computación

Los diccionarios orientados a la computación, que en inglés se denominan *computacional lexicons*, son bases de conocimiento de gran escala, los cuales se empezaron a construir a mediados de los años 80. Algunos ejemplos son WordNet [21][57], CyC [45], ACQUILEX [8], COLMES [26], etc.

Existen dos técnicas fundamentales para la construcción de estos recursos: la técnica enumerativa y la técnica generativa. En la primera, los sentidos para cada vocablo ambiguo son proporcionados de manera explícita. En este grupo se encuentra WordNet, el cual se ha convertido en el diccionario más usado para desambiguación de sentidos de palabras en inglés; sin embargo, no es considerado un recurso perfecto. El problema radica en la especificidad de los sentidos definidos en WordNet; es decir, la manera tan perfecta como éstos han sido definidos, lo cual muchas veces no se ajusta a las necesidades requeridas por las aplicaciones de procesamiento de lenguaje natural, y en especial WSD. Es necesario mencionar que aún no se tiene definido el nivel de especificidad necesaria para poder detectar las diferencias entre varios sentidos de un vocablo ambiguo, inclusive no es posible afirmar si las jerarquías de WordNet pueden o no satisfacer dichas necesidades. Actualmente, la comunidad científica para el procesamiento de lenguaje natural, está enfocando sus investigaciones a esta área.

Recientemente, algunos trabajos en WSD han utilizado diccionarios computacionales generativos [62]. En estos recursos, las relaciones entre sentidos no son prescritas de manera explícita, sino son generadas por reglas que capturan las regularidades existentes al momento de crear las definiciones de dichos sentidos (como metonimia, meronimia, etc.). Buitelaar [15], especifica que la desambiguación de sentidos usando un contexto generativo, empieza con un etiquetado semántico, el cual apunta a una representación compleja del conocimiento, capturando los sentidos relacionados a un vocablo ambiguo de manera sistemática. Como segundo paso, el procesamiento semántico debe derivar interpretaciones dependientes del discurso, obteniendo información más precisa acerca del sentido de la ocurrencia dada. Buitelaar describe el uso de CORELEX, un diccionario para etiquetado semántico no especificado [72][62].

La gran limitante de estos recursos son sus dimensiones, ya que son más grandes que los descritos anteriormente. Buitelaar [15], describe una técnica que genera entradas

automáticas para CORELEX empleando un corpus; así como los beneficios que se obtienen cuando se usan diccionarios de gran escala. Usando este método propuesto, es posible crear diccionarios orientados a un dominio específico.

2.4.2 Métodos basados en corpus

Un método basado en corpus explota un repositorio de ejemplos, a partir de los cuales se generan modelos matemáticos caracterizados por el uso de métodos empíricos. A mediados de los años 60, el uso de métodos estadísticos se vio menguado, debido al descubrimiento de reglas lingüísticas formales, tales como las teorías de Zellig Harris [27] y las teorías transformacionales de Noam Chomsky [16], de tal manera que los estudios se enfocaron a los análisis lingüísticos, tomando como referencia las oraciones en vez del texto como un todo; así como el uso de ejemplos orientados a dominios específicos. Durante los siguientes diez a quince años sólo un pequeño grupo de lingüistas siguieron trabajando con corpus, frecuentemente con fines lexicográficos y pedagógicos. Pese a ello, en esta época se desarrollaron algunos corpus importantes tales como: *Brown Corpus* [40], *Trésor de la Langue Française* [33], *Lancaster-Oslo-Bergen* (*LOB*) *Corpus* [35], etc.

a. Desambiguación supervisada

Los métodos de desambiguación supervisada etiquetan semánticamente un vocablo ambiguo tomando como referencia un repositorio de sentidos compilado previamente. Éste es entrenado con un corpus desambiguado (*training data*), donde para cada ocurrencia de una palabra ambigua, se toma el sentido de dicho vocablo y el contexto en el que se presenta. De esta manera, los algoritmos de aprendizaje pueden inferir, generalizar y aplicar reglas estadísticas e información lingüística tomando en cuenta dicho repositorio. Es necesario señalar, que los algoritmos de aprendizaje toman en cuenta los ejemplos del recurso como un conjunto de categorías, y que las personas que lo preparan han comprendido esta información, de tal manera que pueden combinarla con su propio conocimiento. En definitiva, el objetivo fundamental de la desambiguación supervisada es construir clasificadores capaces de diferenciar sentidos, basándose en el contexto adquirido previamente.

El principal problema de los métodos basados en aprendizaje supervisado, es la carencia de grandes corpus etiquetados semánticamente, los cuales no son suficientes para el entrenamiento de los clasificadores, así como las enormes cantidades de información generadas para cada sentido de un vocablo ambiguo. Pese a que los corpus etiquetados semánticamente de manera manual son extremadamente costosos, existen algunos recursos de este tipo, tales como: *The Linguistic Data Consortium* que proporciona aproximadamente 200,000 oraciones tomadas del *Brown Corpus*, el *Wall Street Journal*, en

el que las ocurrencias de 191 palabras ambiguas son etiquetadas manualmente con los sentidos de WordNet, el *Cognitive Science Laboratory* proporciona 1000 palabras tomadas del *Brown Corpus*, las cuales también han sido etiquetadas tomando como referencia WordNet.

Debido a la falta de estos recursos, se han realizado muchos trabajos para etiquetar automáticamente diversos corpus de entrenamiento, usando métodos basados en *bootstrapping*. Hearst [30] propuso un algoritmo, en cuya fase de entrenamiento se etiqueta semánticamente cada ocurrencia de un conjunto de sustantivos tomando en cuenta el contexto en el que aparece cada uno. Luego, la información estadística extraída del contexto es usada para desambiguar otras ocurrencias. Cuando algún vocablo es desambiguado con éxito, el sistema automáticamente adquiere información estadística adicional procedente del contexto de dicho término, mejorando el recurso de entrenamiento de manera incremental. Hearst indica que un conjunto de al menos diez ocurrencias son necesarias para iniciar el procedimiento de desambiguación y, para obtener una precisión alta son necesarias de veinte a treinta ocurrencias. Otro método de *bootstrapping* basado en clases semánticas para dominios específicos ha sido propuesto por Basili [7].

Brown [10] propuso el uso de corpus paralelos para evitar el uso de pequeños recursos etiquetados semánticamente. La idea es que diferentes sentidos de un vocablo polisémico, frecuentemente son traducidos de distintas maneras en otro lenguaje, por ejemplo *pen* en inglés es *stylo* en francés cuando expresa el sentido de escritura, y *enclos* cuando se refiere al sentido de envoltura. De esta manera al definir las ocurrencias de un vocablo ambiguo usando premisas de otro idioma, es posible determinar su sentido automáticamente. Este método tiene algunas limitaciones, como las ambigüedades que son preservadas en el leguaje destino (en francés *souris* y en inglés *mouse*) y la escasa disponibilidad de corpus paralelos de gran escala [22].

Uno de los principales problemas que presentan los métodos de desambiguación basados en corpus es conocido como *data sparseness*. Éste es causado por el uso de pequeños corpus de entrenamiento y por la diferencia de frecuencias que presentan los sentidos de un vocablo ambiguo en un corpus textual. Por ejemplo en el *Brown Corpus* (un millón de palabras), la palabra *ash* ocurre ocho veces, de las cuales sólo una de ellas hace referencia al sentido de *árbol*. Inclusive algunos sentidos de términos polisémicos no se encuentran en dicho corpus y para que un algoritmo de desambiguación sea exitoso, debe de asegurar que todos los sentidos de los vocablos polisémicos sean cubiertos.

b. Desambiguación no supervisada

La diferencia entre algoritmos de desambiguación supervisada y no supervisada, radica en que los primeros crean clasificadores usando ejemplos obtenidos de textos etiquetados semánticamente, los que a su vez han sido construidos de forma manual. Los métodos de desambiguación no supervisada usan la misma información para inferir características léxico-ambiguas en textos no etiquetados, obteniendo volúmenes de información más densos, solucionando parcialmente el problema de *data sparseness*, el cual parece estar estrechamente ligado con el uso de pequeños corpus etiquetados. Dichos métodos son implementados usando modelos basados en clases y en similitud.

Los modelos basados en clases obtienen un conjunto de palabras que pertenecen a una categoría común, la cual es llamada clase. Brown [11], propone un método en el que dichas clases son derivadas de las propiedades de distribución inmersas en un corpus, mientras que otros autores usan información externa para definirlas. Resnik [61] usa las categorías de WordNet, Yarowsky [82] usa las categorías del tesauro de Roget, Luk [51] usa conjuntos conceptuales derivados de las definiciones de LDOCE (*Longman Dictionary of Contemporary English*).

Estos métodos confían en la premisa: *palabras de una misma clase comparten una temática común*, la cual es muy ambiciosa ya que la información contenida en dichas clases no siempre está relacionada con alguna temática específica. Por ejemplo, *residue* es un hiperónimo de *ash* en WordNet y sus hipónimos forman la clase {*ash*, *cotton*, *seed*, *cake, dottle*}. Obviamente los vocablos de este conjunto, se combinan de manera muy diferente en un texto, por ejemplo *volcano* está muy relacionado con *ash*; pero tiene poca o ninguna relación con los otros miembros del conjunto.

Los modelos basados en similitud explotan la misma idea; con la diferencia de que los vocablos no son agrupados en clases fijas, de tal manera que cada uno de ellos tiene un conjunto diferente de términos similares. Estos modelos explotan métricas entre patrones de co-ocurrencia. Dagan [19] experimentó con la pareja de vocablos (*chapter, describes*) los cuales pese a pertenecer a la misma clase, no aparecen juntos en el corpus usado; sin embargo los términos similares a chapter, tales como: *book*, *introduction*, *section* aparecen como pareja de *describes* en el mismo corpus. La evaluación de Dagan muestra que los métodos basados en similitud tienen mejor rendimiento que los basados en clases. McCarthy *et al*. [54] presentó un algoritmo que utilizando el tesauro de Lin, WordNet y ciertas medidas de similitud semántica, asigna el sentido más predominante a un vocablo ambiguo. En sus experimentos con sustantivos obtuvo 64% de aciertos.

Finalmente, el porcentaje de aciertos proporcionado por sistemas de desambiguación no supervisada es de 5% a 10% menor que los obtenidos con algoritmos supervisados.

2.4.3 El rol del contexto

El contexto es el indicador más relevante para identificar el sentido que expresa un vocablo ambiguo; por consiguiente, la mayoría de trabajos en WSD están enfocados a discriminar sentidos basados en él. Éste es usado de dos maneras:

- Como *bolsa de palabras*, la cual está conformada por un conjunto de términos alrededor del vocablo ambiguo. Para ello, se puede tomar en cuenta la distancia que existe entre los integrantes del contexto y el término ambiguo, de tal manera que las relaciones gramaticales existentes en la oración son ignoradas.

- Como *información relacionada*, donde un conjunto de palabras son agrupadas tomando en cuenta algún tipo de relación o vínculo con el vocablo a desambiguar. Éstas pueden ser relaciones sintácticas, propiedades ortográficas, colocaciones gramaticales, categorías semánticas, preferencias de selección u otras.

El término contexto expresa un concepto muy amplio. Es por ello, que dependiendo del origen y la distancia de las palabras con respecto al término ambiguo, el contexto ha sido dividido en micro-contexto, macro-contexto y dominio o temática específica.

a. Micro-contexto

El contexto local o micro-contexto está conformado por las palabras más cercanas al vocablo ambiguo, las cuales son seleccionadas teniendo en cuenta diversas características, tales como distancia, relaciones sintácticas y colocaciones gramaticales. Algunos trabajos basados en corpus usan micro-contexto. Un ejemplo puede ser encontrado en Weiss [78]. Asimismo, ciertas técnicas basadas en diccionarios, usualmente no diferencian otro tipo de contexto que no sea el micro-contexto.

Schütze [70] afirma que los métodos que tratan el contexto como *bolsa de palabras*, han logrado mejores resultados para sustantivos que para verbos; pero en general son menos efectivos que los métodos que toman en cuenta otro tipo de relaciones. Yarowsky [82], afirma que esta técnica es menos costosa que aquellas que requieren procesamientos más complejos y sus resultados pueden lograr niveles de desambiguación aceptables para determinadas aplicaciones.

Distancia

Esta característica hace referencia a un grupo de palabras cuya distancia con respecto al vocablo ambiguo no supera cierto umbral de referencia establecido. Los primeros trabajos realizados en WSD tomaban como contexto las palabras más cercanas al término a desambiguar. Este método que si bien es cierto, ha dado buenos resultados; aún dista mucho de ser lo suficientemente confiable.

Kaplan [36] afirma que existen pocos estudios que han intentado establecer la distancia óptima para lograr una desambiguación más confiable. Choueka y Lusignan [17] afirman que un contexto conformado por dos palabras es altamente confiable para lograr una desambiguación exitosa, incluso afirman que con una sola palabra es posible lograr un 80% de éxito.

Yarowsky [83] examinó el impacto de diversos umbrales tomando en cuenta palabras y duplas de palabras, ordenándolas dependiendo del acierto que tuvieron al desambiguar ocurrencias previas de términos ambiguos. Yarowsky llegó a la conclusión de que el umbral varía de acuerdo al tipo de ambigüedad. Afirma que las ambigüedades locales requieren un contexto de tres a cuatro palabras, mientras que las ambigüedades contextuales requieren de veinte a cincuenta palabras; sin embargo no se reportó una medida específica, ya que según sus experimentos cada vocablo ambiguo requiere diferentes umbrales y relaciones. Es necesario resaltar que Yarowsky usó como información adicional las categorías gramaticales de las palabras del contexto, lo cual hace difícil determinar el impacto del umbral en el proceso de desambiguación [84][85].

Colocaciones

Una colocación gramatical es un conjunto de dos o más palabras las cuales expresan una idea específica. El significado que expresa cada término de una colocación difiere de la semántica que dichos vocablos proporcionan cuando se usan de manera conjunta. El idioma inglés presenta muchas colocaciones; por ejemplo *cristal clear*, *middle management*, *nuclear family*, *cosmetic surger*, etc. Otro ejemplo más claro en cuanto al uso de colocaciones puede notarse en la expresión *red in the face*, la cual hace referencia a una persona apenada o sonrojada; sin embargo, *blue in the face* hace referencia a una con hambre. Por ende, no sería común escuchar expresiones como *yellow in the face* o *green in the face*, ya que éstas denotarían posibles errores idiomáticos.

Algunos lingüistas, como Khellmer [38], argumentan que el *diccionario mental* del ser humano está conformado por colocaciones y vocablos individuales, de tal manera que un mismo vocablo relacionado con diferentes palabras expresa significados heterogéneos, como por ejemplo *bank river* y *bank investment.*

Kintsch y Mross [39] demostraron que el segundo término de una colocación define el sentido del primero. Yarowsky [83] fue uno de los primeros investigadores que usó colocaciones gramaticales para desambiguar sentidos de palabras. Él definió colocación gramatical como la co-ocurrencia de dos vocablos con alguna relación específica. Asimismo, concluyó que en el caso de ambigüedad binaria, existe un sentido por colocación, es decir, que el sentido que expresa un vocablo polisémico está determinado por el segundo término de la colocación a la que pertenece.

Relaciones sintácticas

En muchos trabajos de WSD, la información sintáctica es utilizada para determinar el sentido de una palabra ambigua. Wilks [79][80] combina colocaciones gramaticales y relaciones sintácticas. Kelley [37], Dahlgren [20] y Atkins [5] especificaron ciertas reglas que justifican la presencia o ausencia de determinantes, pronombres, complementos de sustantivos, preposiciones, relaciones sujeto-verbo, verbo-objeto entre otras.

Hearst [30] experimentó desambiguando sustantivos. Para ello, segmentó el texto en grupos de verbos, frases preposicionales y de sustantivos. Luego, examinó ítems que se encontraban aproximadamente a tres segmentos de frase del vocablo ambiguo. Yarowsky [83] observó varios tipos de comportamiento basados en la categoría sintáctica de las palabras y llegó a la conclusión de que el éxito de desambiguación de un vocablo puede depender de las estructuras sintácticas que lo rodean. De esta manera, afirma que los verbos derivan mejor información de sus objetos que de sus sujetos, que los adjetivos derivan mayor información de los sustantivos a los que modifican, y que los sustantivos son mejor desambiguados cuando tienen adjetivos o sustantivos adyacentes.

En trabajos recientes, la información sintáctica sólo es usada para categorizar gramaticalmente las palabras en conjunción con otra clase de información. La evidencia sugiere que la desambiguación exitosa de un vocablo depende del método usado, el cual debe de elegirse tomando en cuenta la categoría gramatical y características específicas del vocablo ambiguo [53].

b. Macro-contexto

El macro-contexto, que en inglés se denomina *topical context*, está conformado por palabras de gran contenido semántico (por lo general sustantivos, adjetivos y verbos), las cuales co-ocurren con un sentido específico del vocablo ambiguo, usando varias oraciones como fuente de información. A diferencia del micro-contexto, el cual ha sido muy usado desde mediados del siglo pasado, el macro-contexto ha sido menos utilizado. Los métodos que hacen uso de esta característica, explotan la redundancia en el texto; es decir, intentan ubicar grupos de palabras que se encuentren semánticamente relacionadas con un tópico

específico. Por ejemplo, sería posible determinar el sentido *juego de béisbol* al que hace referencia el vocablo ambiguo *base*, si su macro-contexto está conformado por términos como *pitcher* y *ball*.

Por lo general, los trabajos basados en macro-contexto, generalmente usan una *bolsa de palabras* que agrupa un conjunto no ordenando de términos con gran contenido semántico, los cuales son encontrados en el macro-contexto del término ambiguo.

Yarowsky [82] usó un contexto de 100 palabras para obtener clases de términos relacionados semánticamente. Asimismo, usó dicha cantidad para desambiguar un vocablo polisémico tomando como referencia el tesauro de Roget. Voorhees [77] experimentó con diferentes métodos estadísticos, usando dos oraciones como *topical context*. Gale [24] tomó un contexto de aproximadamente cincuenta palabras, indicando que mientras éstas son más cercanas al vocablo ambiguo, la elección del sentido es más confiable. Los resultados obtenidos por Gale sólo mejoraron de 86% a 90% cuando amplió el número de palabras en el contexto, de 6 (cantidad de términos que generalmente se usan como micro-contexto) a 50 palabras.

En un estudio similar, Gale [23] llegó a la conclusión que si en un discurso se presentan diferentes ocurrencias de un vocablo ambiguo, es muy probable que éstas hagan referencia al mismo sentido. Leacock [43][44] demostró que ambos tipos de contexto (micro-contexto y *topical context*) son necesarios para obtener resultados más confiables. Los estudios de Yarowsky indican que la información obtenida del macro-contexto puede ser usada para desambiguar sustantivos, mientras que para verbos y adjetivos los resultados disminuyen dramáticamente cuando se usan muchas oraciones como parte del contexto [84].

Brown y Yule [9] sugieren que los métodos que utilizan contextos muy amplios, deberían de dividir el texto en sub-tópicos, los que a su vez deberían de agruparse en secciones de textos que se encuentren conformados por diferentes párrafos. Ellos afirman que la segmentación automática de texto en unidades más pequeñas, podría ser de mucha ayuda para dichos métodos. Leacock [44] consideró el rol del micro y macro-contexto, intentando definir el impacto de cada uno de ellos. Sus resultados indican que para clasificadores estadísticos, el micro-contexto es superior al macro-contexto.

c. Dominio

El término dominio en lenguaje natural puede ser definido como una agrupación de diferentes fuentes de información textuales que hagan referencia a un tópico específico de información. El hecho de desambiguar sentidos usando un dominio, se encuentra implícito en varias técnicas de WSD basadas en inteligencia artificial, como la propuesta por Schank

[68], en la cual se elige un sentido tomando en cuenta no sólo el contexto del término ambiguo; sino el sentido general del tópico o discurso. Los métodos basados en la hipótesis: una palabra sólo tiene un sentido por discurso, demuestran sus limitaciones cuando sólo usan esta fuente de información para desambiguar sentidos. Por ejemplo en la oración *the lawyer stopped at the bar for a drink*, si sólo se confiara en la información del tópico o dominio, el vocablo *bar* asumiría un sentido incorrecto, ya que para este caso el tópico hace referencia a leyes.

Gale [23] cuestiona en demasía el hecho que una palabra sólo tenga un sentido por discurso. Dahlgren [20] afirma que el dominio no elimina la ambigüedad para algunas palabras. Para ello, toma como ejemplo el sustantivo *hand* el cual tiene aproximadamente 16 sentidos y puede expresar 10 de ellos en cualquier texto.

La influencia del dominio para etiquetar semánticamente una palabra, depende de factores como el estilo del texto (cuan técnico es por ejemplo) y la relación existente entre los sentidos del vocablo ambiguo; es decir, si éstos están fuerte o débilmente polarizados, así como su uso especializado y su uso común. Por ejemplo, en la enciclopedia francesa *Encyclopaedia Universalis*, el término *intérêt* (interés) aparece 62 veces en el artículo *Interés-Finanzas*, en el que hace referencia al sentido de finanzas, 139 veces en el artículo *Interés-Filosofía*, en el que hace referencia a un sentido no financiero; sin embargo en el artículo *Tercer mundo*, aparece dos veces para cada uno de estos sentidos.

2.5 Medidas de similitud semántica

Muchas aplicaciones que procesan lenguaje natural se basan en la similitud semántica de unidades lingüísticas de diversa índole (palabras, sentidos, oraciones, documentos). Así, tareas como la desambiguación de sentidos de palabras, la detección de cadenas léxicas, el establecimiento de relaciones inter o intra-documentales, la agrupación de textos, etc., necesitan medidas precisas de similitud semántica.

En WSD, la similitud entre unidades lingüísticas es un caso particular de la similitud entre conceptos. La similitud semántica es aquella en que las unidades a comparar se proyectan sobre un espacio semántico (un diccionario computacional un tesauro, una ontología) usando algún algoritmo de comparación específico.

El método propuesto cuenta con un módulo de etiquetado automático de sentidos, el cual determina cierta magnitud semántica entre dos conceptos, usando para ello una medida de similitud específica y tomando como referencia el diccionario computacional WordNet. A continuación se describen ciertas medidas de similitud, las cuales han sido propuestas

para medir la proximidad semántica entre conceptos. Es posible implementar estas medidas sobre diferentes espacios semánticos, tales como WordNet, el tesauro de Roget, etc. Finalmente, es necesario aclarar que dichas medidas han sido implementadas para el idioma inglés tomando a WordNet como espacio semántico.

2.5.1 Medida de Lesk

El algoritmo original de Lesk [46] determina el sentido de un vocablo ambiguo, comparando su glosa (definición del sentido de una palabra en WordNet), con las glosas correspondientes a los términos considerados como sus vecinos contextuales. El sentido elegido, es aquel que tiene mayores intersecciones o traslapes (*word overlapping*) con las glosas de cada uno de sus vecinos. Existen dos hipótesis que respaldan esta técnica: la primera asegura que es posible elegir el sentido de un vocablo ambiguo teniendo en cuenta los sentidos que expresan cada uno de sus vecinos. Intuitivamente los vocablos que tienden a aparecer juntos deben de estar relacionados de alguna manera, ya que éstos normalmente trabajan juntos para comunicar alguna idea.

La segunda hipótesis asegura que es posible identificar que dos sentidos están relacionados, si es que se encuentran palabras coincidentes en las definiciones de cada uno de ellos. La intuición en este punto es igual de razonable que la anterior, ya que dos sentidos relacionados, frecuentemente son definidos usando las mismas palabras, y por consiguiente debería de existir una referencia común en ambas definiciones. Por ejemplo, en la oración *the rate of interest at my bank is…*, cualquier ser humano podría intuir que *bank* hace referencia a una institución financiera y no a un banco de peces, ya que como puede observarse cada uno de los vecinos de *bank* expresa un significado financiero. En WordNet, las glosas de las palabras *rate, interest* y *bank* tienen vocablos en común; ya que *interest* y *bank* son definidas usando los términos *money* y *mortgage*, y las glosas de *interest* y *rate* comparten el término *charge*.

La limitación principal de esta técnica es que las glosas del diccionario, por lo general, son muy breves, de tal manera que no incluyen suficiente vocabulario para identificar los sentidos relacionados. Banerjee–Pedersen [6] sugieren una adaptación del algoritmo basándose en WordNet. Esta adaptación consiste en tomar en cuenta las glosas de los vecinos del vocablo ambiguo, explotando los conceptos jerárquicos de WordNet. Asimismo, ellos sugieren una variación en la manera de asignar el puntaje a una glosa, de tal manera que si *n* palabras consecutivas son iguales en ambas glosas, éstas deberán de tener mayor puntaje que si sólo coincidiera una sola palabra.

Suponiendo que *bark* es el vocablo ambiguo y sus vecinos son *dog* y *tail*. El algoritmo original de Lesk obtendría las coincidencias de términos entre las glosas de los

sentidos de *dog* y *bark*. Luego obtendría las coincidencias entre las glosas de *tail* y *bark*. El sentido de *bark* que obtenga el máximo número de coincidencias es el seleccionado. La adaptación del algoritmo de Lesk considera estas mismas coincidencias y además añade las glosas de los sentidos de las palabras que se encuentran relacionadas semántica o léxicamente con *dog, bark* y *tail*, de acuerdo a las jerarquías de WordNet.

2.5.2 Medida de Leacock–Chodorow

Esta medida está basada en el concepto de longitudes de rutas [41][42]. La ruta más corta entre dos conceptos es aquella que incluye el menor número de conceptos intermedios. Este valor es escalado por la profundidad de la jerarquía usada, la cual está definida como la longitud entre el nodo raíz y el nodo más lejano. Dicha medida está definida por la ecuación 1.

$$\text{relación}_{\text{lch}}(c_1, c_2) = \max[-\log(\text{RutaMasCorta}(c_1, c_2)/(2 \times D))] \quad (1)$$

$\text{RutaMasCorta}(c_1, c_2)$ es la longitud de la ruta más corta entre dos conceptos y D es la profundidad máxima de la taxonomía usada. Esta medida ha sido implementada en la librería *WordNet::Similarity* [61], tomando como referencia la jerarquía de hiperónimos de WordNet, la cual está definida sobre sustantivos. El máximo número de niveles en esta jerarquía es 16.

2.5.3 Medida de Resnik

Resnik introduce una medida de relación semántica, basada en el concepto de *contenido de información* (*information content*). Para ello, es necesario asignar un valor a cada concepto de la jerarquía, basándose en la evidencia encontrada en un corpus [64].

Dicho término es una simple medida de la especificación de un concepto. Un concepto con gran *contenido de información* es muy específico a un tópico particular, mientras que uno con bajo *contenido de información* está asociado a tópicos más generales. Por ejemplo, la expresión *carving fork* tiene un alto *contenido de información*, mientras que *entity* tiene un bajo *contenido de información.*

El *contenido de información* de un concepto es estimado contando su frecuencia en un corpus de gran escala, determinándole su probabilidad estadística. De acuerdo a Resnik, el logaritmo negativo de esta probabilidad determina el *contenido de información* del concepto. Ver ecuación 2.

$$\text{IC}(\text{concepto} = -\log(\text{P}(\text{concepto})) \quad (2)$$

Si se tuviera un texto de gran escala etiquetado semánticamente, contar la frecuencia de un concepto no sería complicado, ya que cada concepto estaría asociado a un sentido; pero en caso contrario, Resnik sugiere contar el númerode ocurrencias de una palabra en el corpus y luego dividir ese valor por el número de sentidos que tiene el término ambiguo, asignando este valor a cada concepto. Por ejemplo, supongamos que el término *bank* ocurre 20 veces en un corpus, y existen dos conceptos asociados a dicho vocablo, uno para *river bank* y el otro para *financial bank*. Cada uno de estos conceptos recibiría un valor de 10; en cambio si las ocurrencias de *bank* se presentaran en un texto etiquetado con sentidos, obviamente la información sería más consistente.

La frecuencia de un concepto incluye la frecuencia de todos sus conceptos subordinados, ya que el conteo de un concepto es añadido a su inmediato superior. Es necesario notar que los conteos de los conceptos más específicos son añadidos a los más genéricos; y no de manera contraria. Los conceptos ubicados en los niveles más altos de una jerarquía tendrán mayor probabilidad, lo que significará un bajo *contenido de información*, ya que éstos representan conceptos muy generales.

La idea principal de esta medida es: la mayor o menor relación semántica entre dos conceptos depende de la cantidad de información que ellos comparten en común. Esta cantidad está determinada por el *contenido de información* común entre un nodo común y un par de conceptos, que en inglés se denomina *least common subsumer* (LCS). La medida de Resnik se calcula con la ecuación 3.

$$\mathrm{sim}_{\mathrm{res}}(c_1, c_2) = \mathrm{IC}(\mathrm{lcs}(c_1, c_2)) \tag{3}$$

Esta medida no considera el *contenido de información* del par de conceptos a comparar, ni tampoco la longitud de la ruta entre ambos. La principal limitante de esta técnica es que algunos pares de conceptos compartirían el mismo valor de similitud, ya que existe la posibilidad de que el mismo LCS sea asignado a más de un par de conceptos. Por ejemplo, *vehicle* es el LCS de *jumbo jet*, *house trailer* y *ballistic missile*. Por ende, estas parejas recibirían el mismo puntaje en su comparación.

2.5.4 Medida de Jiang–Conrath

Jiang–Conrath [34] usan el concepto de *contenido de información* planteado por Resnik y las longitudes de rutas entre conceptos. Esto resulta una técnica híbrida para computar la relación semántica de una pareja de conceptos. Esta técnica incluye el *contenido de información* de ambos conceptos, así como su LCSVer la ecuación 4.

$$\mathrm{dist}_{\mathrm{jcn}}(c_1, c_2) = \mathrm{IC}(c_1) + \mathrm{IC}(c_2) - 2 \times \mathrm{IC}(\mathrm{lcs}(c_1, c_2)) \tag{4}$$

2.5.5 Medida de Lin

La medida de Lin [51] está basada en su teorema de similitud. Éste establece que la similitud entre dos conceptos está dada por la razón entre la cantidad de información necesaria para establecer la información común de ambos conceptos y la necesaria para describirlos. Dicha información se encuentra reflejada en el *contenido de información* reportado por el LCS y el de cada uno los conceptos propiamente dichos.

Esta medida es muy parecida a la presentada por Jiang–Conrath; aunque ambas fueros desarrolladas independientemente. Dicha medida puede ser vista como la intersección del *contenido de información* de los dos conceptos a comparar, dividido por la suma del *contenido de información* de ambos, tal como se muestra en la ecuación 5.

$$\text{related}_{\text{lin}}(c_1, c_2) = \frac{2 \times \text{IC}(\text{lcs}(c_1, c_2))}{\text{IC}(c_1) + \text{IC}(c_2)} \tag{5}$$

Capítulo 3
Infraestructura empleada

El propósito de este capítulo es describir detalladamente los diferentes recursos de información utilizados en este trabajo, tales como WordNet, SemCor; y herramientas como MINIPAR y WordNet::Similarity, la cual es una librería que implementa algunas de las medidas de similitud semántica explicadas en el capítulo anterior.

3.1 Analizador sintáctico MINIPAR

El análisis sintáctico, más conocido en inglés como *parsing*, es un proceso que consiste en tomar cierta entrada textual y producir como salida una representación estructurada de dicha información. Un analizador sintáctico para un lenguaje natural como inglés, español, etc., es un programa que construye árboles de estructura de frase o de derivación para las oraciones de dicho lenguaje. Un buen analizador sintáctico, debe de proporcionar un análisis gramatical correcto para cada oración ingresada, separando los términos en constituyentes y etiquetando cada uno de ellos. Asimismo, debe de suministrar información adicional acerca de las clases semánticas (persona, género) de cada palabra y también la clase funcional (sujeto, objeto directo, etc.) de los constituyentes de la oración.

MINIPAR [50] es un conjunto de librerías que proporcionan la funcionalidad necesaria para realizar un análisis sintáctico de amplia cobertura para el idioma inglés. Dichas librerías permite representar una oración como una red de nodos y enlaces, donde los nodos representan las categorías gramaticales y los enlaces los tipos de relaciones de dependencia. MINIPAR fue evaluado con el corpus SUSANNE, un subconjunto del corpus *Brown*, logrando reconocer el 88% de la relaciones de dependencia, con una efectividad del 80% sobre las mismas. Alam [4] realizó un experimento usando 584 oraciones, donde MINIPAR logró reconocer de manera correcta 77.6% de los términos analizados. Asimismo, en dicho trabajo se determinó que la mayoría de errores de MINIPAR inciden en las siguientes categorías:

- Errores de etiquetado, en los cuales algunos sustantivos son etiquetados como verbos.
- Errores de enlaces (*attachment*), en los cuales algunas frases preposicionales que deben de ser ligadas al sustantivo inmediato anterior, son ligadas a los verbos.
- Entradas léxicas faltantes, en las cuales algunas palabras específicas tales como *download*, no se encuentran en el diccionario de MINIPAR. Esto introduce errores, ya que por defecto, los vocablos no encontrados son etiquetados como sustantivos.

– Falta de capacidad para analizar oraciones no gramaticales, ya que en el mundo real es imposible esperar que el usuario ingrese sólo oraciones gramaticales. Si bien es cierto que MINIPAR produce un árbol sintáctico para este tipo de oraciones no gramaticales, dicho árbol no se encuentra formado correctamente; por ende no es posible extraer la información semántica contenida en dicha expresión. La tabla 3 y la tabla 4 listan las diferentes relaciones y categorías gramaticales que son tomadas en cuenta por MINIPAR.

Tabla 3. Relaciones gramaticales proporcionadas por MINIPAR

Relación gramatical	**Descripción/Ejemplo**
Aux	*should* $\xleftarrow{aux}$ *resign*
Be	*is* $\xleftarrow{be}$ *sleeping*
C	*that* $\xleftarrow{c}$ *John loves Mary*
Comp1	Primer complemento
Det	*the* $\xleftarrow{\text{det}}$ *hat*
Gen	*Jane' s* $\xleftarrow{gen}$ *uncle*
Have	*have* $\xleftarrow{have}$ *dissappered*
Inv-aux	Auxiliar invertido. *Will* $\xleftarrow{inv\ aux}$ *you stop it*
Inv-be	Verbo *to be* invertido. *Is* $\xleftarrow{inv\ be}$ *she sleeping*
Inv-have	Verbo *have* invertido. *Have* $\xleftarrow{inv\ have}$ *you slept*
Mod	Relación entre una palabra y su modificador adjunto
Pnmod	Modificador post nominal
P-spec	Especificador de frase preposicional
Pcomp-n	Complemento nominal de preposición
Post	Post determinante
Pre	Pre determinante
Pred	Predicado de una cláusula
Wha, whn, whp	Elemento *wh*
Obj	Verbo de objeto
Subj	Sujeto de verbo
S	Sujeto

3.2 Diccionario WordNet

WordNet es un sistema de referencia léxica, el cual fue desarrollado en la universidad de Princeton bajo la dirección de George A. Miller. Este recurso combina

Tabla 4. Categorías gramaticales proporcionadas por MINIPAR

Categoría Gramatical	**Descripción**
Det	Determinante
PreDet	Pre-Determinante
PostDet	Post-Determinante
NUM	Número
C	Cláusula
I	Frase inflexional
V	Verbo y frase verbal
N	Sustantivo y frase sustantival
NN	Modificador sustantivo-sustantivo
P	Preposición
PpSpec	Frase preposicional
A	Adjetivo/Adverbio
Have	Verbo tener
Aux	Verbo auxiliar
Be	Alguna forma del verbo *to be*
COMP	Complemento
V_N intransitive verbs	Verbo con un argumento
V_N transitive verbs	Verbo con dos argumentos
V_N_I complement	Verbo seguido de cláusula as
U	No definido

muchas características usadas para WSD en un solo sistema ya que define los diferentes sentidos que puede presentar un vocablo, agrupa conjuntos de sinónimos o *synsets*, los cuales representan un concepto léxico común y organiza diversas jerarquías de relaciones existentes entre palabras.

Los sentidos de WordNet comprenden un conjunto de sinónimos o *synsets*, los cuales aparecen al inicio de la glosa (término empleado para hacer referencia a la definición de un sentido). Asimismo, el número que se encuentra al inicio de algunas glosas es la frecuencia de los valores obtenidos del corpus SemCor. Estas características pueden ser apreciadas en la tabla 5, cuya información ha sido extraída de este diccionario. Dicha tabla muestra los diferentes sentidos para el sustantivo *car*.

WordNet está conformado por tres bases de datos correspondientes a sustantivos, verbos y una para adjetivos y adverbios. Cada una de ellas, está conformada por entradas léxicas que corresponden a formas ortográficas individuales. Asimismo, las diferentes

jerarquías creadas, toman en cuenta la categoría gramatical de la palabra. Las tablas 6, 7 y 8 muestran dichas jerarquías para sustantivos, verbos y, adjetivos y adverbios.

Las relaciones de sinonimia e hiponimia, son las que con más frecuencia se usan en WordNet. Es posible definir un conjunto de sinónimos (*synset*) como diferentes lemas que hacen referencia al mismo significado. Dos lemas de WordNet son considerados sinónimos si ellos pueden ser sustituidos en un contexto tal como se puede apreciar en el *synset* del vocablo *car*, conformado por los términos *auto, automobile, machina* y *motorcar*.

Tabla 5. Sentidos del sustantivo "car" según WordNet 2.0

Sentido	**Definición de glosa**
1	*(598) car, auto, automobile, machine, motorcar -- (4-wheeled motor vehicle; usually propelled by an internal combustion engine; "he needs a car to get to work").*
2	*(24) car, railcar, railway car, railroad car -- (a wheeled vehicle adapted to the rails of railroad;"three cars had jumped the rails").*
3	*(1) cable car, car -- (a conveyance for passengers or freight on a cable railway; "they took a cable car to the top of the mountain").*
4	*car, gondola -- (car suspended from an airship and carrying personnel and cargo and power plant).*
5	*car, elevator car -- (where passengers ride up and down; "the car was on the top floor").*

Tabla 6. Jerarquías para sustantivos en WordNet 2.0

Relación	**Definición**	**Ejemplo**
Hypernym	Relación en la cual el significado de una palabra incluye al de otra.	*bird → sparrow*
Hyponym	Relación en la cual el significado de una palabra está incluido en otra.	*lunch → meal*
Has-member	Relación que hace referencia a los miembros que forman parte de un grupo.	*faculty → professor*
Member-of	Relación que hace referencia al grupo al que pertenece un miembro.	*pilot → crew*
Has-part	Relación que hace referencia a las partes que forman un todo.	*table → leg*
Part-of	Relación que hace referencia al todo al que pertenece una parte.	*motor → car*
Antonym	Relación en la que un par de palabras expresan ideas opuestas o contrarias.	*leader → follower*

En vez de representar conceptos usando términos lógicos, WordNet los representa como *synsets*. Cada miembro de un *sysnset* está relacionado con sus *synsets* inmediatos superiores e inferiores mediante relaciones hiperónimas e hipónimas. Para poder encontrar cadenas o relaciones que vayan de lo más genérico a lo más específico, es posible formar cadenas transitivas de relaciones hipónimas o hiperónimas. Un ejemplo de la cadena hiperónima del sustantivo *valley* se muestra en la figura 3, en la cual, los *synsets* más específicos son mostrados en los primeros renglones hasta llegar al concepto más genérico. La cadena empieza en *valley* y termina en *entity*

Tabla 7. Jerarquías para verbos en WordNet 2.0

Relación	**Definición**	**Ejemplo**
Hypernym	Relación en la cual el significado de una palabra incluye al de otra u otras.	*fly → travel*
Troponym	Relación en la cual la acción expresada por un verbo es una manera particular de hacer algo.	*walk → stroll*
Entails	Relación en la cual el primer término refleja una acción causal y el segundo expresa un resultado.	*teach → learn*
Antonym	Relación en la cual un par de palabras expresan ideas opuestas o contrarias.	*leader → follower*

Tabla 8. Jerarquías para adjetivos y adverbios en WordNet 2.0

Relación	**Definición**	**Ejemplo**
Antonym	Relación en la cual un par de palabras expresan ideas opuestas o contrarias.	*heavy → light*

En vez de representar conceptos usando términos lógicos, WordNet los representa como *synsets*. Cada miembro de un *sysnset* está relacionado con sus *synsets* inmediatos superiores e inferiores mediante relaciones hiperónimas e hipónimas. Para poder encontrar cadenas o relaciones que vayan de lo más genérico a lo más específico, es posible formar cadenas transitivas de relaciones hipónimas o hiperónimas. Un ejemplo de la cadena hiperónima del sustantivo *valley* se muestra en la figura 3, en la cual, los *synsets* más específicos son mostrados en los primeros renglones hasta llegar al concepto más genérico. La cadena empieza en *valley* y termina en *entity*.

Las jerarquías de conceptos en WordNet toman en cuenta la categoría gramatical de cada vocablo. Para sustantivos, existe una relación de hiperonimia entre dos conceptos cuando uno de ellos es una clase del otro, tal como se especifica en la tabla 6.

Por ejemplo, *car* es un hiperónimo de *motor vehicle*. Para verbos existe la relación de troponimia, es decir una forma peculiar de ejecutar una acción. Por ejemplo *walking* es un tropónimo de *moving*. Cada jerarquía puede ser visualizada como un árbol, el cual tiene un concepto muy general asociado con un nodo raíz y conceptos más específicos que serían las hojas del árbol. Por ejemplo, un nodo raíz podría representar el concepto *entity*, mientras que los nodos hijos estarían asociados con *carving fork* y *whisk broom.* Las jerarquías hiperónimas de sustantivos comprenden el 70% del total de relaciones existentes en WordNet.

---

valley, vale

=> natural depression, depression

=> geological formation, formation

=> natural object

=> object, physical object

=>entity

Figura 3. Cadena de hiperónimos para el sustantivo "valley".

Las longitudes de rutas entre conceptos han sido empleadas en otras redes para representar relaciones semánticas. Sin embargo, esto sólo es apropiado cuando dichas longitudes tienen una interpretación consistente. Este caso no aplica a WordNet, ya que los conceptos ubicados en las partes altas del árbol son más generales que los que se encuentran ubicados en las partes bajas; por lo tanto la longitud de una ruta entre dos conceptos generales puede sugerir grandes diferencias semánticas, mientras que la longitud entre dos conceptos específicos sugeriría escasa diferencia semántica.

Por ejemplo, *mouse* y *rodent* están separados por una ruta de longitud uno, de tal manera que la diferencia semántica entre ambos es casi nula. El hecho de que la longitud de las rutas puedan ser interpretadas de una manera diferente dependiendo en que parte del árbol de WordNet ocurran, ha permitido el desarrollo de un número significativo de medidas de similitud y relación semántica. Las razones por las cuales el uso de WordNet se ha generalizado en muchas aplicaciones que involucran procesamiento de lenguaje natural, es porque éste proporciona el conjunto más amplio de información léxica agrupado en un solo recurso y por su disponibilidad gratuita; ya que puede ser descargado de la red.

Los primeros usos que se hicieron de WordNet fueron en el área de recuperación de información. Voorhees [66] y Smeaton [67] crearon una base de conocimiento usando las jerarquías de WordNet. Li [47] propuso un algoritmo basado en WordNet para WSD donde la desambiguación era resuelta usando medidas de similitud semántica. Leacock [42] usó WordNet para solucionar los problemas de *data sparseness*. Hawkins [28] construyó sistemas que trabajan con información contextual y de frecuencia basadas en WordNet. Fellabum [21] propuso un sistema que hace uso del agrupamiento (*clustering*) sintáctico y las distinciones semánticas extraídas de WordNet.

WordNet es el recurso más usado para obtener métricas de similitud semántica como las que han sido descritas en el capítulo anterior. Algunos trabajos sobre este tema son los de Agirre y Rigau [3] quienes emplean WordNet para determinar la distancia conceptual entre conceptos, mientras que Mihalcea [56] explota la densidad semántica y las glosas de WordNet para la desambiguación de sentidos de palabras. Otros trabajos que también usan WordNet son los presentados por Jiang–Conrath [34], Agirre y Martinez [2], Haynes [29], Banerjee–Pedersen [6].

3.3 Corpus SemCor

SemCor es un corpus léxico etiquetado semánticamente, el cual fue creado por la universidad de Princeton. Éste es un subconjunto del corpus *English Brown*. Actualmente, SemCor contiene al menos 700,000 palabras etiquetadas con su categoría gramatical y más de 200,000 palabras son proporcionadas con su respectivo lema y número de sentido tomando como referencia WordNet. Las palabras cuya categoría gramatical hace referencia a preposiciones, determinantes, pronombres y verbos auxiliares no son etiquetadas semánticamente, al igual que caracteres no alfanuméricos, interjecciones y términos coloquiales.

Más en detalle SemCor consta de un total de 352 archivos. En 186 de ellos, sustantivos, adjetivos, verbos y adverbios son etiquetados con su categoría gramatical, lema y sentido; mientras que en los 166 textos restantes, sólo los verbos son etiquetados con su lema y sentido. El número total de *tokens* en SemCor es de 359,732 en el primer conjunto de archivos, de los cuales 192,639 han sido etiquetados semánticamente, mientras que el segundo grupo, este número asciende a 316,814 *tokens*, de los cuales 41,497 ocurrencias de verbos han sido etiquetadas semánticamente.

A continuación, la figura 4 muestra un ejemplo del formato usado por SemCor. Para ello, tomaremos como ejemplo la oración *the Fulton County Grand Jury said Friday an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place*.

Como se puede observar en dicha figura, el formato utilizado por el corpus SemCor se encuentra basado en el uso de etiquetas, las cuales son muy similares a las que se utilizan en el formato XML (por sus siglas en inglés *extensible markup language*). De esta manera las estiquetas *<s>...</s>* especifican los límites de una oración, * ... * delimita un párrafo y *<wf ...> </wf>* delimita la información gramatical proporcionada para cada vocablo de una oración.

```
<p pnum=1>
<s snum=1>
<wf cmd=ignore pos=DT>The</wf>
<wf cmd=done rdf=group pos=NNP lemma=group wnsn=1 lexsn=1:03:00::
    pn=group>Fulton_County_Grand_Jury</wf>
<wf cmd=done pos=VB lemma=say wnsn=1 lexsn=2:32:00::>said</wf>
<wf cmd=done pos=NN lemma=friday wnsn=1 lexsn=1:28:00::>Friday</wf>
<wf cmd=ignore pos=DT>an</wf>
<wf cmd=done pos=NN lemma=investigation wnsn=1 lexsn=1:09:00::>
    investigation</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=atlanta wnsn=1 lexsn=1:15:00::>Atlanta</wf>
<wf cmd=ignore pos=POS>'s</wf>
<wf cmd=done pos=JJ lemma=recent wnsn=2 lexsn=5:00:00:past:00> recent</wf>
<wf cmd=done pos=NN lemma=primary_election wnsn=1 lexsn=1:04:00::>
    primary_election</wf>
<wf cmd=done pos=VB lemma=produce wnsn=4 lexsn=2:39:01::> produced</wf>
<punc>“</punc>
<wf cmd=ignore pos=DT>no</wf>
<wf cmd=done pos=NN lemma=evidence wnsn=1 lexsn=1:09:00::>evidence </wf>
<punc>”</punc>
<wf cmd=ignore pos=IN>that</wf>
<wf cmd=ignore pos=DT>any</wf>
<wf cmd=done pos=NN lemma=irregularity wnsn=1 lexsn=1:04:00::>
    irregularities</wf>
<wf cmd=done pos=VB lemma=take_place wnsn=1 lexsn=2:30:00::> took_place</wf>
<punc>.</punc>
</s>
</p>
```

Figura 4. Formato de SemCor

3.4 Librería WordNet::Similarity

WordNet::Similarity, es una librería que implementa medidas de proximidad semántica, las cuales se encuentran basadas en la estructura y contenido de WordNet. Las medidas implementadas en esta librería han sido divididas en medidas de relación y similitud semántica de acuerdo a los trabajos presentados por Budanitsky y Hearst [14].

Las medidas de similitud semántica usan la jerarquía de hiperónimos, y cuantifican cuan parecido o similar puede ser un concepto *A* con respecto a un concepto *B*. Por ejemplo, una medida de similitud debería de mostrar que *automobile* es más parecido a *boat* que a *tree*, debido a que *automobile* y *boat*, comparten a *vehicle* como antecesor en la jerarquía de sustantivos de WordNet. WordNet 2.0 tiene nueve jerarquías de sustantivos que incluyen 80,000 conceptos y 554 jerarquías de verbos que conforman 13,500 conceptos. Algunas de las medidas de similitud implementadas sólo pueden procesarse sobre vocablos que tienen la misma categoría gramatical, como los sustantivos *cat* y *dog*, o los verbos *run* y *walk*. Pese a que WordNet también incluye adjetivos y adverbios, éstos no se encuentran organizados en dicha jerarquía, de tal manera que las medidas de similitud no pueden ser aplicadas.

Sin embargo, los conceptos pueden relacionarse de muchas maneras más allá de ser similares unos con otros. Por ejemplo, *wheel* es una parte de *car*, *night* es el opuesto de *day*, *snow* es hecho de *water*, *knife* es usado para cortar *bread*, etc. WordNet proporciona otras relaciones, tales como: *parte de*, *es hecho de*, y *es un atributo de*. Toda esta información puede soportar la creación de ciertas medidas de relación semántica. Estas medidas tienden a ser más flexibles, y permiten calcular valores que cuantifiquen la relación semántica entre palabras de diferentes categorías gramaticales; por ejemplo el verbo *murder* y el sustantivo *gun*. Esta librería se encuentra disponible en la red, específicamente en *www.d.umn.edu/~tpederse/similarity.html*.

3.4.1 Medidas de similitud semántica

Tres de las seis medidas de similitud semántica que se implementan en esta librería están basadas en el *contenido de información* que aporta un nodo común a un par de conceptos, que en inglés se denomina *least common subsumer* (LCS). El *contenido de información* es un valor que denota la especificidad de un concepto y el LCS es un valor que toma en cuenta el nodo antecesor que proporcione mayor *contenido de información* para ambos conceptos. Estas medidas son las propuestas por Resnik (*res*), Lin (*lin*) y Jiang–Conrath (*jcn*). El corpus que por defecto fue usado para computar los valores correspondientes al *contenido de información* para cada nodo de WordNet es SemCor; sin

embargo, existen programas utilitarios disponibles en la librería *WordNet::Similarity*, que permiten al usuario computar dichos valores usando el *Brown Corpus*, *Penn Treebank*, *the British National Corpus* o cualquier otro corpus.

Dos medidas de similitud están basadas en la longitud de rutas entre parejas de conceptos, específicamente las medidas propuestas por Leacock–Chodorow (*lch*) y Wu–Palmer (*wup*). La medida propuesta por Leacock–Chodorow encuentra la ruta más corta entre dos conceptos y escala ese valor por la longitud máxima encontrada en la jerarquía de hiperónimos. La medida creada por Wu–Palmer encuentra la profundidad del LCSde un par de conceptos, y luego escala dicho valor teniendo en cuenta la suma de las profundidades de cada nodo. La profundidad de un concepto es simplemente la distancia de dicho nodo al nodo raíz.

WordNet::Similarity soporta el uso de raíces hipotéticas, característica que puede ser habilitada o deshabilitada. Cuando está habilitada, un nodo raíz agrupa la información de todos los conceptos referentes a sustantivos y otro nodo raíz agrupa los conceptos referentes a los verbos. Si se encuentra deshabilitada, los conceptos de sustantivos y verbos se encontrarán en la misma jerarquía.

3.4.2 Medidas de relación semántica

Las medidas de relación semántica son más generales que las anteriores, ya que éstas pueden utilizarse entre palabras que tengan diferente categoría gramatical, de tal manera que no se encuentran limitadas a usar una jerarquía específica. Tres de estas medidas han sido implementadas en *WordNet::similarity*, específicamente las propuestas por Hirst–St–Onge (*hso*), Banerjee–Pedersen (*lesk*) y Patwardhan (*vector*).

La medida propuesta por Hirst–St–Onge clasifica las relaciones en WordNet asignándoles una dirección, y luego establece una relación entre dos conceptos encontrando una ruta que no sea muy larga y que no cambie de dirección frecuentemente. Las otras dos medidas (*lesk* y *vector*) incorporan información de las glosas de WordNet.

La medida propuesta por Banerjee–Pedersen encuentra traslapes de vocablos entre las glosas de dos conceptos y además usa aquellos conceptos con los que se encuentran directamente enlazados a través de las diferentes jerarquías de WordNet. La medida propuesta por Patwardhan, crea una matriz de co-ocurrencia para cada vocablo existente en las glosas de WordNet tomando como referencia cualquier corpus de texto. Luego, representa cada glosa con un vector que es el promedio de los vectores de los vocablos co-ocurrentes.

3.4.3 Uso de WordNet::Similarity

La implementación de la librería *WordNet::Similarity* está basada en algunos módulos desarrollados previamente, los cuales forman parte de las librerías proporcionadas por CPAN (*Comprehensive Perl Archive Network*), tales como el paquete *Text-Similarity* usado para poder encontrar los vocablos comunes en las glosas proporcionadas por WordNet y el paquete *WordNet::QueryData* [63], el cual permite crear un objeto de consulta a las bases de datos textuales de WordNet.

WordNet::Similarity ha sido implementado en Perl siguiendo los lineamientos de programación orientada a objetos. Dicha librería proporciona diversos utilitarios y métodos específicos para el uso y la implementación de nuevas medidas de similitud y relación semántica. La figura 5 muestra un diagrama UML (por sus siglas en inglés *Unified Modeling Language*), en el que se especifica las clases y métodos implementadas en el librería *WordNet::Similarity*.

La utilidad *similarity.pl* permite al usuario obtener un valor que cuantifica la similitud o relación semántica entre parejas de conceptos usando una medida específica. El formato que éste utiliza para especificar un sentido específico de un vocablo ambiguo es *word#pos#sense*; por ejemplo *car#n#3* hace referencia al tercer sentido del sustantivo *car*. También permite especificar todos los sentidos asociados a un vocablo usando el formato *word#pos*. Por ejemplo en la figura 6., el primer comando obtienen el valor de similitud entre el segundo sentido del sustantivo *car* (*railway car*) y el primer sentido del sustantivo *bus* (*motor coach*). El segundo comando obtiene la pareja de sentidos con mayor grado de similitud para los sustantivos *car* y *bus*. En el tercer comando el argumento *allsenses* permite obtener los valores de similitud entre cada uno de los sentidos del sustantivo *car* y el primer sentido del sustantivo *bus*. En los tres comandos anteriores se usó la medida de Lin (*WordNet::Similarity::lin*).

En la figura 7, se crea un objeto de la clase lin y luego encuentra la similitud entre el primer sentido del sustantivo *car* (*automobile*) y el segundo sentido del sustantivo *bus* (*network bus*) usando el método *getRelatedness*.

La librería *WordNet::Similarity* proporciona la capacidad de realizar un seguimiento detallado para los procesos que computan las diferentes medidas de similitud. Por ejemplo, para las medidas basadas en longitudes de ruta, el seguimiento muestra las rutas intermedias entre los conceptos. Para las medidas que se basan en el *contenido de información* permite supervisar las rutas entre conceptos y también la búsqueda del LCS El seguimiento para la medida *hso* muestras las rutas encontradas en WordNet, mientras que para la medida *lesk* muestra el traslape de glosas encontradas entre dos conceptos.

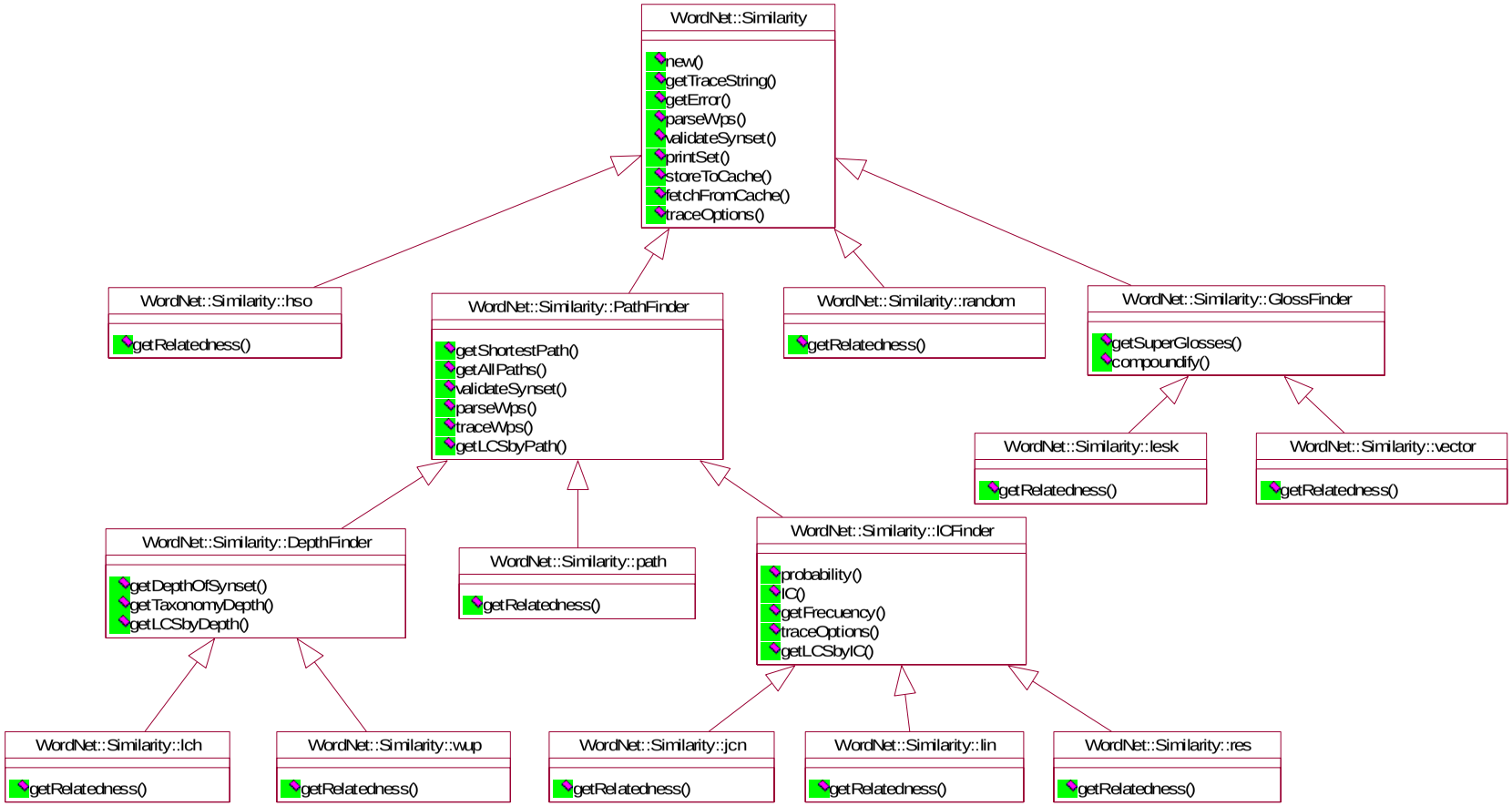


Figura 5. Clases implementadas en la librería WordNet::Similarity

```
> similarity.pl --type WordNet::Similarity::lin car#n#2 bus#n#1
car#n#2  bus#n#1  0.530371390319309        # railway car versus motor coach

> similarity.pl --type WordNet::Similarity::lin car#n bus#n
car#n#1  bus#n#1  0.618486790769613        # automobile versus motor coach

> similarity.pl --type WordNet::Similarity::lin --allsenses car#n bus#n#1
car#n#1  bus#n#1  0.618486790769613        # automobile versus motor coach
car#n#2  bus#n#1  0.530371390319309        # railway car versus motor coach
car#n#3  bus#n#1  0.208796988315133        # cable car versus motor coach
```

Figura 6. Uso de la librería WordNet::Similarity para comparar sentidos

```
#!/usr/bin/perl -w

use WordNet::QueryData;                        # use interface to WordNet
use WordNet::Similarity::lin;                  # use Lin measure

 $wnObj = new WordNet::QueryData;                # create a WordNet object
$linObj = new WordNet::Similarity::lin($wnObj); # create a lin object

$value = $linObj -> getRelatedness ('car#n#1', 'bus#n#2');  # how similar?
```

Figura 7. Comparación de sentidos usando la librería WordNet::Similarity

El módulo *similarity.pm* es la superclase de todos los módulos, y proporciona servicios generales usados por todas las medidas como la validación de identificadores de *synsets*, seguimiento y almacenamiento de los resultados en la memoria de la computadora. Existen cuatro módulos que proporcionan toda la información requerida para cualquiera de las medidas soportadas: *pathfinder.pm*, *ICFinder.pm*, *depthFinder.pm* y *LCSFinder.pm*.

El módulo *pathfinder.pm* proporciona el método *getAllPaths()*, el cual encuentra todas las rutas entre dos *synsets*, y *getShortestPath()* determina la longitud de la ruta más corta entre dos conceptos en cualquiera de las jerarquías proporcionadas por WordNet.

El módulo *ICFinder.pm* proporciona el método *IC ()*, el cual obtiene el valor escalar del *contenido de información* de un *synset*. Los métodos *probability()* y *getfrecuency()* encuentran la probabilidad y la frecuencia de un *synset* basándose en cualquier corpus que haya sido usado para computar el *contenido de información*. Estos valores son calculados previamente, de tal manera que estos métodos son de sólo lectura.

El módulo *depthFinder.pm* proporciona métodos que leen valores previamente calculados por la utilidad *wnDepths.pl*, la cual obtiene la profundidad de un *synset* y su ubicación en la jerarquía de hiperónimos. La profundidad de un *synset* se calcula con el método *getDepthOfSynset()* y la máxima profundidad de una jerarquía se calcula con el método *getTaxonomyDepth()*.

El módulo *LCSFinder.pm* proporciona métodos que encuentran el LCS de dos conceptos usando tres criterios diferentes. Dichos criterios son necesarios desde que existe herencia múltiple de conceptos en WordNet y además diferentes LCS pueden ser seleccionados para una pareja de conceptos, ya que alguno de ellos puede tener múltiples padres en una jerarquía. El método *getLCSbyIC()* escoge el LCS para una pareja de conceptos que tenga el *contenido de información* más alto, *getLCSbyDepth()* selecciona el LCS que tenga la profundidad más alta y *getLCSbyPath()* selecciona el LCS que tenga la ruta más corta.

Capítulo 4
Método propuesto
para desambiguar sentidos de palabras

En esta tesis se presenta un método para desambiguar sentidos de palabras en forma automática. Dicho método se basa en la información sintáctica obtenida del contexto del vocablo ambiguo, el uso de una base de datos de recursos sintácticos previamente compilada, y diversas medidas de similitud de conceptos proporcionadas por la librería *WordNet::Similarity* [63] y WordNet 2.0 [57].

El método propuesto no puede clasificarse como un método supervisado; pese a que usa diferentes fuentes de información, tales como WordNet para la desambiguación y similitud de sentidos de conceptos y SemCor para la creación de una base de datos de recursos sintácticos. Asimismo, no utiliza oraciones etiquetadas semánticamente para realizar algún tipo de entrenamiento previo o construir clasificadores semánticos. Tampoco podría considerársele un método no supervisado ya que no realiza ningún tipo de clasificación o agrupación automática de sentidos usando corpus de gran escala. Este método está basado en el conocimiento, debido a los recursos de información que utiliza, tales como WordNet, SemCor y otros corpus textuales.

En general, este método elige un sentido para un vocablo ambiguo teniendo en cuenta su contexto sintáctico. Dicho contexto a veces es muy pobre, en cuyo caso el método lo enriquece tomando en cuenta ciertas relaciones sintácticas presentes en diferentes niveles del árbol de dependencias. Una vez obtenido dicho contexto se consulta a una base de datos de recursos sintácticos para obtener aquellos términos que son usados en contextos similares al del vocablo ambiguo.

Este conjunto de términos similares, los cuales tienen cierta relación semántica entre ellos, definirán el sentido del vocablo polisémico usando el módulo que etiqueta automáticamente sentidos. Dicho módulo se basa en el algoritmo de McCarthy *et al*. [55][54], el cual se encarga de obtener el sentido más frecuente o predominante de una palabra usando el tesauro de Dekang Lin [51] y las medidas de similitud semántica implementadas por Ted Pedersen *et al*. [61].

A diferencia de McCarthy *et al*., el objetivo de esta tesis es encontrar el sentido que expresa una palabra en un contexto específico y no el más frecuente; por lo tanto no se utiliza el tesauro de Lin, ya que por la manera como éste ha sido construido, proporciona sinónimos para el sentido más predominante de un vocablo ambiguo, que no siempre es el que se expresa en un contexto específico.

Por ejemplo, el sentido más frecuente del término ambiguo *aster* hace referencia a una persona famosa; sin embargo, si dicha ocurrencia aparece en una oración referente a astronomía; su sentido ya no sería el más frecuente.

4.1 Descripción del sistema propuesto

El sistema implementado se encuentra conformado por diferentes módulos construidos de manera independiente, para que de esta manera puedan reutilizarlos en otras tareas concernientes al procesamiento de lenguaje natural. En esta primera parte del capítulo se describirá cada uno de éstos en forma genérica, y en las siguientes secciones se especificará detalladamente sus entradas, la funcionalidad que implementa cada uno, sus salidas y su participación dentro del sistema. La figura 8 muestra un diagrama de bloques que describe los módulos existentes en el sistema y la manera como interactúan.

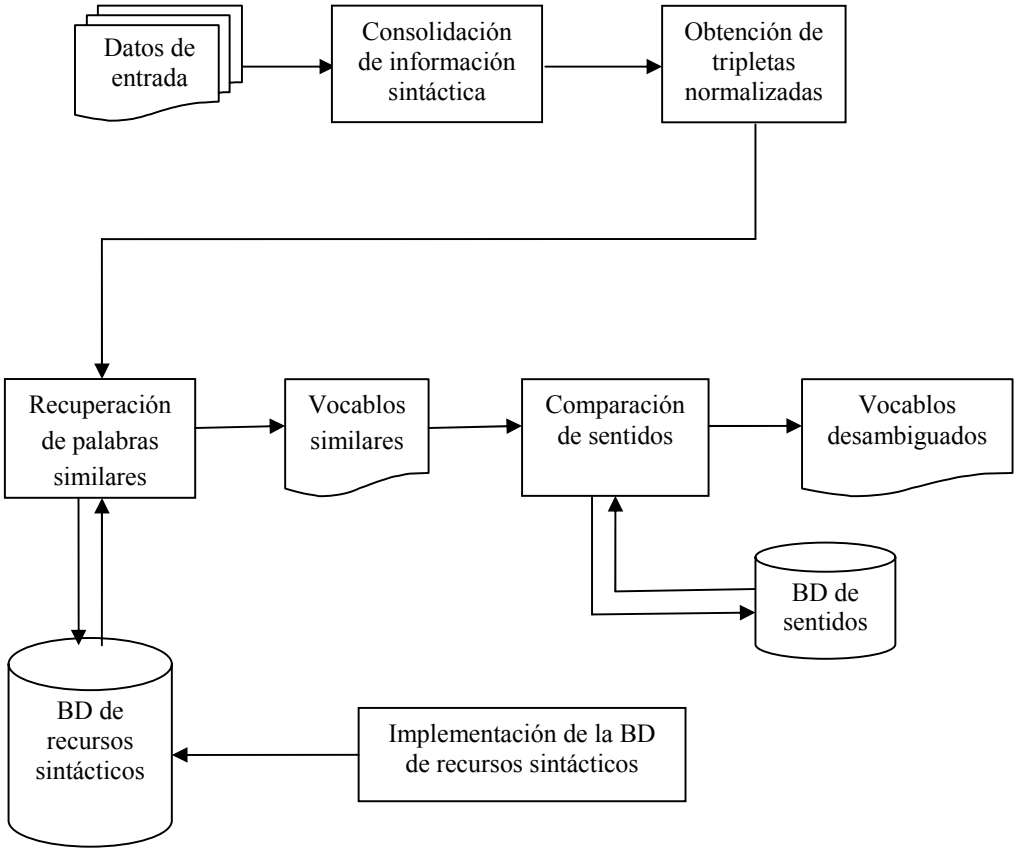


Figura 8. Arquitectura general del sistema

4.1.1 Implementación de la base de datos de recursos sintácticos

El objetivo principal de este módulo es implementar las estructuras de datos necesarias para la creación de una base de datos de recursos sintácticos. La fuente de información necesaria para la creación de este recurso puede ser cualquier corpus de texto en inglés. Este módulo es el encargado de procesar dicha fuente adecuándola al formato de texto requerido por el sistema. Asimismo, debe de analizar sintácticamente las oraciones y obtener las diferentes relaciones de dependencia que puedan existir. Una vez consolidada toda esta información sintáctica, dicho módulo procederá a almacenarla en la base de datos.

4.1.2 Base de datos de recursos sintácticos

Este recurso almacena principalmente las diversas relaciones de dependencia sintáctica que se presentan en un corpus, organizando esta información estadísticamente de acuerdo al modelo de espacio vectorial, el cual es usado generalmente en la clasificación de textos, y en este caso será aplicado a la información extraída de un corpus de texto.

Toda la información estadística existente en la base de datos, es calculada previamente antes que los diferentes módulos empiecen a interactuar; para de esta manera mejorar el rendimiento del sistema. Dicha base de datos también almacena información sintáctica y morfológica de los diferentes vocablos que forman parte de las relaciones de dependencia.

Finalmente, han sido tres las bases de datos creadas, las cuales difieren en el tipo de relaciones de dependencia que almacena cada una de ellas; más no en su estructura de datos. A continuación se enumeran estos tres recursos:

- Base de datos conformada por tuplas basadas en relaciones de dependencia convencional.
- Base de datos conformada por tuplas basadas en relaciones de dependencia sin preposición.
- Base de datos conformada por tuplas basadas en relaciones de dependencia sin preposición, y especial.

4.1.3 Consolidación de información sintáctica

Este módulo es el encargado de analizar y consolidar sintácticamente textos y oraciones que ingresan al sistema, generando árboles de dependencia sintáctica usando las librerías de MINIPAR [50]. Otra de las funciones de este módulo es la de sincronizar la información sintáctica y gramatical proporcionada por SemCor con la suministrada por

MINIPAR. Toda esta información es plasmada en archivos planos usando un formato tabular. Algunos ejemplos se muestran en capítulo 4, específicamente en la tabla 11 y la tabla 12.

4.1.4 Obtención de tripletas normalizadas

Este módulo es el encargado de obtener las tripletas de dependencia sintáctica proporcionadas por el contexto local de algún vocablo ambiguo. Para ello, toma como entrada los árboles de dependencia obtenidos por el módulo anterior. Otra de las funciones de este módulo es enriquecer el contexto sintáctico cuando éste es muy pobre. Por ejemplo, en la expresión *the dog was in the park*, el conjunto de los modificadores sintácticos del sustantivo *dog* está conformado por el término *the*. Este contexto es considerado muy pobre ya que dicho vocablo puede ser usado como modificador de cualquier sustantivo. La manera de enriquecer el contexto es usar modificadores presentes en los diferentes niveles del árbol sintáctico que mantengan alguna relación de dependencia con el vocablo ambiguo.

4.1.5 Recuperación de palabras similares

Este módulo es el encargado de obtener un conjunto de palabras similares al vocablo ambiguo teniendo en cuenta su contexto sintáctico obtenido por el módulo anterior. Dicho conjunto se encuentra ordenado por un peso de similitud con respecto al término polisémico.

El motor de recuperación de este módulo se encuentra basado en el modelo de espacio vectorial o el esquema TF-IDF (por sus siglas en inglés *term frecuency–inverse document frecuency*). Estos valores se calculan para cada tupla sintáctica almacenada en la base de datos, de tal manera que cuando ésta es consultada, su contenido se organiza en una estructura de datos que soporta vectores de *n* dimensiones. El término ambiguo se organiza como un vector donde cada uno de sus modificadores es una dimensión. Al compararlo con cada uno de los vectores del recurso se obtienen sus vocablos similares. Los vectores con los cuales se compara el vector del término polisémico son discriminados teniendo en cuenta la categoría gramatical de éste. La tabla 9 muestra los términos similares al vocablo *doctor*, específicamente a su primer sentido, el cual hace referencia al sentido médico. Tanto los vocablos, como los pesos de similitud han sido tomados exactamente como los proporciona este módulo. Los vocablos se representan mediante la notación *vocablo#categoría gramatical*.

4.1.6 Etiquetado automático de sentidos

El objetivo de este módulo es obtener un valor que refleje la similitud o relación semántica entre cada uno de los sentidos del vocablo ambiguo y los términos similares obtenidos por el módulo de recuperación de palabras similares. El algoritmo de comparación utilizado se encuentra basado en el modelo propuesto por McCarthy *et al*. Asimismo, las medidas implementadas en este módulo son las propuestas por Jiang–Conrath [34], la medida de Lin [49]; y una medida de relación semántica que se basa en el algoritmo de Lesk creada por Banerjee–Pedersen, más conocida como medida de Lesk adaptada. Se escogieron estas tres medidas porque fueron las que mejor resultado obtuvieron en el análisis realizado por Pedersen [61].

Tabla 9. Términos similares al vocablo "doctor"

Vocablo similar	**Peso de similitud**
calorimeter#n	0.745
extern#n	0.577
chemistry_laboratory#n	0.577
comer#n	0.577
corkscrew#n	0.577
dead_end#n	0.577
diabetic#n	0.577
harshness#n	0.577
mutual_understanding#n	0.577
psychic_phenomenon#n	0.577
sedative#n	0.577
variable_resistor#n	0.577
great_deal#n	0.504
good_day#n	0.497
bind#n	0.477
body_of_water#n	0.477
conscientious_objector#n	0.477
determining_factor#n	0.477
dud#n	0.477
fake#n	0.477
fingerprint_specialist#n	0.477

Es necesario mencionar que los sentidos de los vocablos ambiguos son definidos por WordNet y las medidas de similitud han sido implementadas usando la librería *WordNet::Similariy*.

4.1.7 Base de datos de sentidos

Esta base de datos fue creada para almacenar valores de similitud entre un par de sentidos. Dichos valores se calculan usando las tres medidas especificadas en el párrafo anterior. La finalidad de este recurso es mejorar el rendimiento del módulo encargado de etiquetar de sentidos, ya que resulta menos costoso consultar a una base de datos por un valor específico, que procesar todo el algoritmo correspondiente a la medida de similitud elegida, realizando diversas consultas y cálculos sobre las bases textuales de WordNet. Asimismo, otro de los objetivos de este recurso, es que pueda ser reutilizado por otras aplicaciones que requieran valores de similitud semántica entre dos glosas definidas en WordNet, tomando como referencia alguna de las medidas de similitud implementadas.

4.2 Implementación de la base de datos de recursos sintácticos

La base de datos de recursos sintácticos juega un rol importante en el método de desambiguación propuesto. El objetivo principal de este recurso es almacenar pares de palabras que se encuentren bajo cierta relación de dependencia sintáctica tomando como referencia el corpus de texto usado para su entrenamiento, que en este caso es SemCor. Esta base de datos también almacena los pesos calculados para cada par de vocablos siguiendo el modelo del espacio vectorial.

La creación de dicho recurso toma en cuenta la información lingüística proporcionada por el analizador sintáctico (categorías gramaticales y tipo de relaciones sintácticas), estadística proporcionada por el modelo de espacio vectorial y conocimiento técnico sobre bases de datos (índices, procedimientos almacenados, disparadores y vistas) para obtener el mayor rendimiento posible. Este recurso es utilizado explícitamente por el módulo de recuperación de vocablos similares al término ambiguo tomando en cuenta su contexto local.

Es necesario comentar que este módulo obtiene como producto final tres bases de datos, cada una de la cuales está construida siguiendo criterios lingüísticos diferentes aplicados a la obtención de las tripletas de dependencia sintáctica. Las bases de datos han sido implementadas en SQL Server 2000 y cada una de ellas contiene aproximadamente 50,000 tuplas que agrupan casi medio millón de registros. La figura 9 muestra un diagrama de bloques que describe los procesos seguidos en la implementación de dicho recurso.

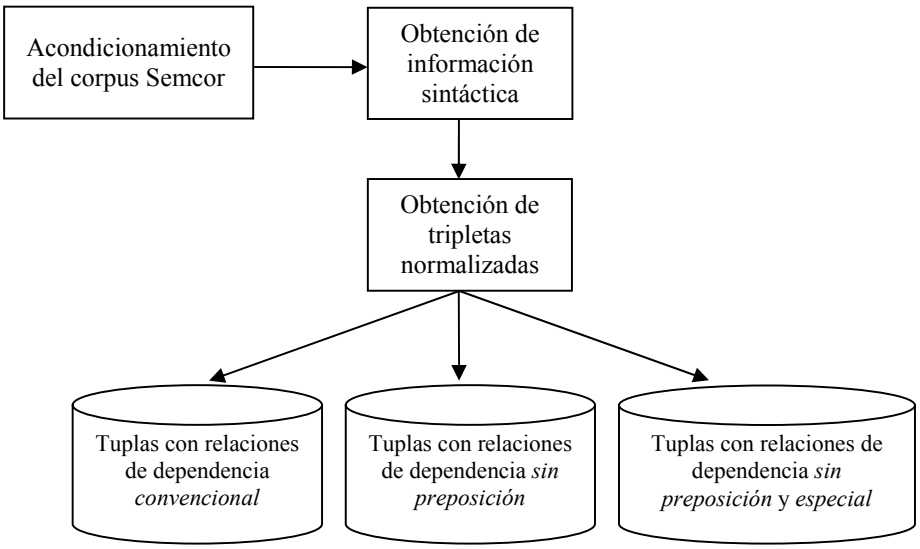


Figura 9. Implementación de la base de datos de recursos sintácticos

4.2.1 Acondicionamiento del corpus SemCor

Como se describió en el capítulo 3, SemCor agrupa un conjunto de archivos, cada uno de los cuales contiene diferentes oraciones representadas mediante un lenguaje de etiquetas. Para cada vocablo se proporciona su información gramatical y su número de sentido tomando como referencia WordNet. Dicho número se denomina *wnsn* por sus siglas en inglés (*WordNet sense number*), el cuál está acotado por etiquetas en el formato de SemCor.

Este módulo proporciona la funcionalidad necesaria para transformar cada uno de los archivos de SemCor al formato requerido por el analizador sintáctico. Si bien es cierto que dicho analizador proporciona información gramatical de cada uno de los vocablos del corpus, SemCor también la suministra; por ende otra de las tareas de este proceso es maximizar su uso y almacenarla en la base de datos, ya que el método de desambiguación propuesto utiliza ambas fuentes de información para poder discriminar gramaticalmente los vectores del recurso contra los que se comparará el vector del vocablo ambiguo.

Este módulo genera dos archivos de salida por cada archivo de entrada, donde cada uno de éstos agrupa proposiciones *normales* o *especiales*. A cada vocablo de la proposición *especial*, se le adjunta un par de características proporcionadas por SemCor, tales como su categoría gramatical y su *wnsn*, el cual podría ser utilizado como atributo de los vectores para diferenciar los contextos sintácticos para cada sentido del vocablo ambiguo, tarea que

es más apropiada de un algoritmo de desambiguación supervisada. A continuación se muestra una oración de SemCor representada como proposición *normal* y *especial*.

– Proposición *normal*: *When the crowd was asked whether it wanted to wait one more term to make the race, it voted no and there were no dissents.*

– Proposición *especial*: *When(?,WRB) the(?,DT) crowd(1,NN) was(?,VBD) asked(1,VB) whether(?,IN) it(?,PRP) wanted(1,VB) to(?,VB) wait(2,VB) one(1,JJ) more(3,JJ) term(2,NN) to(?,TO) make(5,VB) the(?,DT) race(1,NN) COMMA(?,?) it(?,PRP) voted(3,VB) no(1,RB) and(?,CC) there(?,EX) were(5,VB) no(?,DT) dissents(2,NN) PUNTO(?,?)*

4.2.2 Obtención de información sintáctica

El principal objetivo de este módulo es aprovechar la funcionalidad proporcionada por el analizador sintáctico, examinando las oraciones con formato normal, para de esta manera generar un árbol de dependencia sintáctica para cada una de ellas. Es necesario aclarar que MINIPAR comprende un conjunto de librerías codificadas y compiladas en Visual C++, las cuales han sido referenciadas por este módulo para implementar un programa que genere dicho árbol. El formato tabular usado para expresar tales resultados se muestra en la tabla 10.

Tabla 10. Formato usado por el árbol de dependencias sintácticas

Columna	**Parámetro**
Columna 1	Número de palabra
Columna 2	Vocablo dependiente
Columna 3	Lema y categoría gramatical
Columna 4	Número de vocablo que gobierna la relación
Columna 5	Relación gramatical
Columna 6	Vocablo que gobierna en la relación de dependencia

El analizador solicita como punto de entrada cualquier archivo de texto plano, cuyas oraciones se encuentren delimitadas por un punto, de tal manera que éste proporcionará como salida otro archivo de texto conformado por los árboles de dependencia de cada oración. La tabla 11 muestra el árbol de dependencias generado al procesar la oración: *The Fulton County Grand Jury said Friday an investigation of Atlanta's recent primary election produced no evidence that any irregularities took place*. Es necesario aclarar que el símbolo "~" es usado en la columna 3, cuando el lema es igual al vocablo dependiente.

Este módulo también tiene implementada la funcionalidad de combinar la información de las proposiciones *normales* y *especiales*. Para ello, anexa al árbol de dependencias obtenido de una proposición *normal*, la información gramatical proveniente de SemCor. Para la oración especificada en el ejemploa anterior sería: *The(?,DT) Fulton_County_Grand_Jury(1,NNP) said(1,VB) Friday(1,NN) an(?,DT) investigation(1,NN) of(?,IN) Atlanta(1,NN) 's(?,POS) recent(2,JJ) primary_election(1,NN) produced(4,VB) no(?,DT) evidence(1,NN) that(?,IN) any(?,DT) irregularities(1,NN) took_place(1,VB) PUNTO(?,?)*. La tabla 12 muestra el formato final del árbol después de haberle anexado la información proveniente de ambas proposiciones.

Es necesario mencionar que el uso de ambas proposiciones no es obligatorio; sin embargo, al haber utilizado el corpus SemCor como fuente de entrenamiento creeemos que es importante aprovechar la información gramatical existente en dicho corpus. Si se hubiera utilizado cualquier otro corpus que no presente dicha información, las proposiciones *especiales* no serían tomadas en cuenta.

Este proceso, además de ser utilizado en la implementación de la base de datos de recursos sintácticos, también es usado en la arquitectura principal del sistema para obtener los árboles de dependencia del contexto del vocablo ambiguo.

Tabla 11. Árbol de dependencias sintácticas

Nº	**Vocablo dependiente**	**Lema, POS**	**Nº**	**Relación**	**Vocablo que gobierna la relación**
1	*The*	~, Det	2	*Det*	*Fulton_County_Grand_Jury*
2	*Fulton_County_Grand_Jury*	~, N	3	*S*	*say*
3	*said*	say, V	E0	*I*	*fin*
4	*Friday*	_DATE, N	3	*Guest*	*say*
5	*an*	~, Det	6	*Det*	*investigation*
6	*investigation*	~, N	12	*S*	*produce*
7	*of*	~, Prep	6	*Mod*	*investigation*
8	*Atlanta*	~, N	11	*Gen*	*primary_election*
9	*'s*	~, U	8	*Poss*	*Atlanta*
10	*recent*	~, A	11	*Mod*	*primary_election*
11	*primary_election*	~, N	7	*pcomp-n*	*of*
12	*produced*	produce, V	3	*Sc*	*say*
15	*no*	~, PostDet	16	*Post*	*evidence*
16	*evidence*	~, N	12	*Obj*	*produce*
19	*that*	~, N	E2		
20	*any*	~, Det	22	*Det*	*took_place*
21	*irregularities*	irregularity, N	22	*Nn*	*took_place*
22	*took_place*	~, N	E2		

4.3 Obtención de tripletas normalizadas

El objetivo principal de este módulo es obtener tripletas de dependencia sintáctica procesando el contexto del vocablo ambiguo; por ende recibe como entrada los árboles de dependencia obtenidos por el módulo encargado de obtener la información sintáctica. Este proceso no sólo interviene en la arquitectura principal del sistema para la obtención del contexto sintáctico del vocablo ambiguo; sino también en el proceso que implementa la base de datos de recursos sintácticos.

La extracción de tripletas no es un proceso mecánico, ya que muchas veces el conjunto de modificadores sintácticos de un vocablo no posee el *contenido de información* necesaria como para poder identificarlo a través de éstos. Por ende, uno de los grandes problemas a la hora de obtener información sintáctica del contexto local de un vocablo es la pobreza de las relaciones sintácticas. Esto se debe a la ineficiencia del analizador sintáctico empleado, o al uso de relaciones de dependencia tradicionales, que sólo toman en cuenta las flechas de dependencia que salen del vocablo; más no las que llegan a él.

Este módulo pretende proporcionar términos coherentes fuertemente relacionados con el vocablo en cuestión. Para ello, se han utilizado diversos criterios lingüísticos que enriquecen el contexto local de una palabra, teniendo en cuenta diferentes tipos de relaciones de dependencia presentes en el árbol sintáctico de la oración, las cuales han sido creadas con el objetivo de proporcionar mayor cantidad de términos relacionados a un vocablo ambiguo, para que el proceso de desambiguación sea más confiable.

4.3.1 Relaciones de dependencia

Una relación de dependencia es una correspondencia binaria asimétrica entre un vocablo, al que en adelante se le denominará *cabeza* (vocablo que gobierna la relación) y otro llamada modificador (vocablo dependiente de la *cabeza*). Las gramáticas de dependencia representan las oraciones como un conjunto de relaciones, las cuales forman un árbol que conectan todas las palabras en una oración. Un vocablo en una oración puede tener diferentes modificadores; sin embargo, sólo puede modificar a un término a la vez. Además de ello, la raíz del árbol de dependencia no modifica a ninguna otra palabra y puede ser llamada la *cabeza* de la oración [49].

Por ejemplo, en la figura 10 se muestra la estructura de dependencia de una oración, en la cual es posible que varias flechas salgan de una misma palabra con dirección a sus modificadores; sin embargo, sólo una flecha puede llegar a un vocablo, la cual hace referencia al término al que modifica. También se puede apreciar que al término *chased* no le llega ninguna flecha, ya que éste es la *cabeza* de la oración.

Tabla 12. Árbol de dependencia con información gramatical de SemCor

Nº	**Vocablo dependiente**	**Lema, POS**	**Nº**	**Rela-ción**	**Vocablo que gobierna la relación**	**Wnsn**	**POS Sem Cor**
1	*The*	~, Det	2	Det	*Fulton_County_ Grand_Jury*	?	DT
2	*Fulton_County _Grand_Jury*	~, N	3	S	*say*	1	NNP
3	*said*	say, V	E0	I	*fin*	1	VB
4	*Friday*	_DATE, N	3	Guest	*say*	1	NN
5	*an*	~, Det	6	Det	*investigation*	?	DT
6	*investigation*	~, N	12	S	*produce*	1	NN
7	*of*	~, Prep	6	Mod	*investigation*	?	IN
8	*Atlanta*	~, N	11	Gen	*primary_election*	1	NN
9	*'s*	~, U	8	Poss	*Atlanta*	?	POS
10	*recent*	~, A	11	Mod	*primary_election*	2	JJ
11	*primary_electi on*	~, N	7	Pcomp -n	*of*	1	NN
12	*produced*	produce, V	3	Sc	*say*	4	VB
15	*no*	~, PostDet	16	Post	*evidence*	?	DT
16	*evidence*	~, N	12	Obj	*produce*	1	NN
19	*that*	~, N	E2			?	IN
20	*any*	~, Det	22	Det	*took_place*	?	DT
21	*irregularities*	Irregulari ty, N	22	Nn	*took_place*	1	NN
22	*took_place*	~, N	E2			1	VB

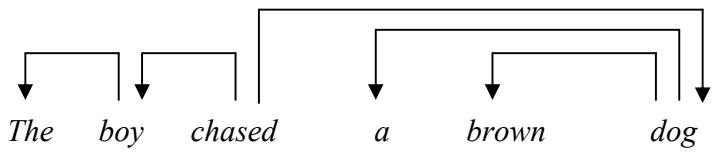


Figura 10. Dependencia sintáctica tradicional

Para poder representar un árbol de dependencias sintácticas se usará una lista de tuplas, donde cada una de ellas representa un nodo (vocablo) en el árbol de dependencias. El formato utilizado es el siguiente:

(palabra, *categoría*, *cabeza*, *relación*, *categoría de SemCor*, *wnsn)*

donde:

palabra es el vocablo representado por el nodo,
categoría es la categoría gramatical del vocablo según el analizador sintáctico,
cabeza especifica el vocablo al que modifica,
relación es una etiqueta que expresa la relación de dependencia,
categoría de SemCor es la categoría gramatical del vocablo según SemCor y,
wnsn es el número de sentido del vocablo según WordNet 2.0.

Es necesario aclarar que los dos últimos integrantes de la tupla (*categoría de SemCor* y *wnsn*), sólo se utilizan porque el recurso de texto empleado fue SemCor, el cual proporciona ésta información. Sin embargo, si se fuera a usar otro corpus de texto, que no proporcionase dicha información, estos dos últimos parámetros podrían excluirse de la tupla. La tabla 13 muestra las tuplas de dependencia obtenidas del ejemplo anterior.

Tabla 13. Tripletas de dependencia sintáctica

Palabra	**Categoría**	***Cabeza***	**Relación**
The	N	*boy*	Spec.
boy	V	*chased*	Subj.
A	Prep	*dog*	Spec.
brown	N	*dog*	Adjn.

La tarea que realiza este módulo no sólo es extraer las diferentes tripletas de dependencia, sino que también obtiene las categorías gramaticales proporcionadas por SemCor y el analizador sintáctico, así como el número de sentido de cada término de la oración. La extracción de las diversas tripletas ha sido realizada basándose en ciertos criterios lingüísticos, los cuales se describen en esta sección. Cada uno de éstos, usa como fuente de información el árbol de dependencia obtenido por el módulo encargado de obtener información sintáctica.

Finalmente, cada nodo del árbol se almacena en la base de datos de recursos sintácticos que contiene las estructuras de información necesarias para representar dichas tuplas sintácticas. De esta manera, una tupla sintáctica podría ser representada por la siguiente notación:

$$Tupla\ (cabeza) = \{mod\_1, mod\_2, mod\_3, ..., mod\_n\}$$

Tomando en cuenta la oración anterior, las siguientes expresiones se almacenarían en la base de datos:

Tupla (*boy)* = {*The*}
Tupla (*chased*) = {*boy*}
Tupla (*dog*) = {*a, brown*}

4.3.2 Tipo de relaciones de dependencia

Algunos trabajos anteriores han usado la información sintáctica existente en el contexto local del vocablo ambiguo como fuente de información principal para su desambiguación. Sin embargo, los criterios usados para la obtención de las tripletas sintácticas se han basado en relaciones de dependencia tradicionales [50].

En este trabajo se han realizado varias pruebas usando relaciones de de dependencia *convencional*, llegando a la conclusión que éstas no aportan el contexto sintáctico suficiente para una desambiguación exitosa. Es por ello, que se han creado ciertas variantes de éstas con la finalidad de incrementar la calidad y cantidad en cuanto a los términos con los que el vocablo ambiguo mantiene algún tipo de relación semántica.

a. Relación de dependencia sintáctica convencional

Una relación de dependencia sintáctica *convencional* es aquella en la que una pareja de vocablos mantiene una relación de dependencia tradicional especificada por el árbol de dependencias sintácticas. Más explícitamente, las flechas que salen de un vocablo hacia otros se consideran los modificadores sintácticos del primero. Por ejemplo, en la oración *he likes the woman very much*, las tuplas que mantienen esta relación de dependencia se muestran a continuación:

Tupla (*like*) = {*he, woman, very much*}
Tupla (*woman*) = {*the*}

El problema de este tipo de relación, es que muchos de los modificadores encontrados no aportan significado o semántica en cuanto a la relación de dos conceptos; por ejemplo en el segundo caso el determinante *the* puede ser modificador de cualquier sustantivo, en cambio en el primer caso la pareja (*like, woman*), que al igual que (*woman, the*) se encuentran en una relación de dependencia sintáctica, ofrece mayor *contenido de información*.

b. Relación de dependencia sintáctica sin preposición

Una relación de este tipo es aquella que excluye la preposición cuando ésta se presenta como modificador de alguna *cabeza*, ya que no aporta el *contenido de información*

necesario como para asociar dos vocablos que tienen cierto grado de relación semántica, de tal manera que el término que depende de ésta pasa a ser el modificador de dicha *cabeza*.

Por ejemplo, cuando el analizador sintáctico procesa específicamente la parte resaltada en negrita de la proposición: *The jury further said in term end presentments that the City Executive Committee, which had over all* ***charge of the election****...*, obtiene las relaciones mostradas en la tabla 14.

Tabla 14. Ejemplo de dependencia sintáctica sin preposición

Nº	**Vocablo dependiente**	**Lema, POS**	**Nº**	**Relación**	**Vocablo que gobierna la relación**
18	*charge*	~, V	E1	*i*	*fin*
19	*of*	~, Prep	18	*mod*	*charge*
20	*the*	~, Det	21	*det*	*election*
21	*election*	~, N	19	*pcomp-n*	*of*

Si se obtuvieran las tuplas de dependencia sintáctica basadas en relaciones *convencionales*, tendríamos tres tuplas, las cuales se representan usando la notación especificada anteriormente y también pueden ser apreciadas en el árbol de dependencias mostrado en la figura 11.

Tupla (*charge*) = {*of*}
Tupla (*of*) = {*election*}
Tupla (*election*) = {*the*}

Como se puede observar en las tuplas obtenidas, la pareja *(charge, of)*, al igual que el caso del determinante y el sustantivo *(election, the)*, no aporta suficiente *contenido de información* como para poder encontrar semejanzas o diferencias semánticas con otras parejas. En cambio, si se hubiera obtenido las tuplas basándose en relaciones de dependencia sintáctica *sin preposición*, habríamos obtenido la pareja (*charge, election*), la cual obviamente presenta mayor relación semántica que la anterior pareja *(charge, of)*.

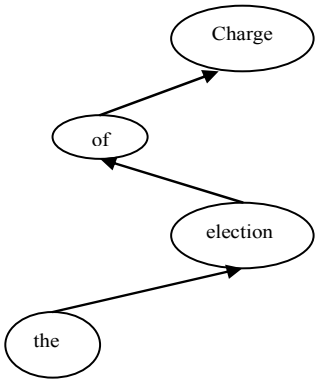


Figura 11. Dependencia sintáctica con preposición

El bajo *contenido de información* de una tupla cuyo modificador es una preposición es más notorio en los verbos, ya que muy frecuentemente éstos son modificados por frases preposicionales y si además se tiene en cuenta que los verbos derivan mayor información para su desambiguación de sus objetos que de sus sujetos, se concluye que es necesario eliminar las preposiciones de las tuplas sintácticas [83]. Finalmente, las tuplas obtenidas basándose en relaciones de dependencia sintáctica *sin preposición* se especifican a continuación, al igual que el árbol de dependencia resultante en la figura 12.

Tupla (charge) = {*election*}
Tupla (election) = {*the*}

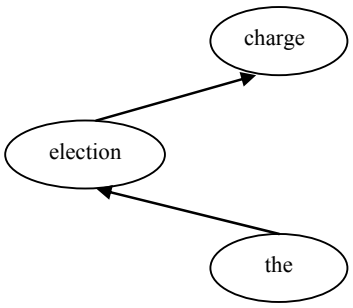


Figura 12. Dependencia sintáctica sin preposición

c. Relación de dependencia sintáctica especial

Pese al uso de las dos relaciones especificadas anteriormente, muchas veces el contexto sintáctico local sigue siendo pobre; es por ello que se ha creado esta nueva relación de dependencia. Una relación de dependencia sintáctica *especial* incluye como parte del conjunto de los modificadores de la *cabeza*, el vocablo al que modifica la *cabeza*, incrementando el número de términos relacionados semánticamente con el vocablo ambiguo.

Por ejemplo, cuando el analizador sintáctico procesa específicamente la parte resaltada en negrita de la proposición *it urged that the city take steps to* ***remedy this problem***, obtiene las relaciones de dependencia especial que se muestran en la tabla 15. Si se obtuviera las tuplas sintácticas tomando en cuenta las relaciones de dependencia *convencional*, se hubiese obtenido la tuplas que se muestran a continuación:

Tupla (*remedy*) = {*problem*}
Tupla (*problem*) = {*this*}

En cambio, si se hubiera obtenido las tuplas sintácticas tomando en cuenta las relaciones de dependencia sintáctica *especial*, entonces la tuplas obtenidas hubieran sido las siguientes:

Tupla (*remedy*) = {*problem*}
Tupla (*problem*) = {*this, remedy*}

Tabla 15. Ejemplo de dependencia sintáctica especial

Nº	**Vocablo dependiente**	**Lema, POS**	**Nº**	**Relación**	**Vocablo que gobierna la relación**
11	*remedy*	~, V	E0	*i*	*fin*
14	*this*	~, Det	15	*det*	*problem*
15	*problem*	~, N	11	*obj*	*remedy*

En cambio, si se hubiera obtenido las tuplas sintácticas tomando en cuenta las relaciones de dependencia sintáctica *especial*, entonces la tuplas obtenidas hubieran sido las siguientes:

Tupla (*remedy*) = {*problem*}
Tupla (*problem*) = {*this, remedy*}

Como se puede observar, el conjunto de modificadores de la *cabeza problem* presentan mayor relación semántica con éste cuando se usan las relaciones de dependencia *especial*, ya que en el primer caso el vocablo *remedy* es modificado por *this*, lo cual no aporta nada importante en cuanto a semántica, ya que *this* podría modificar a cualquier sustantivo, en cambio en el segundo caso, el término *remedy* forma parte de los modificadores de *problem*, lo cual hace que esta tupla tenga mayor *contenido de información*.

4.3.3 Aspectos generales para la obtención de tripletas

Esta sección describe algunos aspectos generales que es necesario tomar en cuenta al momento de obtener tripletas sintácticas, de tal manera que éstas tengan la mayor calidad posible en lo que a *contenido de información* se refiere. En cualquiera de los tres tipos de relaciones descritos anteriormente cabe la posibilidad de encontrar *cabeza*s léxicamente iguales; pero con diferente categoría gramatical. Dichas *cabeza*s son consideradas heterogéneas en este recurso, ya que los contextos de dos vocablos con estas características suelen ser diferentes. La información sintáctica proporcionada por SemCor y el analizador sintáctico es relevante para encontrar este tipo de vocablos. Por ejemplo, en las siguientes oraciones es posible apreciar el vocablo *dictate*, el cual asume diferentes categorías gramaticales en cada oración.

*I **dictate** a letter*
Tupla (*dictate*) = {*I, letter*}

*I followed the **dictates** of my conscience*
Tupla (*dictate*) = {*the, conscience*}

Dicho vocablo puede ser verbo o sustantivo, de tal manera que ambas instancias de *dictate* serán tomadas como cabezas diferentes. De esta manera, después de haber definido las diferentes relaciones de dependencia tomadas en cuenta en la construcción de la base de datos y también en el análisis del contexto del vocablo ambiguo, se enumera las tres bases de datos implementadas*:*

- Base de datos conformada por tuplas basadas en relaciones de dependencia convencional.
- Base de datos conformada por tuplas basadas en relaciones de dependencia sin preposición.
- Base de datos conformada por tuplas basadas en relaciones de dependencia sin preposición y especial.

Pese a los diferentes tipos de relaciones que se han tomado en cuenta para enriquecer el contexto local de un vocablo ambiguo, aún existen muchas términos cuyos modificadores carecen de relación semántica suficiente para que ésta pueda ser desambiguada exitosamente. Éste es un problema que aún no ha sido resuelto por la comunidad científica para el procesamiento de lenguaje natural; sin embargo, existen métodos alternativos como la inclusión de palabras co-ocurrentes o aquellas que tienen mayor relevancia en todo el texto, incluso algunos usan colocaciones gramaticales.

Definitivamente es necesario enriquecer el contexto no sólo en volumen sino en calidad; ya que como afirma Yarowsky, sólo es necesario 3 ó 4 términos para resolver una ambigüedad local; por ende es necesario determinar el impacto de cada una de las técnicas alternativas mencionadas en el proceso de desambiguación de sentidos de palabras [83][84][85].

4.4 Recuperación de palabras similares

El objetivo principal de este módulo, es obtener términos que presenten cierta similitud semántica con el vocablo ambiguo, para lo cual se toma en cuenta su contexto y todas aquellas tuplas almacenadas en la base de datos. En esta sección se describirá la manera en que se organizan dichas tuplas en el recurso sintáctico; así como el motor de

recuperación de información empleado. Los vocablos similares recuperados en adelante serán denominados *vecinos* del término ambiguo.

El motor de recuperación de palabras similares se encuentra basado en el modelo de espacio vectorial, también conocido como el esquema TF-IDF (por sus siglas en inglés *term frecuency-inverse document frecuency*), el cual generalmente es aplicado a tareas de clasificación y similitud de documentos, representando cada documento por un vector y comparándolos usando esquemas multidimensionales; sin embargo, para esta tarea un documento es equivalente a una de las tuplas existentes en el recurso, cada una de las cuales son definidas con la notación que se muestra a continuación:

$$\text{Tupla}(i) = \{(\text{mod}_1\ \text{frec}_{1,}), (\text{mod}_2\ \text{frec}_{2,}), ..., (\text{mod}_n\ \text{frec}_{n,})\}$$

donde:

mod_n es el nombre del modificador y,
frec_n es la frecuencia del mod_n con la cabeza_n.

Como parte de este modelo, es necesario calcular la frecuencia normalizada de un término (en nuestro trabajo es un modificador) y la frecuencia inversa del mismo. Este último hace referencia al número de *cabeza*s relacionadas con dicho modificador. Finalmente, el peso del vocablo *w* refleja la importancia de la relación entre dicho modificador y una *cabeza* específica. La frecuencia normalizada de un modificador se calcula con la ecuación 6.

$$f_{i,j} = \frac{\text{frec}_{i,j}}{\max \text{frec}_{l,j}} \tag{6}$$

donde:

$\text{frec}_{i,j}$ es la frecuencia del modificador *i* con la cabeza_j y,
$\max \text{frec}_{l,j}$ es la máxima frecuencia de los modificadores de la cabeza_j.

La frecuencia inversa de un modificador puede ser calculada con la ecuación 7.

$$\text{idf}_i = \log \frac{N}{n_i} \tag{7}$$

donde:

N es el número total de *cabeza*s en la base de datos y,

n_i es el número de *cabeza*s con las cuales se relaciona el modificador *i*.

Finalmente, el peso de la relación entre un modificador y una *cabeza* se calcula con la ecuación 8.

$$w_{i,j} = f_{i,j} \times idf_i \tag{8}$$

donde:

$f_{i,j}$ es la frecuencia normalizada del modificador *i* con una $cabeza_j$ y,
idf_i es la frecuencia inversa del modificador *i*.

El TF muestra la importancia de un modificador respecto a la *cabeza* que modifica, de tal manera, que el peso de la relación aumenta si el modificador aparece más a menudo con dicha *cabeza*, sucediendo lo contrario con el IDF, que denota la importancia de un modificador respecto al resto de *cabeza*s del recurso, de tal manera que el peso de un modificador disminuye si aparece más a menudo con todas las demás *cabeza*s del recurso, y aumenta cuando aparece con la menor cantidad de *cabeza*s posibles, ya que los modificadores muy frecuentes discriminan poco a la hora de representar la *cabeza* mediante un vector.

Después de haber calculado las fórmulas mencionadas, cada *cabeza* es representada en el sistema de recuperación de información implementado por un conjunto de tuplas cuaternarias, las cuales son definidas con la siguiente notación:

$$\text{Tupla}(i) = \{(mod_1, tf_1, idf_1, w_1), (mod_2, tf_2, idf_2, w_2), ..., (mod_n, tf_n, idf_n, w_n)\}$$

Tal como se puede apreciar en la arquitectura principal del sistema, este módulo recibe como entrada una tupla, que se encuentra en función del vocablo que se desea desambiguar. Ésta es representada por un vector *q*, y cada tupla del recurso es representada por un vector v_j sucesivamente, de tal manera que es muy fácil intuir que el número de dimensiones del vector v_j, es mucho mayor al número de dimensiones del vector *q*, ya que en varios casos las dimensiones de *q* oscilan entre 1 y 5 dimensiones como máximo, lo cual depende del número de modificadores de éste, mientras que las dimensiones de los vectores del recurso varían entre 1 y 13,500; lo cual depende del tamaño del corpus de texto usado. El módulo de recuperación de palabras similares se encarga de comparar el vector *q* con los aproximadamente 50,000 vectores v_j que contiene el recurso, siendo necesario para ello igualar previamente las dimensiones de ambos vectores rellenando de ceros aquellas dimensiones que no tengan valor para cualquiera de los dos vectores a comparar; construyendo de esta manera un sistema de vectores de *n* dimensiones.

El modelo de vectores propuesto para evaluar el grado de similitud semántica entre los vectores *q* y v_j puede ser cuantificado por el coseno del ángulo que forman, tal como se muestra en la ecuación 9.

$$\text{Valor} = \frac{\vec{v}_j \bullet \vec{q}}{\left|\vec{v}_j\right| \times \left|\vec{q}\right|}$$

$$\text{Valor} = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} (w_{i,j})^2} \times \sqrt{\sum_{i=1}^{t} (w_{i,q})^2}} \qquad (9)$$

De esta manera, al comparar *q* con cada una de los vectores de nuestro recurso se obtendrán los vecinos del vocablo ambiguo. El valor de comparación entre éstos oscila entre 0 y 1, de tal manera que el módulo encargado de recuperar los vocablos similares presenta como parámetro configurable un *umbral de comparación*, el cual permite seleccionar los vectores que superen dicho valor.

El sistema que implementa este módulo puede ser configurado para que tenga en cuenta no sólo la categoría gramatical de la *cabeza* ambigua, sino también de las 50,000 *cabeza*s que conforman el recurso. Por ejemplo, puede ser posible que se desee desambiguar una *cabeza* cuya categoría gramatical sea sustantivo y compararla con *cabeza*s del recurso que tengan la misma categoría gramatical o en su defecto con todas las *cabeza*s sin importar dicha característica.

4.5 Etiquetado automático de sentidos

El objetivo de este módulo es etiquetar semánticamente un vocablo ambiguo, para lo cual toma como entrada dicho vocablo y un conjunto de vecinos o palabras similares, cada uno con su categoría gramatical y un peso de similitud establecido por el módulo encargado de la recuperación de palabras similares. La categoría gramatical es necesaria, ya que ciertas medidas de similitud usan jerarquías específicas de WordNet, las cuales tienen en cuenta esta característica. El peso refleja una aproximación o relación semántica entre el término ambiguo y cada uno de sus vecinos.

Este módulo presenta dos medidas de similitud semántica: la planteada por Jiang–Conrath [34] y la propuesta por Lin [49]. Además de éstas, proporciona una medida de relación semántica que se basa en el algoritmo de Lesk creada por Banerjee–Pedersen [6], más conocida como medida de Lesk adaptada. Se escogieron estas tres medidas porque

fueron las que mejor resultado obtuvieron en el análisis realizado por Pedersen *et al.* [58]. Cabe mencionar que el proceso de etiquetado automático de sentidos se basa en la información proporcionada por WordNet 2.0, el uso de medidas de similitud semántica implementadas en la librería *WordNet::Similarity* y la aplicación del algoritmo de McCarthy *et al*.[54].

La medida de Jiang–Conrath se basa en el *contenido de información* y las longitudes de rutas entre conceptos. A diferencia de Resnik [64], que fue quien introdujo este concepto, Jiang–Conrath no sólo usan el *contenido de información* proporcionado por el LCS de ambos conceptos, sino también incluyen el *contenido de información* proporcionado por cada uno de los conceptos comparados.

Lin también se basa en el mismo principio cuantificando la información común entre dos conceptos, como la proporción entre el LCS y el *contenido de información* de cada uno ellos. Esta medida y la anterior son muy parecidas; sin embargo, fueron desarrolladas en forma independiente.

La medida de Lesk adaptada es una variación del algoritmo propuesto inicialmente por Lesk, el cual se basa en la comparación de glosas. Esta adaptación consiste en tomar en cuenta las glosas de los vecinos del término ambiguo y, explotando los conceptos jerárquicos de WordNet, incluir las glosas de aquellos vocablos con los cuales se encuentra relacionado cada uno de los vecinos del vocablo ambiguo.

El objetivo principal del algoritmo de McCarthy *et al*. [54] es etiquetar un vocablo ambiguo con su sentido más predominante, basándose en un conjunto de términos similares ponderados, los cuales son proporcionados por el tesauro de Lin. Dicho recurso selecciona tales términos teniendo en cuenta el sentido más usual del vocablo polisémico. Por ejemplo, cuando se desea identificar el sentido del vocablo *star*, se obtienen los siguientes vocablos similares: *celebrity, idol, VIP, prominent, leading* entre otros; los cuales permiten inducir que el sentido de *star* será el que hace referencia a espectáculo; sin embargo es posible que en algunos casos haga referencia al sentido de astrología, lo cual no sería detectado, ya que este método siempre concluirá que *star* hace referencia al sentido de espectáculo porqué no toma en cuenta el contexto del vocablo ambiguo. Pese a dicha limitación, curiosamente este método es el que ha logrado mejores resultados que todos aquellos métodos de desambiguación existentes hasta el momento, los cuales usan el contexto para intentar seleccionar el sentido correcto. El algoritmo de McCarthy *et al*. se detalla brevemente a continuación. Sea:

- *w* el vocablo ambiguo,
- $V_w = \{v_1, v_2, v_3 ... v_k\}$, los vecinos de *w*,

- $P_w = \{(p, v_1), (p, v_2), (p, v_3)...(p, v_k)\}$ los pesos ponderados entre el vocablo ambiguo y cada vecino y,
- $\text{Sentidos}(w)$ el conjunto de sentidos de *w* según WordNet.

Para cada sentido de *w* ($w_{si} \in \text{Sentidos}(w)$) se obtiene un valor que es usado para seleccionar el sentido del vocablo polisémico. La figura 13 muestra el cálculo del peso para cada sentido del término ambiguo, el cual resulta del cociente de:

- *pswn* (peso de similitud basado en WordNet), que expresa el valor máximo de similitud que resulta de comparar w_{si} y cada uno de los sentidos del vecino con el que se está comparando ($v_{sx} \in \text{Sentidos}(v_i)$) y,
- la suma de todos los *pswn* de cada uno de los sentidos del término ambiguo y el vecino con el que se está comparando.

Finalmente, el puntaje obtenido por cada sentido es expresado por la ecuación 10.

$$\text{Peso}(w_{si}) = \sum_{v_j \in V_w} P(w, v_j) \times \frac{\text{pswn}(w_{si}, v_j)}{\sum_{w_{si} \in \text{sentidos}(w)} \text{pswn}(w_{si}, v_j)} \qquad (10)$$

El *pswn* toma el valor máximo que resulta de la comparación entre un sentido de *w* y cada uno de los sentidos del vecino en curso ($v_{sx} \in \text{Sentidos}(v_j)$); es decir, se maximiza dicho valor, tal como se muestra en la ecuación 11.

$$\text{pswn}(w_{si}, v_j) = \max_{s_x \in \text{sentidos}(v_j)} (\text{pswn}(w_{si}, s_x)) \qquad (11)$$

El método propuesto en esta tesis usa el algoritmo de comparación implementado por McCarthy *et al*. para seleccionar un sentido para un vocablo ambiguo tomando en cuenta un conjunto de vocablos similares, los cuales son obtenidos como una lista ponderada de términos suministrada por el módulo encargado de la recuperación de palabras similares; con la diferencia de que éstos son seleccionados tomando en cuenta el contexto del vocablo ambiguo y no su sentido predominante.

La implementación de este algoritmo ha sido realizado en Perl usando las medidas de similitud proporcionadas por la librería *WordNet::Similarity*. También se ha creado una base de datos de sentidos implementada en SQL Server, que funciona como memoria caché, ya que acelera los procesos de comparación entre sentido-sentido y sentido-vocablo. Cada vez que el proceso solicite la comparación de dos sentidos o el sentido de un término ambiguo que más se aproxime a otro, primero se consulta a la base de datos sobre la existencia de dicho valor, y en caso que no existiese, recién se procede a realizar el cómputo necesario sobre WordNet (dicho cómputo es muy costoso ya que WordNet no es

un motor de base de datos; sino un conjunto de archivos que organizan cierta información), y después se almacena dicho valor con la medida utilizada, de tal manera que cuando esta comparación vuelva a ser solicitada, el valor correspondiente sería tomado directamente de la base de datos. La figura 13 que se presenta a continuación, ilustra este análisis.

4.5.1 Implementación del algoritmo de McCarthy *et al.*

En esta sección se presenta la implementación del algoritmo de McCarthy *et al*. A diferencia del original, éste presenta la interacción con una base de datos que proporciona diversos valores de similitud. Los métodos más importantes se mencionan a continuación:

- El método *consulta_sentidos_BD* consulta a la base de datos de sentidos para obtener el valor de similitud entre dos glosas usando una métrica específica.
- El método *compara_sentido_vecino_BD* consulta a la base de datos de sentidos para obtener el máximo valor de similitud entre una glosa específica y todas las glosas de un vocablo polisémico tomando en cuenta una métrica específica.
- El método *pswn* consulta a WordNet para obtener el máximo valor de similitud entre una glosa y todas las glosas de un vocablo ambiguo, en el caso de que dicha información no exista en la base de datos.
- El método *medida_similitud_WN* consulta las bases textuales de WordNet para computar el algoritmo correspondiente a la métrica de similitud seleccionada. De esta manera obtieneel valor de similitud entre dos sentidos específicos.
- El método *inserta_valor_BD* se encarga de almacenar en la base de datos los valores resultantes de comparar el máximo valor de similitud entre un sentido y todos los sentidos de un vocablo, o el valor de similitud entre dos sentidos. En ambos casos, se toma en cuenta la métrica con la que fue computado dicho valor.

Finalmente, la implementación de dicho algoritmo se detalla en la figura 14. Para ello, se especifican a continuación las variables utilizadas en su desarrollo.

- *w* el vocablo ambiguo,
- *vecinos* el arreglo de palabras similares a *w*,
- *w_vecino* el arreglo de pesos de palabra similares a *w*,
- *w_sentido* el peso final del sentido,
- w_{si}, el *i-ésimo* sentido de *w*,
- *w_similitud* el peso de similitud entre dos sentidos,
- *w_maximo* una variable auxiliar usada por el método *pswn* para computar el máximo peso de similitud entre un sentido y una glosa y,
- *medida* una cadena que designa la métrica de similitud a usar.

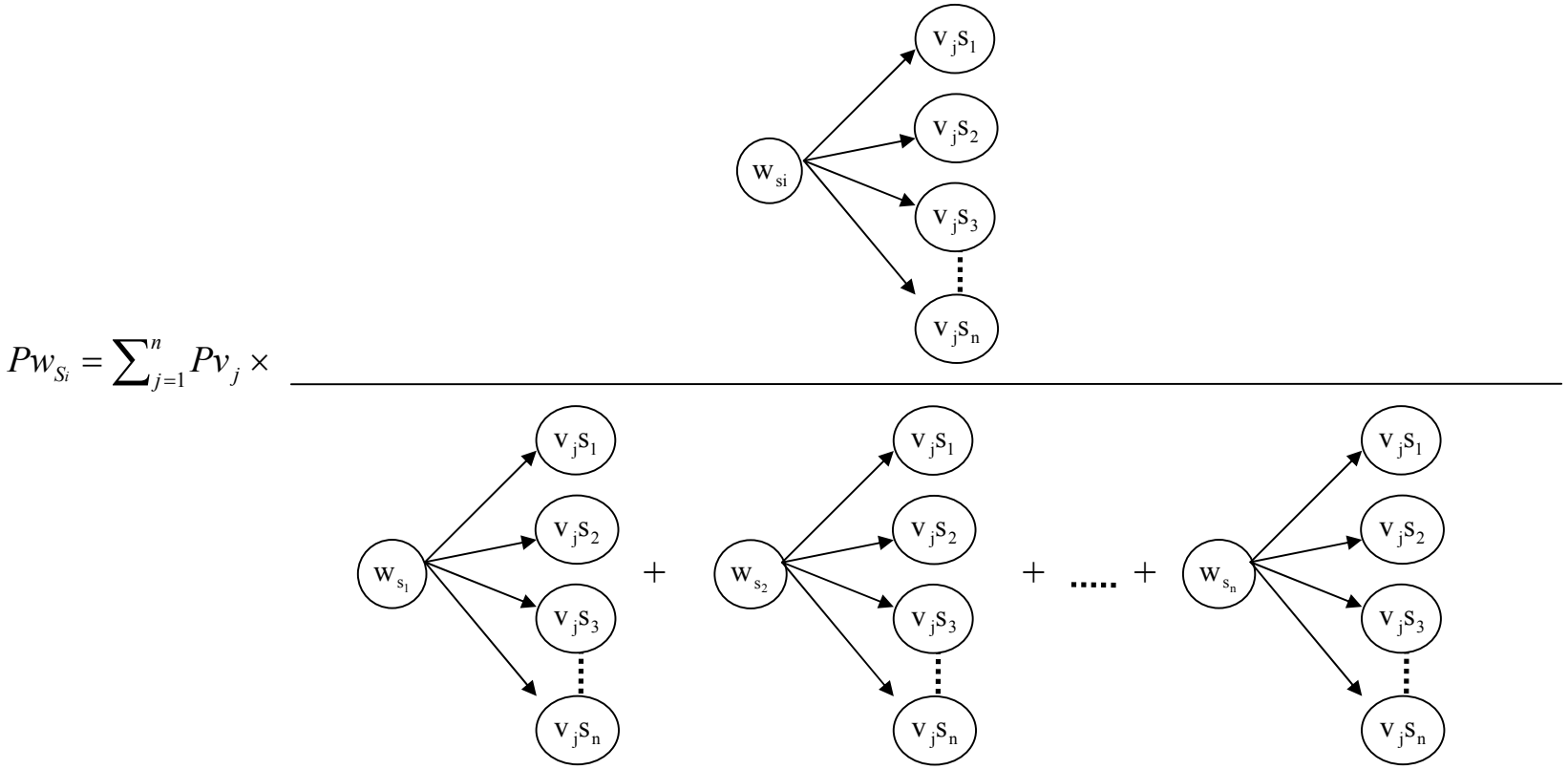


Figura 13. Cálculo del peso para cada sentido de un vocablo ambiguo

```
Para cada sentido i de w
   Para cada vecino j de w
      numerador = consulta_sentido_vecino_BD(w_si,vecinos[j],medida)
      Si numerador = -1
         numerador = pswn(w_si,vecinos[j])
         inserta_valor_BD(w_si,vecinos[j],numerador,medida)
      Fin Si

      Para cada sentido k de w
         w_similitud = consulta_sentido_vecino_BD(w_sk,vecinos[j],medida)
         Si w_similitud = -1
            w_similitud = pswn(w_sk,vecinos[j],medida )
            inserta_valor_BD(w_sk,vecinos[j],w_similitud,medida)
         Fin Si
         denominador = denominador + w_similitud
      Sgte sentido k

      Si denominador > 0
         w_sentido = w_sentido + w_vecino[j] * (numerador/denominador)
      Fin Si
   Sgte vecino j
   almacenar_peso_sentido
Sgte sentido i

pswn(w_si,vecino[j],medida)
w_maximo = 0
Para cada sentido i de vecino[j]
   w_similitud = consulta_sentidos_BD(w_si,sentido_i_vecinos[j],medida)
   Si w_similitud = -1
      w_similitud = medida_similitud_WN(w_si,sentido_i_vecinos[j],medida)
      inserta_valor_BD(w_si,sentido_i_vecinos[j],w_similitud,medida)
   Fin Si
   Si w_similitud > w_maximo
      w_maximo = w_similitud
   Fin Si
Sgte sentido i
Return w_similitud
```

Figura 14. Implementación del algoritmo de McCarthy et al.

Capítulo 5
Resultados experimentales

Este capítulo se encuentra organizado en cuatro secciones. En la primera se define las métricas de evaluación (*recall* y *precision*) utilizadas para verificar los resultados obtenidos por el método propuesto. Después, se detallan ciertas características tomadas en cuenta en la elaboración de los experimentos. La tercera sección presenta los resultados propiamente dichos, mostrando las tablas y gráficas respectivas. Finalmente, se presenta una discusión sobre los resultados obtenidos.

5.1 Métricas de evaluación

Para evaluar el rendimiento del método propuesto, se han aplicado dos métricas de evaluación muy usadas en el área de desambiguación de sentidos de palabras; tales como son *precision* y *recall*.

Un método de desambiguación puede asignar una etiqueta semántica a un vocablo correcta e incorrectamente; incluso podría determinar la carencia de información necesaria como para asignar un sentido específico a un término; en cuyo caso dicha etiqueta sería considerada como nula. *Precision* evalúa aquellos vocablos cuyas etiquetas no son nulas, y *recall* incluye a todos los términos sin tomar en cuenta el valor de sus etiquetas. Consecuentemente, el valor obtenido con *precision* siempre será mayor o igual al reportado por *recall*.

Es muy importante que un método pueda identificar aquellos vocablos que no puede desambiguar; ya que éstos podrían ser procesados por otros que tengan mayor *precision*. De esta manera se podrían obtener resultados más confiables. Las ecuaciones concernientes a estas métricas se muestran en las ecuaciones 12 y 13.

$$\text{precision} = \frac{\text{buenas}}{\text{buenas} + \text{malas}} \tag{12}$$

$$\text{recall} = \frac{\text{buenas}}{\text{buenas} + \text{malas} + \text{nulas}} \tag{13}$$

donde:

buenas es el número de respuestas correctas proporcionadas por el sistema,
malas es el número de respuestas incorrectas proporcionadas por el sistema y,
nulas es el número de respuestas nulas proporcionadas por el sistema.

5.2 Descripción de experimentos

Esta sección puntualiza de manera explícita algunas características que han sido tomadas en cuenta en el desarrollo de los experimentos. Éstas se detallan a continuación:

5.2.1 Categoría gramatical del término ambiguo

Las diferentes categorías gramaticales, tales como sustantivos, adjetivos, adverbios y verbos, influyen directamente en el proceso de desambiguación de un vocablo. Yarowsky [83] afirma que los verbos derivan mayor información de desambiguación de sus objetos que de sus sujetos, mientras que los adjetivos derivan mejor información de los sustantivos a los que modifican y éstos son mejor desambiguados cuando tienen adjetivos o sustantivos adyacentes.

Los experimentos realizados han tomado en cuenta dicha característica. La estructura de jerarquías de WordNet es la razón principal por la que la medida de Lin y la propuesta por Jiang–Conrath requieren vecinos cuya categoría gramatical sea la misma que el término ambiguo. Sin embargo, la medida de Lesk adaptada, basada en la comparación de glosas, no toma en cuenta dicha característica.

En los experimentos se han desambiguado sustantivos; ya que éstos se encuentran organizados en una jerarquía de hiperónimos. Los adjetivos y adverbios no cuentan con una jerarquía de este tipo y pese a que los verbos si están organizados como hiperónimos; las definiciones de sus sentidos son tan sutiles que los resultados a obtener no son prometedores y además, son los que más sentidos presentan en WordNet.

5.2.2 Número de vecinos

Uno de las características que aún no ha sido claramente definida por la comunidad lingüística actual, es el número de términos similares (vecinos) necesarios para lograr un proceso de desambiguación exitoso. Debido a esta razón, los experimentos desarrollados han tomado como referencia varias cantidades, específicamente 10, 20, 30, 40, 50, 60, 70, 100 y 1000 vecinos. Es necesario notar que el número de vecinos es una variable configurable en el módulo de recuperación de términos similares.

5.2.3 Número de términos en el contexto sintáctico

El número de términos contextuales necesarios para obtener vecinos que tengan una estrecha relación semántica con el vocablo ambiguo, es otra característica que tampoco se

encuentra claramente definida. Los experimentos que se presentan en esta sección han sido realizados sobre sustantivos cuyo contexto sintáctico comprenda al menos dos términos.

5.2.4 Recurso empleado

Éste es otro de los principales factores que influyen en el proceso de desambiguación. Por lo general, la mayoría de métodos basados en contextos previamente compilados, generan las relaciones de dependencia usando corpus textuales de varios cientos de millones de palabras; sin embargo en esta tesis se usó SemCor, corpus que no rebasa el medio millón de palabras.

La base de datos de recursos sintácticos utilizada y generada con SemCor, es aquella que se encuentra compuesta por tuplas que cumplen cualquiera de las tres relaciones de dependencia presentadas en este trabajo (*convencional, sin preposición y especial*).

5.2.5 Medidas de similitud

Las medidas de similitud utilizadas por el algoritmo de etiquetado automático de sentidos, han sido las propuestas por Jiang–Conrath, la medida de Lin y la medida de relación semántica presentada por Banerjee–Pedersen. Estas fueron elegidas por ser las que mejores resultados obtuvieron en el trabajo publicado por Pedersen *et al*.[60].

5.3 Resultados

Para cada uno de los experimentos que se presentan a continuación, se han tomado 300 sustantivos elegidos aleatoriamente de las oraciones contenidas en los diferentes archivos de SemCor. Las características generales del grupo seleccionado se presentan en la tabla 16.

Tabla 16. Características de los sustantivos a desambiguar

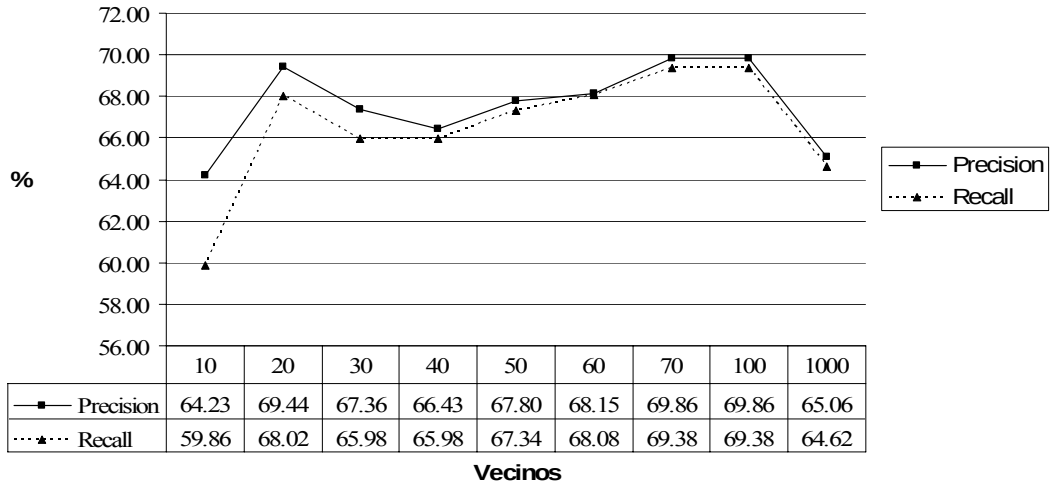
Característica	**Cantidad**
Total de sustantivos	300
Sustantivos sin contexto confiable	89
Sustantivos no existentes o no ambiguos según WordNet	64
Vocablos ambiguos según WordNet	147

Los resultados correspondientes a cada experimento son presentados en gráficas de dos dimensiones, donde el eje *x* representa al número de vecinos usados, y el eje *y* el

porcentaje de aciertos reportado por cada una de las métricas de evaluación. Veánse las figuras 13 a 17.

5.3.1 Experimento usando la medida de Jiang–Conrath

Como se puede observar en la figura 15, las gráficas no muestran un comportamiento predecible; sin embargo, se puede apreciar que los resultados obtenidos se mantienen levemente constantes cuando el número de vecinos empleados varía entre 30 y 60. En dichos casos se obtienen valores entre 66% y 68% para ambas métricas de evaluación. El resultado más alto es obtenido cuando se usan 70 ó 100 vecinos, en cuyo caso se reportaron valores de 69.86% para precision y 69.38% para recall. Asimismo, el resultado más bajo se obtuvo al desambiguar un vocablo usando 10 vecinos, en cuyo caso se obtuvo valores de 64.23% y 59.86% para precision y recall respectivamente.



	10	20	30	40	50	60	70	100	1000
Precision	64.23	69.44	67.36	66.43	67.80	68.15	69.86	69.86	65.06
Recall	59.86	68.02	65.98	65.98	67.34	68.08	69.38	69.38	64.62

Vecinos

Figura 15. Resultados obtenidos por la medida de Jiang–Conrath

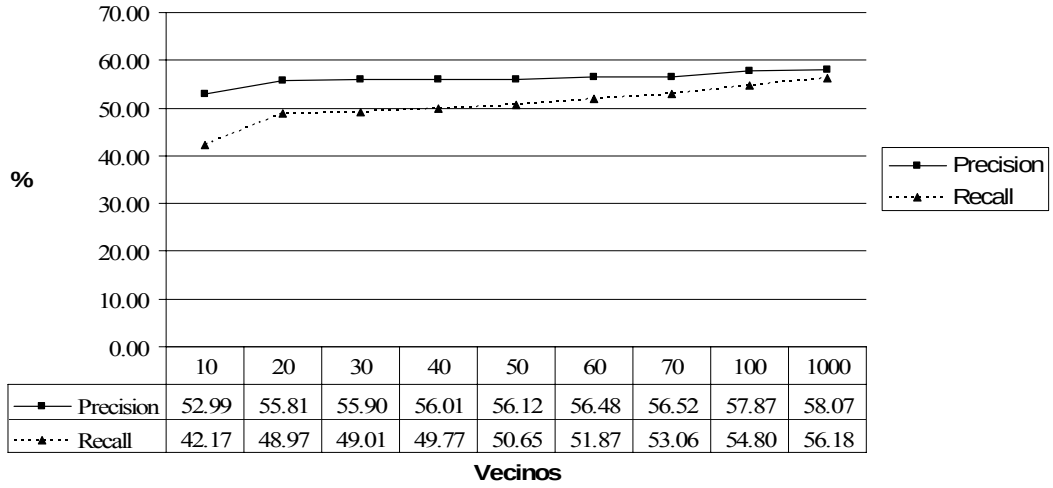
Curiosamente, el segundo mejor resultado se obtuvo al usar 20 vecinos, lo cual puede ser justificado por la discontinuidad de la gráfica. Tal discontinuidad demuestra que los buenos resultados que puede reportar esta medida, no dependen de la cantidad de vecinos procesados (a mayor número de vecinos los resultados no mejoran); sino de la calidad semántica que aporta cada uno de ellos.

Finalmente, la diferencia entre la sumatoria total de los valores reportados para cada una de las métricas de evaluación es mínimo (67.58 para *precision* y 66.52 para *recall*), lo cual significa que esta medida siempre intenta etiquetar un vocablo; de tal manera que son muy pocas las veces que asigna un valor nulo a una etiqueta. Por estas razones, el *recall* reportado es relativamente alto tomando como referencia el valor obtenido con *precision*.

5.3.2 Experimento usando la medida de Lin

Como se puede observar en la figura 16, las gráficas obtenidas por la medida de Lin muestran mayor regularidad que la anterior, ya que los resultados tienden a mejorar cuando el número de vecinos se incrementa. De esta manera, el mejor resultado se obtiene cuando se utilizaron 1000 vecinos en el proceso de desambiguación. Sin embargo, es necesario aclarar que dicha mejora no es muy relevante. Por ejemplo, cuando el número de vecinos se encuentra entre 20 y 1000, la diferencia entre el mayor y menor resultado obtenido por *precision*, es tan sólo de 3%.

Finalmente, el valor acumulado reportado por *recall* es relativamente bajo comparado con el acumulado obtenido por *precision*. Esto significa que éste, a diferencia del método anterior, etiqueta más vocablos como nulos, lo cual puede ser beneficioso o no, dependiendo de las circunstancias. Por ejemplo, si existieran otros métodos alternos con mejor *precision* que puedan procesar los vocablos con etiquetas nulas, el hecho de contar con un *recall* bajo es favorable, ya que si un método no cuenta con la información necesaria como para asignar una etiqueta, es muy probable que otro método pueda hacerlo.



	10	20	30	40	50	60	70	100	1000
Precision	52.99	55.81	55.90	56.01	56.12	56.48	56.52	57.87	58.07
Recall	42.17	48.97	49.01	49.77	50.65	51.87	53.06	54.80	56.18

Vecinos

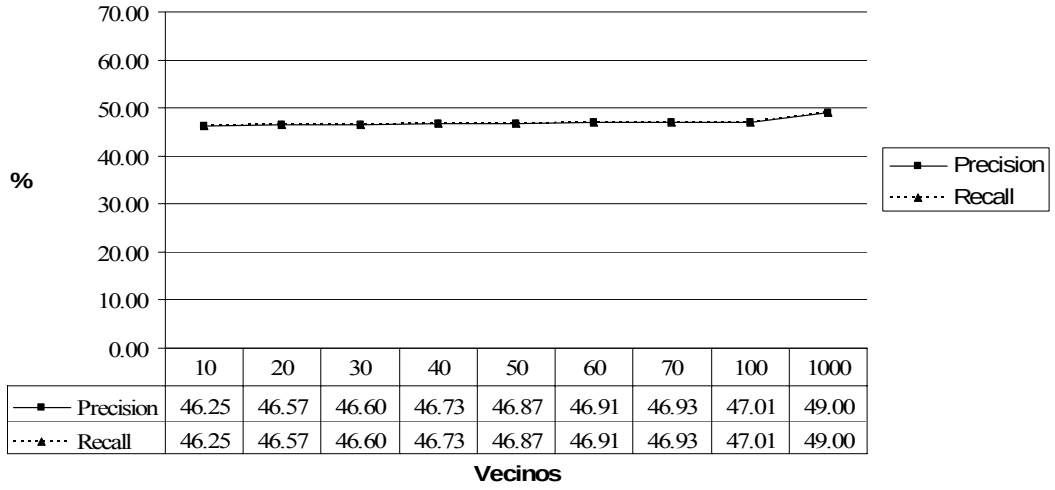
Figura 16. Resultados obtenidos por la medida de Lin

5.3.3 Experimento usando la medida de Banerjee–Pedersen

Como se puede apreciar en la figura 17, la medida presentada por Banerjee–Pedersen es la que obtiene los resultados más bajos. Una de las características de ésta medida la regularidad de su gráfica, la cual es independiente del número de vecinos utilizados. La explicación de esto radica en la modificación del algoritmo de Lesk implementada en esta medida; la cual no sólo compara el término ambiguo con cada una de

las glosas de sus vecinos; sino que también se compara con cada una de los vocablos con los cuales se relaciona cada vecino tomando como referencia las jerarquías de WordNet.

Luego, los valores reportados por *recall* y *precision* son los mismos, lo cual significa que este método siempre es capaz de dar una respuesta ya sea mala o buena. Esta característica surge como consecuencia directa de la comparación de glosas en la que se basa el algoritmo de Lesk. Esto significa que al comparar la definición de dos sentidos usando el algoritmo de Lesk; siempre se obtendrá algun valor escalar que refleje la proximidad semántica entre ambos conceptos. Finalmente, al igual que la medida de Lin y basándose en el comportamiento de la gráfica obtenida, mientras más términos similares al vocablo ambiguo, esta medida reportará mejores resultados.



	10	20	30	40	50	60	70	100	1000
Precision	46.25	46.57	46.60	46.73	46.87	46.91	46.93	47.01	49.00
Recall	46.25	46.57	46.60	46.73	46.87	46.91	46.93	47.01	49.00

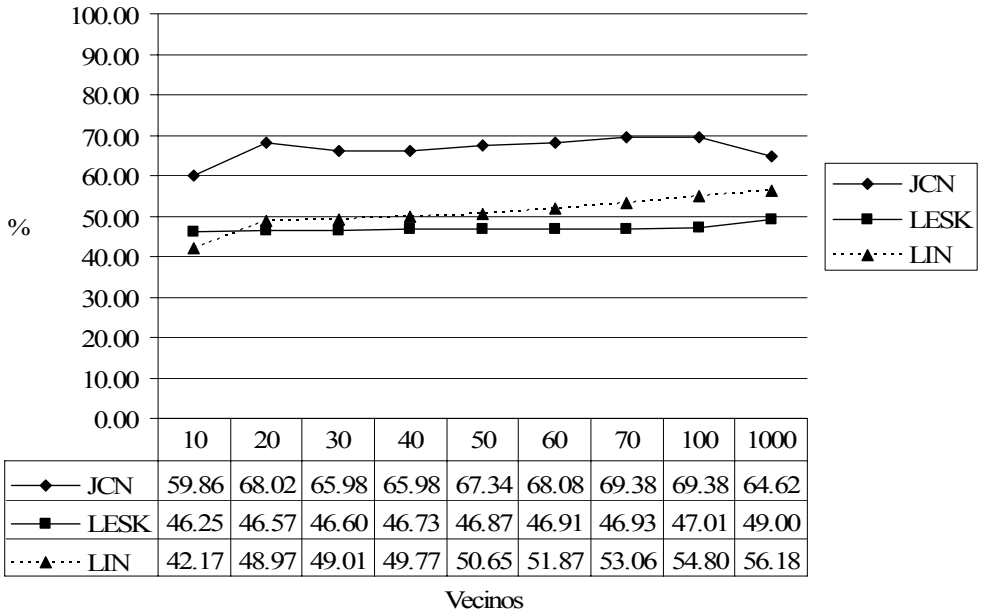
Figura 17. Resultados obtenidos por la medida Lesk adaptada

5.3.4 Comparación de medidas

En esta sección se presenta la comparación tabular y gráfica entre las tres medidas de similitud especificadas en este capítulo. La figura 18 muestra los resultados para la métrica *precision*, mientras que la figura 19 muestra los resultados para la métrica *recall*.

El objetivo de estas gráficas es determinar la medida de similitud cuyo rendimiento haya sido el más alto. Para ello, es necesario comparar los valores reportados por cada una tomando en cueta las métricas de evaluación *precision* y *recall*. En términos generales, la medida de Jiang–Conrath [34] es la que obtuvo los mejores resultados para ambas métricas de evaluación, seguida por la medida de Lin [51], y finalmente la medida de Lesk adaptada [46].

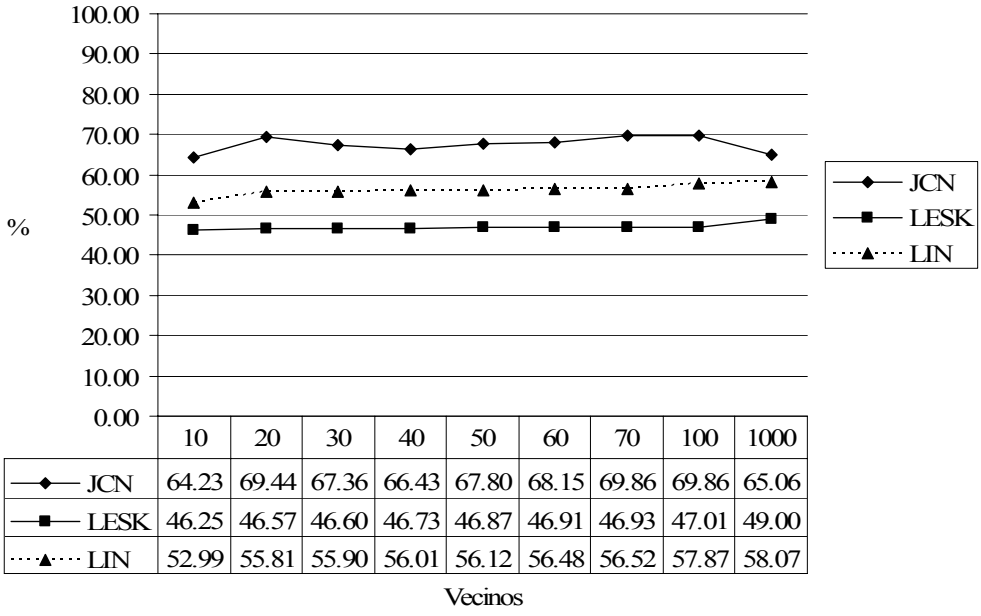
Es necesario aclarar que el éxito de desambiguación de un vocablo depende de la medida de similitiud usada y el recurso sintáctico empleado; por lo tanto, es muy probable que dichos resultados pudieran cambiar si se utilizaran otros vocablos en los experimentos planteados. Pese a ello, es posible determinar de manera genérica la medida que tiene mayor aceptación entre los diversos vocablos ambiguos.



	10	20	30	40	50	60	70	100	1000
JCN	59.86	68.02	65.98	65.98	67.34	68.08	69.38	69.38	64.62
LESK	46.25	46.57	46.60	46.73	46.87	46.91	46.93	47.01	49.00
LIN	42.17	48.97	49.01	49.77	50.65	51.87	53.06	54.80	56.18

Vecinos

Figura 18. Valores reportados por la métrica de evaluación "precision"



	10	20	30	40	50	60	70	100	1000
JCN	64.23	69.44	67.36	66.43	67.80	68.15	69.86	69.86	65.06
LESK	46.25	46.57	46.60	46.73	46.87	46.91	46.93	47.01	49.00
LIN	52.99	55.81	55.90	56.01	56.12	56.48	56.52	57.87	58.07

Vecinos

Figura 19. Valores reportados por la métrica de evaluación "recall"

5.4 Discusión

De las tres medidas de similitud utilizadas por el programa que etiqueta automáticamente un vocablo, la que mejores resultados ha conseguido es la propuesta por Jiang–Conrath que reportó valores de 69.86% para *precision* y 69.38% para *recall*. Ambos valores son realmente altos; sin embargo, es necesario considerar que éstos han sido calculados tomando en cuenta sólo aquellos vocablos ambiguos cuyo contexto sintáctico está conformado por al menos dos palabras, de tal manera que de los 300 vocablos procesados, 89 no cumplen con dicha condición y 64 no presentan ambigüedad o no existen en WordNet. Esto último, se debe a la existencia de sustantivos compuestos, los cuales son proporcionados por el analizador sintáctico, tales sustantivos como los siguientes: Commission_Chairman_Charles_O._Emmerich, Atlanta_Bar_Association, etc. De esta manera, al descontar los vocablos que no cumplen con las características mencionadas, los experimentos utilizaron sólo 147 vocablos ambiguos tomando como referencia WordNet, con contextos mayores o iguales a dos términos.

Si bien es cierto que la medida propuesta por Jiang–Conrath es la que mejores resultados ha obtenido, las tres medidas presentadas tienen ventajas y desventajas, de tal forma que es posible mejorar cada una de ellas de manera muy particular, dependiendo de la implementación y los recursos que necesite cada algoritmo. Por ejemplo, la irregularidad mostrada en la gráfica de la medida de Jiang–Conrath, es un claro indicador que su algoritmo no requiere de muchos vecinos para desambiguar un vocablo; más bien lo que necesita son vecinos con mucho contenido semántico. Por ende, para mejorar sus resultados es necesario suministrar términos de mucha relación semántica con el vocablo a desambiguar.

Ahora bien, la medida de Lin proporciona mejores resultados cuando se incrementa el número de vecinos. Por ende, si se desease mejorar su rendimiento es necesario proveerle algunos miles de vecinos extras; sin embargo, surge el problema del costo computacional. Realmente, ya es muy costoso procesar 1000 vecinos como para intentar procesar más. Entonces, es necesario optimizar las estructuras de datos y el algoritmo propiamente dicho de esta medida, para que sea capaz de mejorar sus resultados sin afectar su rendimiento computacional.

Con respecto a la medida de Lesk adaptada, la que peores resultados obtuvo, se puede observar que los valores obtenidos para *precisión* y *recall* son iguales. Esto se debe a la comparación de glosas en la que se basa su algoritmo, lo cual hace que siempre proporcione una etiqueta para un vocablo ambiguo. Los bajos resultados obtenidos se deben a que sólo se utilizaron vecinos cuya categoría gramatical fuesen sustantivos; característica que es indiferente para esta medida; ya que es posible comparar las glosas de

un sustantivo con las de un adjetivo o las de un verbo; lo cual no es posible en las otras dos medidas presentadas. Por ende, si se suministrasen vecinos con otras categorías gramaticales, es muy probable que sus resultados incrementen.

Finalmente, es muy difícil comparar la eficiencia de estas medidas bajo condiciones iguales para cada una de ellas; ya que como se explicó, las características que influyen directamente en los resultados de una no son relevantes para otra y viceversa. Consecuentemente, es factible mejorar los resultados de cada una, incidiendo en las características específicas de cada una.

Capítulo 6
Conclusiones y trabajo futuro

En esta tesis se ha presentado un panorama genérico sobre WSD, así como las diferentes técnicas y recursos de conocimiento utilizados para enfrentar este problema; sin embargo, pese a todo el esfuerzo desplegado en esta área, WSD sigue siendo un problema abierto, a lo cual se suma la dificultad de determinar y definir correctamente los sentidos de las palabras.

Aunque WordNet ha llegado a convertirse en el recurso léxico por defecto, éste tiene algunas limitaciones, tal como la clara distinción que hace entre las definiciones de los sentidos de un vocablo; lo cual muchas veces no es aplicable a las expresiones que se dan en el mundo real. Pese a estos inconvenientes, parece que el uso de grandes recursos de información y una organización del conocimiento correcta serán los pilares para dar una solución final a este problema.

6.1 Conclusiones

- La desambiguación de sentidos de palabras es una tarea que involucra muchos procesos y recursos lingüísticos, tales como analizadores sintácticos, morfológicos, medidas de similitud semántica, diccionarios computacionales, bases de datos de recursos sintácticos y de sentidos.

- Las relaciones de dependencia sintáctica presentadas (*sin preposición* y *especial*), son una buena forma de adquirir el contexto sintáctico de un término ambiguo; sin embargo, aún es posible mejorar esta información usando los diferentes niveles del árbol de dependencia sintáctica.

- Definitivamente, la desambiguación exitosa de una palabra depende del contexto en el que se presenta. Es por ello, que el módulo de recuperación de términos similares se basa en la comparación de contextos sintácticos. La importancia de estos términos puede verse reflejada en los buenos resultados obtenidos, los que a su vez demuestran que el modelo del espacio vectorial es un buen método a seguir para organizar el recurso sintáctico adecuadamente.

- El uso de un espacio semántico que presente cierta organización de la información, tal como WordNet, es muy necesario en WSD, ya que ésta es una de las maneras mas eficientes para suministrar conocimiento a una computadora. Por lo tanto, mientras

mejor organizado se encuentre, las medidas de similitud semántica obtendrán cada vez mejores resultados.

- El algoritmo para la obtención del sentido más predominante propuesto por McCarthy *et al*. [54], cuyos términos similares son proporcionados por el tesauro de Lin, obtiene mejores resultados cuando es utilizado para especificar el sentido que expresa un vocablo ambiguo en un contexto específico, en cuyo caso sus términos similares son suministrados dependiendo de su contexto sintáctico.

- Una de las deficiencias del método propuesto, es que los contextos sintácticos de todos los vocablos procesados son tratados de igual manera; sin tener en cuenta que algunas características de dichos contextos proporcionan mayor o menor información. Esto depende principalmente de la categoría gramatical del vocablo ambiguo.

6.2 Aportaciones

Las aportaciones que se derivan de este trabajo se han dividido en dos rubros: aportaciones al conocimiento y técnicas, las cuales serán descritas a continuación.

6.2.1 Aportaciones al conocimiento

- Creación de un método de desambiguación no supervisado que ha obtenido mejores resultados que los reportados por el mejor método no supervisado conocido hasta el momento, el cual ha sido propuesto por McCarthy *et al*. [54]. El mejor resultado que se obtuvo en nuestro trabajo fue de 69.86% de *precision* usando la medida de Jiang–Conrath, mientras que el mejor resultado obtenido por McCarthy *et al*. fue de 67% de *precision* usando la medida de Lesk.

- Adquisición de las características del contexto mediante relaciones de dependencia sintáctica no estándares; es decir, relaciones que generalmente no son tomadas en cuenta para este propósito. Específicamente, me refiero a las relaciones de dependencia *sin preposición* y *especial*.

6.2.2 Aportaciones técnicas

Actualmente, la resolución de la desambiguación de sentidos de palabras está enfocada al uso de grandes recursos de información y herramientas lingüísticas muy específicas a los diferentes procesos que forman parte de WSD. Este trabajo ha seguido tales lineamientos; por ende se han desarrollado algunas de estas herramientas y dos

importantes recursos de información. Ambos pueden ser usados en cualquier área relativa al procesamiento de lenguaje natural. A continuación se describe cada una de estas aportaciones.

a. Analizador sintáctico de dependencias basado en MINIPAR

MINIPAR es un conjunto de librerías suministradas por Dekang Lin, las cuales han sido desarrolladas en lenguaje C. Éstas proporcionan la funcionalidad necesaria para analizar sintáctica y morfológicamente los vocablos de una oración. En esta tesis se ha implementado un analizador sintáctico, el cual utiliza dichas librerías con la finalidad de generar árboles de dependencia sintáctica para todas las oraciones de un corpus de texto.

b. Programa para la extracción de tripletas de dependencia sintáctica

Este programa permite extraer relaciones de dependencia *convencional*, *sin preposición* y *especial*, utilizando como fuente de información el árbol proporcionado por el analizador sintáctico.

Es necesario mencionar que las relaciones *convencionales* utilizadas por la lingüística computacional no identifican plenamente el contexto sintáctico de un vocablo; es por ello que se han creado relaciones *sin preposición* y *especial*. En la primera, la preposición es excluida cuando se presenta como modificador de algún vocablo, de tal manera que el término dependiente de una preposición, pasa a ser el modificador de dicho vocablo. En las relaciones de dependencia *especial* se incluye como parte del conjunto de los modificadores de un vocablo, el término al que modifica dicho vocablo.

c. Programa para la recuperación de palabras similares

Este programa permite obtener términos similares a un vocablo específico, basándose en su contexto sintáctico. Para ello, se compara dicho contexto con otros 50,000 previamente compilados en una base de datos. La similitud de dos contextos es computada usando las estadísticas proporcionadas por el modelo de espacio vectorial (el esquema TF-IDF), el cual generalmente ha sido aplicado a tareas de clasificación y similitud de documentos.

Este modelo organiza las relaciones de dependencia encontradas en un contexto como tuplas, las cuales se encuentran conformadas principalmente por un término principal o *cabeza* y sus modificadores sintácticos. Posteriormente; se crean vectores multidimensionales para cada *cabeza*; donde cada modificador representa una dimensión. Finalmente, la similitud de dos vectores está dada por el coseno del ángulo que forman en un espacio multidimensional.

d. Programa para el etiquetado automático de sentidos

Este programa permite elegir el sentido de un vocablo basándose fundamentalmente en sus términos similares y en un espacio semántico, tal como WordNet. El programa de etiquetado semántico se basa en el algoritmo propuesto por McCarthy *et al*.[54], utilizado para obtener el sentido predominante de un vocablo ambiguo y en ciertas medidas de similitud, tales como las propuestas por Jiang–Conrath, Lin y la métrica de relación propuesta por Banerjee–Pedersen.

e. Base de datos de valores de similitud

Muchas veces, el programa de etiquetado semántico necesita cuantificar la proximidad semántica entre dos glosas y también, obtener el sentido de un vocablo que más se asemeje a otro sentido específico, basándose en una métrica de similitud.

El costo computacional que implica calcular esta información es elevado, ya que WordNet está organizado como un conjunto de archivos planos y cada vez que es necesario calcular estos datos, se tiene que recorrer ciertas jerarquías de información dependiendo del tipo de métrica que se utilice.

Ésta es la razón principal por la que se ha creado una base de datos que almacena más de cien mil valores correspondientes a las mediciones especificadas. Además de acelerar el proceso de desambiguación, dicho recurso puede ser utilizado por cualquier aplicación de procesamiento de lenguaje natural.

f. Base de datos de vocablos relacionados sintácticamente

Las tripletas de dependencia sintáctica obtenidas de SemCor han sido almacenadas en bases de datos. Dependiendo del tipo de relación de dependencia utilizada, se ha creado tres recursos sintácticos:

- Base de datos conformada por tuplas basadas en relaciones de dependencia convencional.
- Base de datos conformada por tuplas basadas en relaciones de dependencia sin preposición.
- Base de datos conformada por tuplas basadas en relaciones de dependencia sin preposición y especial.

Cada una de éstas, está conformada por alrededor de medio millón de pares de vocablos bajo cierta relación de dependencia. Además, cada tripleta cuenta con un valor que especifica el grado de relación semántica existente entre ambos términos.

6.2.3 Publicaciones generadas

- J. Tejada-Cárcamo, A. Gelbukh, H. Calvo. Desambiguación de sentidos de palabras usando relaciones sintácticas como contexto local. Tutorials and Workshops of Fourth Mexican International Conference on Artificial Intelligence, ISBN 968-891-094-5, 2005.

- J. Tejada-Cárcamo. El impacto de relaciones sintácticas y similitud semántica en la desambiguación de sentidos de palabras (preparado).

6.3 Trabajo futuro

Si bien es cierto que la investigación realizada en este trabajo ha obtenido buenos resultados, también ha dejado muchas dudas y nuevas percepciones del área. A continuación, se presentan algunas ideas basadas en la experiencia adquirida, cuyos resultados no son predecibles y la única manera de evaluar su comportamiento es implementándolas.

- Determinar el impacto del tamaño del recurso sintáctico en el método de desambiguación propuesto, ya que el recurso empleado fue construido procesando un corpus pequeño, tal como lo es SemCor, el cual no supera el millón de palabras. Pese a dicha característica, los resultados obtenidos fueron exitosos. Por ende, es necesario experimentar con corpus más extensos (varios millones de palabras) y también, con alguno de menores dimensiones que SemCor.

- Enriquecer el contexto de un vocablo tomando en cuenta varias características lingüísticas, tales como la información existente en los diferentes niveles del árbol sintáctico, colocaciones gramaticales, términos co-ocurrentes y la semántica que expresa el dominio o texto en el que se presenta. Para ello, es necesario crear un método que combine estas fuentes de información y proporcione como resultado un contexto idóneo, en el que cada uno de sus integrantes presente una estrecha relación semántica con el vocablo ambiguo.

- Mejorar el proceso de obtención de términos similares propuesto, complementando el modelo de espacio vectorial con medidas de similitud y relación semántica. Para ello, es necesario computar valores de similitud entre los modificadores de una tupla y su respectiva *cabeza*. El peso de cada modificador será ponderado dependiendo de su valor de similitud y el obtenido por el modelo de espacio vectorial.

- Probar otra alternativa para mejorar el proceso de obtención de términos similares, creando un recurso léxico que almacene los términos necesarios para definir el sentido de un vocablo ambiguo, tomando en cuenta su contexto sintáctico y categoría gramatical. Este recurso puede ser construido procesando los resultados obtenidos por el método planteado, clasificando aquellos términos que desambigüen exitosamente un vocablo, y desechando los que no logren tal propósito.

- Evaluar el impacto de ciertas medidas de relación semántica en el proceso de desambiguación de sentidos de palabras, tales como las propuestas por Hirst–St–Onge, Banerjee–Pedersen, comparándolas con algunas medidas de similitud, tales como las propuestas por Resnik, Lin, Jiang–Conrath, Leacock–Chodorow y Wu–Palmer.

- Modificar el algoritmo de etiquetado semántico propuesto; de tal manera que éste no utilice una sola medida de similitud o relación semántica a la vez; sino que emplee todas aquellas medidas implementadas de una manera conjunta. Es decir, aquel vocablo cuyo sentido no pueda ser definido por una medida; será etiquetado por otra. Para ello, es necesario tomar en cuenta los valores que cada una de éstas reportó para las métricas de evaluación.

- Modificar el módulo encargado de etiquetar semánticamente un vocablo ambiguo. Para ello, se usará el mismo algoritmo de etiquetado semántico, con la funcionalidad adicional de hacerlo tantas veces como el número de medidas de similitud implementadas. De esta manera, cada una de dichas medidas asignará un sentido al término ambiguo. Finalmente, el sentido que más se repita será el asignado a dicho vocablo.

Índice de términos

Glosario

Ambigüedad. Término que hace referencia a aquellas estructuras gramaticales que pueden entenderse de varios modos o admitir distintas interpretaciones y dar, por consiguiente, motivo a dudas, incertidumbre o confusión.

Ambigüedad léxica. La ambigüedad léxica es aquella que se presenta en la categoría gramatical de un vocablo. Es decir, un vocablo puede tener más de un rol gramatical en diferentes contextos.

Ambigüedad semántica. La ambigüedad semántica es aquella que se presenta en una estructura gramatical, de tal manera que ésta puede expresar diferentes sentidos dependiendo del contexto local, el tópico global y el mundo pragmático en el que se manifiesta.

Ambigüedad sintáctica. La ambigüedad sintáctica, también conocida como estructural, es aquella que se presenta en oraciones, de tal manera que éstas puedan ser representadas por más de una estructura sintáctica.

Analizador sintáctico. Un analizador sintáctico para un lenguaje natural, es un programa que construye árboles de estructura de frase o de derivación para las oraciones de dicho lenguaje, además de proporcionar un análisis gramatical, separando los términos en constituyentes y etiquetando cada uno de ellos. Asimismo, puede proporcionar información adicional acerca de las clases semánticas (persona, género) de cada palabra y también la clase funcional (sujeto, objeto directo, etc.) de los constituyentes de la oración.

Antonimia. La antonimia es una relación entre dos palabras que expresan ideas opuestas o contrarias. Por ejemplo, los vocablos virtud y vicio; claro y oscuro; antes y después.

Aprendizaje supervisado. El aprendizaje supervisado se asemeja al método de enseñanza tradicional con un profesor que indica y corrige los errores del alumno hasta que éste aprende la lección. En el caso de la desambiguación supervisada se entrena un clasificador usando un corpus de texto etiquetado semánticamente para obtener el contexto en el que usualmente se presenta cada sentido del vocablo ambiguo. Este clasificador desmabiguará solo aquellos vocablos y sentidos que hayan participado en el entrenamiento previo.

Aprendizaje no supervisado. En el apredizaje no supervisado no existe un profesor que corrija los errores al alumno; ya que éste recuerda más al autoaprendizaje. El alumno

dispone del material de estudio; pero nadie lo controla. En el caso de la desambiguación no supervisada tambien es posible usar un corpus de texto etiquetado como fase entrenamiento; sin embargo los algoritmos no supervisados generalizan esta información para cualquier vocablo ambiguo; aunque no haya estado presente en el corpus de entrenamiento.

Árbol de constituyentes. Un árbol de constituyentes es una estructura de datos que pemite categorizar una oración en sus partes de oración.En el llamado sistema o método de constituyentes la principal operación lógica es la inclusión de elementos en conjuntos, así éstos pertenecen a una oración o a una categoría. Según esta aproximación, una oración es segmentada en constituyentes, cada uno de los cuales es consecuentemente segmentado. Así, esto favorece un punto de vista analítico.

Árbol de dependencias. Un árbol de dependencias es una estructura de datos que pemite obtener las relaciones de dependencia sintáctica entre un núcleo y conjunto de modificadores. La aproximación de dependencias se centra en las relaciones entre las unidades sintácticas últimas, es decir, en las palabras. La principal operación aquí consiste en establecer relaciones binarias. Según esta idea, una oración se construye de palabras, unidas por dependencias.

Biblioteconomía. La biblioteconomía es la disciplina encargada de la conservación, organización y administración de las bibliotecas, incluso cuando son digitales. Esta disciplina es utilizada por los sistemas de recuperación de información.

Bootstrapping. *Bootstrapping* es un término inglés que se refiere al proceso mediante el cual se han desarrollado o implementado soluciones cada vez más complejas a partir de otras más simples. El entorno más simple sería, quizás, un editor de textos muy sencillo y un programa ensamblador. Utilizando estas herramientas, se puede escribir un editor de texto más complejo y un compilador simple para un lenguaje de más alto nivel y así sucesivamente, hasta obtener un entorno interado de desarrollo y un lenguaje de programación de muy alto nivel.

Cabeza. *Cabeza* es un término utilizado en este trabajo para referenciar al vocablo que gobierna una relación de dependencia sintáctica, de tal manera que se se puede obtener muchos modificadores sintácticos para una misma *cabeza*.

Categoría gramatical. El término categoría gramatical o parte de la oración, que en inglés se denomina POS (*part of speech*) es una clasificación de las palabras de acuerdo a la función que desempeñan en la oración. La gramática tradicional distingue nueve categorías gramaticales: sustantivo, determinante, adjetivo, pronombre, preposición, conjunción,

verbo, adverbio, interjección. No obstante, para algunos lingüistas, las categorías gramaticales son una forma de clasificar ciertos rasgos gramaticales, como por ejemplo: modo, aspecto, tiempo y voz.

Colocación gramatical. Una colocación gramatical es un conjunto de dos o más palabras las cuales expresan una idea específica. El significado que expresa cada término de una colocación difiere de la semántica que dichos términos proporcionan cuando se usan de manera conjunta.

Contexto local. El contexto local de un vocablo, que también es conocido como micro-contexto, engloba a un conjunto de palabras cercanas a dicho vocablo. Esta cercanía puede estar limitada por una vecindad de palabras co-ocurrentes, por la oración en la cual se encuentra dicho vocablo o incluso, por el árbol sintáctico al cual pertenecen.

Dependencia convencional. Una relación de dependencia sintáctica *convencional* es aquella en la que una pareja de vocablos mantiene una relación de dependencia tradicional especificada por el árbol de dependencias sintácticas. Más explícitamente, las flechas que salen de un vocablo hacia otros se consideran los modificadores sintácticos del primero

Dependencia sin preposición. Una relación de dependencia especial es aquella que excluye la preposición cuando ésta se presenta como modificador del núcleo de una relación, de tal manera que el término que depende de ésta pasa a ser el modificador del núcleo.

Dependencia especial. Una relación de dependencia sintáctica *especial* incluye como parte del conjunto de los modificadores del núcleo de la relación, el vocablo al que modifica el mismo núcleo. De esta manera, incrementa su número de modificadores.

Diccionario computacional. Un diccionario computacional surge al convertir un diccionario normal, creado exclusivamente para el uso humano, a formato electrónico. Estos diccionarios proveen información sobre sentidos de vocablos ambiguos, lo cual puede ser explotado por el área de desambiguación de sentidos de palabras.

Dominio. El término dominio hace referencia a la temática general que expresa un documento o texto en su totalidad.

Esquema TF-IDF. El esquema TF-IDF o modelo de espacio vectorial es una arquitectura que generalmente se aplica a tareas de clasificación y similitud de documentos. Ésta representa un documento con un vector y, cuando se desea comparar dos documentos, se compara dos vectores multidimensionales.

Etiqueta semántica. Este término se utiliza para hacer referencia a un sentido específico de un vocablo ambiguo, tomando en cuenta alguna fuente de información. Por ejemplo, WordNet.

Herramientas lingüísticas. Esta expresión hace referencia a diversos programas o aplicaciones usados en el procesamiento de lenguaje natural, tales como analizadores sintácticos, morfológicos, diccionarios electrónicos, ontologías, entre otros.

Hiperónimo. Un hiperónimo es una palabra cuyo significado incluye al de otra u otras. Por ejemplo, pájaro respecto a jilguero y gorrión

Hipónimo. Un homónimo es una palabra cuyo significado está incluido en el de otra. Po ejemplo, gorrión respecto a pájaro.

Recuperación de información. La recuperación de información es la ciencia encargada de buscar información en archivos de diversos tipos, en meta-datos y en bases de datos textuales, de imágenes o de sonidos. La plataforma sobre la cual es posible realizar dichas búsquedas se extiende desde computadoras de escritorio, redes de computadoras privadas o públicas hasta intranets e internet

Lema. Lema es el vocablo más representativo de un conjunto de palabras que comparten cierta morfología común. Por ejemplo el lema de los vocablos perrito, perraso, perra; es perro.

Lingüística computacional. La lingüística computacional puede considerarse una disciplina de la lingüística aplicada y la inteligencia artificial. Tiene como objetivo la creación e implementación de programas computacionales que permitan la comunicación entre el hombre y la computadora, ya sea mediante texto o voz.

Macro-contexto. El macro-contexto está conformado por palabras de gran contenido semántico (sustantivos, adjetivos y verbos), las cuales co-ocurren con un sentido específico de un vocablo ambiguo, usando varias oraciones como fuente de información.

Meronimia. La meronimia es la relación semántica entre una unidad léxica que denota una *parte* y lo que denota el correspondiente *todo*.

Metonimia. La metonimia consiste en designar algo con el nombre de otra cosa tomando el efecto por la causa o viceversa, el autor por sus obras, el signo por la cosa significada, etc. Por ejemplo, las canas por la vejez, leer a Virgilio, por leer las obras de Virgilio; el laurel por la gloria, etc.

Peso de similitud. El peso de similitud entre dos definiciones es un valor calculado por alguna medida de similitud o relación semántica. Este peso refleja cuan similares o parecidas pueden ser dos palabras, basándose en la definición de cada una.

Polisemia. Polisemia a la capacidad que tiene una sola palabra para expresar muy distintos significados.Pluralidad de significados de una palabra o de cualquier signo lingüístico y de un mensaje, con independencia de la naturaleza de los signos que lo constituyen.

Sentido predominante. El sentido predominante de un vocablo es aquel que generalmente suele ser el más usado por los hablantes de una lengua específica.

Recurso sintáctico. Un recurso sintáctico, en este trabajo, es una base de datos que almacena relaciones de dependencia sintácticas, las cuales pueden ser extraídas desde cualquier corpus de texto.

Sinonimia. La sinonimia es una relación de semejanza de significados entre determinadas palabras (llamadas sinónimos) u oraciones. También consiste en usar voces sinónimas de significación similar para amplificar o reforzar la expresión de un concepto.

Synset. *Synset* es un término usado en WordNet, donde hace referencia a un conjunto de sinónimos, comprendidos por palabras o colocaciones. Dicho conjunto se encuentra conectado con otros *synsets* mediante relaciones jerárquicas existentes en WordNet.

Tesauro. El tesauro es un sistema que organiza el conocimiento basado en conceptos que muestran relaciones entre vocablos. Las relaciones expresadas comúnmente incluyen jerarquía, equivalencia y asociación (o relación). Los tesauros también proporcionan información como sinonimia, antonimia, homonimia, etc.

Tripletas de dependencia. Una tripleta de dependencia sintáctica esta conformada por dos vocablos unidos bajo cierta relación de dependencia. En este trabajo también se le denomina tupla sintáctica.

Troponimia. La troponimia es una relación que se da entre verbos. Ésta considera que las distinciones de *modo* son las más importantes a la hora de diferenciar un hipónimo verbal de su hiperónimo. Esta se encuentra definida en WordNet como *una manera particular de hacer algo*, es decir, como un tipo de implicación léxica.

Vecinos. Es un conjunto conformado por vocablos que mantienen cierta relación semántica con otro en específico. Dichos vocablos son seleccionados comparando diferentes contextos locales, de tal manera que los contextos más parecidos seleccionan a los vecinos.

Bibliografía

[1] Agirre, E. and Martinez, D. (2000). Exploring automatic word sense disambiguation with decision lists and the web. In Proceedings of the Coling 2000 Workshop: Semantic Annotation and Intelligent Annotation, Centre Universitaire, Luxembourg, 2000.

[2] Agirre, E. and Martinez, D. (2001). Knowledge sources for word sense disambiguation. In Matousek, V., Mautner, P., Moucek, R. and Tauser, K. editors, Proceedings of the Fourth International Conference TSD 2001, Plzen, Notes in Computer Science, 1-10, Berlin, 2001.

[3] Agirre, E. and Rigau, G. (1996). Word sense disambiguation using conceptual density. In Proceedings of the 16th International Conference on Computational Linguistics, 16-22, 1996.

[4] Alam, H. *et al* (2002). Extending a Broad-Coverage Parser for a General NLP Toolkit, 2002. 454-460

[5] Atkins, Beryl T. S. (1987). Semantic ID tags: Corpus evidence for dictionary senses. Proceedings of the Third Annual Conference of the UW Center for the New OED, Waterloo, Ontario, 17-36.

[6] Banerjee and Pedersen, T. (2002). An adapted Lesk algorithm for word sense disambiguation using WordNet. In Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, February 2002.

[7] Basili, Roberto; Della Rocca, Michelangelo and Pazienza, Maria Tereza (1997). Towards a bootstrapping framework for corpus semantic tagging. ACL-SIGLEX Workshop Tagging Text with Lexical Semantics: Why, What, and How?, April 4-5, 1997, Washington, D.C., USA, 66-73.

[8] Briscoe, Edward J. (1991). Lexical issues in natural language processing. In Klein, Ewan H. and Veltman, Frank (Eds.) Natural Language and Speech, Proceedings of the Symposium on Natural Language and Speech, 26-27 November 1991, Brussels, Belgium, Springer-Verlag, Berlin, 39-68.

[9] Brown, Gillian and Yule, George (1983). Discourse analysis. Cambridge Textbooks in Linguistics Series, Cambridge University Press, Cambridge, United Kingdom, 1983.

[10] Brown, Peter F.; Della Pietra, Stephen; Della Pietra, Vincent J. and Mercer Robert L. (1991). Word sense disambiguation using statistical methods. Proceedings of the 29th Annual Meeting of Association for Computational Linguistics, Berkeley, California, 264-270.

[11] Brown, Peter F.; Della Pietra, Vincent J.; deSouza, Peter V.; Lai, Jennifer C. and Mercer, Robert L. (1992). Class-based n-gram models of natural language, Computational Linguistics, 18(4), 467-479.

[12] Bryan, Robert M. (1973). Abstract thesauri and graph theory applications to thesaurus research. In Sedelow, Sally Yeates (Ed.), Automated Language Analysis, 1972-3, University of Kansas Press, Lawrence, Kansas, 45-89.

[13] Bryan, Robert M. (1974). Modelling in thesaurus research. In Sedelow, Sally Yeates *et al.* (Ed.), Automated Language Analysis, 1973-4. University of Kansas Press, Lawrence, Kansas, 44-59.

[14] Budanitsky, A. and Hirst, G. (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, June 2001.

[15] Buitelaar, Paul (1997). A lexicon for underspecified semantic tagging. ACL-SIGLEX Workshop Tagging Text with Lexical Semantics: Why, What, and How?, April 4-5, 1997, Washington, D.C., 25-33.

[16] Chomsky, Noam (1957). Syntactic structures, Mouton, The Hague.

[17] Choueka, Yaacov and Lusignan, Serge (1985). Disambiguation by short contexts. Computers and the Humanities, 19, 147-158.

[18] Dagan, Ido and Itai, Alon (1994). Word sense disambiguation using a second language monolingual corpus. Computational Linguistics, 20(4), 563-596.

[19] Dagan, Ido; Marcus, Shaul and Markovitch, Shaul (1993). Contextual word similarity and estimation from sparse data. Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, 22-26 June 1993, Columbus, Ohio.

[20] Dahlgren, Kathleen G. (1988). Naive Semantics for Natural Language Understanding, Kluwer Academic Publishers, Boston. 258 pp.

[21] Fellbaum, C.; Palmer, M.; Trang Dang, H.; Delfs, L. and Wolff, S. (2001). Manual and Automatic Semantic Annotation with WordNet. In Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources: Applications, Customizations, Carnegie Mellon University, Pittsburg, PA, 2001.

[22] Gale, William A.; Church, Kenneth W. and Yarowsky, David (1992a). Using bilingual materials to develop word sense disambiguation methods. Proceedings of the Internacional Conference on Theoretical and Methodological Issues in Machine Translation, 101-112.

[23] Gale, William A.; Church, Kenneth W. and Yarowsky, David (1992c). One sense per discourse. Proceedings of the Speech and Natural Language Workshop, San Francisco, Morgan Kaufmann, 233-37.

[24] Gale, William A.; Church, Kenneth W. and Yarowsky, David (1993). A method for disambiguating word senses in a large corpus, Computers and the Humanities, 26, 415-439.

[25] Gelbukh, Alexander and Bolshakov, Igor (2001). Computational Linguistics, Models, Resources and Applications, 2001.

[26] Grishman, Ralph; MacLeod, Catherine and Meyers, Adam (1994). COMLEX syntax: Building a computational lexicon. Proceedings of the 15th International Conference on Computational Linguistics, COLING'94, 5-9 August 1994, Kyoto, Japan, 268-272.

[27] Harris, Zellig S. (1951). Methods in structural linguistics. The University of Chicago Press, Chicago, xv-384 pp.

[28] Hawkins, P. (1999). DURHAM: A Word Sense Disambiguation System. PhD thesis, Laboratory for Natural Language Engineering, Department of Computer Science, University of Durham, Durham, 1999.

[29] Haynes, S. (2001). Semantic tagging using WordNet examples. In Proceedings of Senseval-2, Second International Workshop on Evaluating Word Sense Disambiguation Systems, 79-82, Toulouse, 2001.

[30] Hearst, Marti A. (1991). Noun homograph disambiguation using local context in large corpora. Proceedings of the 7th Annual Conf. of the University of Waterloo Centre for the New OED and Text Research, Oxford, United Kingdom, 1-19.

[31] Hirst, G. (1987). Semantic Interpretation and the Resolution of Ambiguity. Cambridge: Cambridge University Press.

[32] Hobbs, Jerry R. (1987). World knowledge and word meaning. Proceedings of the Third Workshop on Theoretical Issues in Natural Language Processing, TINLAP-3, Las Cruces, New Mexico, 20-25.

[33] Imbs, Paul (1971). Trésor de la Langue Française, Dictionnaire de la langue du XIXè et du XXè siècles (1989-1960). Editions du Centre National de la Recherche Scientifique, Paris.

[34] Jiang, J. and Conrath D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings on International Conference on Research in Computational Linguistics, Taiwan, 1997.

[35] Johansson, Stig (1980). The LOB corpus of British English texts: presentation and comments. ALLC Journal*,* 1(1), 25-36.

[36] Kaplan, Abraham (1950). An experimental study of ambiguity and context. Mimeographed, 18 pp, November 1950.

[37] Kelly Edward F. and Stone Philip J. (1975). Computer Recognition of English Word Senses, North-Holland, Amsterdam.

[38] Khellmer, G. (1991). A mint of phrases. In Aijmer, K. and Altenburg, B. (eds.) English Corpus Linguistics: Studies in Honour of Jan Svartvik, London: Longman.

[39] Kintsch, Walter and Mross, Ernest F. (1985). Context effects in word identification, Journal of Memory and Language, 24(3), 336-349.

[40] Kucera, Henri and Francis, Winthrop N. (1967). Computational Analysis of Present-Day American English*,* Brown University Press, Providence.

[41] Leacock, C. and Chodorow M. (1998). Combining local context and WordNet similarity for word sense identification, In C. Fellbaum editor, WordNet: An electronic lexical database, pages 265–283. MIT Press, 1998.

[42] Leacock, C., Chodorow, M. and Miller, G. (1998). Using corpus statistics and WordNet relations for sense identification, Computational Linguistics, 24(1):147-165, 1998.

[43] Leacock, Claudia; Towell, Geoffrey and Voorhees, Ellen (1993). Corpus-based statistical sense resolution. Proceedings of the ARPA Human Language Technology Worskshop, San Francisco, Morgan Kaufman.

[44] Leacock, Claudia; Towell, Geoffrey; and Voorhees, Ellen M. (1996). Towards building contextual representations of word senses using statistical models. In Boguraev, Branimir and Pustejovsky, James (Eds.), Corpus Processing for Lexical Acquisition, MIT Press, Cambridge, Massachusetts, 97-113.

[45] Lenat, Douglas B. and Guha, Ramanathan V., (1990). Building large knowledge-based systems. Addison-Wesley, Reading, Massachusetts.

[46] Lesk, Michael (1986). Automated Sense Disambiguation Using Machine-readable Dictionaries: How to Tell a Pine one from an Ice Cream Cone. Proceedings of the 1986 SIGDOC Conference, Toronto, Canada, June 1986, 24-26.

[47] Li, X., Szpakowics, S. and Matwin, S. (1995). A WordNet-based algorithm for word sense disambiguation. In Proceedings of the 14th International Joint conference on Artificial Intelligence, 1368-1374, Montreal, 1995.

[48] Lin, D. (1993). Principle-based parsing without overgeneration. In 31th Annual Meeting of the Association for Computational Linguistics (ACL 1993), 112-120, Columbus, 1993.

[49] Lin, D. (1997). Using syntactic dependency as a local context to resolve word sense ambiguity. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, pages 64–71, Madrid, July 1997.

[50] Lin, D. (1998). Dependency-based Evaluation of MINIPAR. In *Workshop on the Evaluation of Parsing Systems*, Granada, Spain, May, 1998.

[51] Lin, D. (1998). Automatic retrieval and clustering of similar words. Proceedings of C I G C -. pp. 768-774. Montreal, Canada.

[52] Luk, Alpha K. (1995). Statistical sense disambiguation with relatively small corpora using dictionary definitions. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, Cambridge Massachusetts, 181-188.

[53] Masterman, Margaret (1957). The thesaurus in syntax and semantics, Mechanical Translation, 4, 1-2.

[54] McCarthy, D., Koeling, R., Weeds, J. and Carroll, J. (2004). Finding predominant senses in untagged text. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, 2004.

[55] Mel'čuk, Igor A. (1987). Dependency syntax; theory and practice. State University of New York Press, Albny.

[56] Mihalcea, R. and Moldovan, D. (1999). A method for word sense disambiguation of unrestricted text. In 37th Annual Meeting of the Association for Computational Linguistics (ACL 1999), 152-158, Maryland, 1999.

[57] Miller, G., Leacock, C., Tengi, R., Bunker, R. and Miller, K. (1990). Five papers on WordNet. Special Issue of International Journal of Lexicography, 1990, 3(4).

[58] Miller, George A.; Beckwith, Richard T.; Fellbaum, Christiane D.; Gross, Derek and Miller, Katherine J. (1990). WordNet: An on-line lexical database. International Journal of Lexicography, 3(4), 235-244.

[59] Patrick, Archibald B. (1985). An exploration of abstract thesaurus instantiation. M. Sc. thesis, University of Kansas, Lawrence, Kansas.

[60] Patwardhan, S., Banerjee, S., and Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, 2003.

[61] Pedersen, T.; Patwardhan, S. and Michelizzi (2004).WordNet::Similarity - Measuring the Relatedness of Concepts. Appears in the Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04), May 3-5, 2004, Boston, MA (Demonstration System)

[62] Pustejovsky (1995). The Generative Lexicon. The MIT Press, Cambridge, Massachusetts.

[63] Rennie, J. (2000). WordNet::QueryData: a Perl module for accessing the WordNet database. www.ai.mit.edu/people/jrennie/WordNet.

[64] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, August 1995.

[65] Resnik, P. (1998). WordNet and class–based probabilities. In C. Fellbaum, editor, WordNet: An electronic lexical database, pages 239–263. MIT Press, 1998.

[66] Resnik, Philip (1992). WordNet and distributional analysis: a class-based approach to statistical discovery. I Workshop on Statistically-based Natural Language Processing Techniques, San Jose, California, 48-56.

[67] Richardson, R. and Smeaton, Alan F. (1994). Automatic word sense disambiguation in a KBIR application. Working paper CA-0595, School of Computer Applications, Dublin City University, Dublin, Ireland, 1994.

[68] Schank, Roger C. and Abelson, Robert P. (1977). Scripts, Plans, Goals and Understanding, Lawrence Erlbaum, Hillsdale, New Jersey.

[69] Schütze, Hinrich (1992). Dimensions of meaning. Proceedings of Supercomputing'92. IEEE Computer Society Press, Los Alamitos, California. 787-796.

[70] Schütze, Hinrich (1993). Word space. In Hanson, Stephen J.; Cowan, Jack D. and Giles, C. Lee (Eds.) Advances in Neural Information Processing Systems 5, Morgan Kauffman, San Mateo, California, 5, 895-902.

[71] Sedelow, Sally Yeates and Sedelow, Walter. A. Jr. (1969). Categories and procedures for content analysis in the humanities. In Gerbner, George; Holsti, Ole; Krippendorf, Klaus; Paisley, William J.; and Stone, Philip J. (Eds.), The Analysis of Communication Content, John Wiley & Sons, New York,487-499.

[72] Slator, Brian M. (1992). Sense and preference. Computer and Mathematics with Applications, 23(6/9), 391-402.

[73] Sparck Jones, Karen (1964). Synonymy and semantic classification. Ph. D. thesis, University of Cambridge, Cambridge, England.

[74] Stevenson, M. (2003). Word Sense Disambiguation: The case for Combinations of Knowledge Sources, (2003).

[75] Stevenson, M. and Wilks, Y. (2001). The interaction of knowledge sources in word sense disambiguation. Computational Linguistics, 2001, 27(3):321-349.

[76] Voorhes, Ellen M. (1993), Using WordNet to disambiguate word senses for text retrieval. Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 27 June-1 July 1993, Pittsburgh, Pennsylvania,171-180, 1993.

[77] Voorhees, Ellen M., Claudia Leacock and Geoffrey Towell (1995). Learning context to disambiguate word senses. In Thomas Petsche, Stephen José Hanson and Jude Shavlik eds., Computational Learning Theory and Natural Learning Systems. MIT Press, Cambridge, Massachusetts.

[78] Weiss, S. (1973). Learning to disambiguate. Information Storage and Retrieval, 9, 33-41.

[79] Wilks, Yorick A. (1973). An artificial intelligence approach to machine translation. In Schank, Roger and Colby, Kenneth (Eds.). Computer Models of Thought and Language, San Francisco: W H Freeman, 114-151.

[80] Wilks, Yorick A. (1975b). Preference semantics. In Keenan, E. L. III (Ed.), Formal Semantics of Natural Language, Cambridge University Press, 329-348.

[81] Wilks, Yorick A.; Fass, Dan (1990). Preference semantics: A family history. Report MCCS-90-194, Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico.

[82] Yarowsky, David (1992). Word sense disambiguation using statistical models of Roget's categories trained on large corpora. Proceedings of the 14th International Conference on Computational Linguistics, COLING'92, 23-28 August, Nantes, France, 454-460.

[83] Yarowsky, David (1993). One sense per collocation. Proceeding of ARPA Human Language Technology Workshop, Princeton, New Jersey, 266-271.

[84] Yarowsky, David (1994a). Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French. Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 88-95.

[85] Yarowsky, David (1994b). A comparison of corpus-based techniques for restoring accents in Spanish and French text. Proceedings of the 2nd Annual Workshop on Very Large Text Corpora. Las Cruces, 19-32.