Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Maestría en Ciencias de la Computación

Creación de una base de datos léxica para la desambiguación de palabras

Tesis que para obtener el grado de
Maestro en Ciencias de la Computación
presenta:

Lic. Sofía Paniagua Rivera

Director: Dr. Héctor Jiménez Salazar

Agosto 2005

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

Agradezco a **Dios** por la familia que me dio y por
la oportunidad de seguir adelante.

Agradezco a mi **hijo David** por ser la fuerza que me impulsa
y la alegría en todo momento.

Agradezco a mi **esposo** por todo el apoyo dado incondicionalmente
para culminar una meta más en mi vida.
Pero sobre todo por el amor que me demuestra con cada cosa
que hace por mí.

Agradezco a mi **mamá** por siempre estar conmigo y no dejar
que caiga, y por que por ella soy lo que soy.

Agradezco a mis **hermanos** por ser alegría en mi vida.

Agradezco a mi asesor **Dr. Héctor**, por quien tengo una gran admiración,
por su dedicación, apoyo y paciencia para que lograra terminar este proyecto.

Agradezco a mis sinodales **Dra. Darnes** y **Dr. Sidorov**
por dedicarme tiempo, para terminar este proyecto.

Agradezco a mi amiga **Yatzu** por haberse cruzado en mi camino.

Agradezco a VIEP por su apoyo para difundir este trabajo de
investigación.

Agradezco a CONACYT por otorgarme la beca para estudios de
Posgrado.

Introducción

Una base de datos léxica es una base de conocimiento a gran escala normalmente construida manualmente [4]. Una base de datos léxica también se considera como un diccionario destinado por lo general a desambiguar el sentido de las palabras y que posee términos relacionados con sus características. La desambiguación del sentido de las palabras (WSD) ha sido un tema de investigación en los últimos años y una tarea más en el procesamiento del lenguaje natural (PLN), todo el trabajo de desambiguación involucra empatamiento del contexto de la instancia de una palabra a ser desambiguada con información de una fuente de conocimiento externa, o información acerca de contextos de instancias anteriormente desambiguadas de la palabra derivada de un *corpus*. Es importante destacar que aunque existen recursos y técnicas para el proceso de WSD en otros idiomas (por ejemplo el Inglés), no es un caso igual para el lenguaje español (Se cree que el poco desarrollo se debe principalmente a la riqueza morfológica del lenguaje). Debido a lo anterior, en este trabajo de tesis se incide en la construcción automática de un recurso de este tipo, que permita desambiguar palabras del Español. Dicho recurso es en esencia una base de datos léxica que está conformada por las palabras y sus rasgos, los cuales se obtienen a partir de los contextos agrupados de las mismas palabras.

Se espera así, que la base de datos léxica que se construya sea de gran ayuda para resolver las tareas que tiene el PLN en el proceso de desambiguación apoyando a otras herramientas que necesitan de esta tarea para obtener un mayor beneficio en el proceso de su búsqueda, y así también se espera apoyar en mejorar el desempeño de los sistemas de recuperación en Internet. Particularmente, la base de datos léxica permite la desambiguación de un conjunto de palabras para el lenguaje Español.

Para terminar, la herramienta desarrollada impactará en diversos campos del PLN, principalmente por ser una técnica considerada como un sistema no

supervisado que permite construir una base de datos léxica, a partir de un conjunto de documentos sin formato, garantizando de esta manera su independencia del dominio, lo cual es precisamente algunas de las debilidades de los sistemas construidos hasta ahora.

El contenido de este documento se encuentra distribuido de la siguiente manera: el capítulo I presenta un marco general sobre la desambiguación del sentido de una palabra. El capítulo II presenta la manera en que se puede utilizar una base de datos léxica en el proceso de desambiguación del sentido de una palabra. Las técnicas ocupadas para la creación de la base de datos léxica se muestran en el capítulo III. Las pruebas realizadas son expuestas en el capítulo IV. Finalmente se reportan las conclusiones de este trabajo.

Capítulo 1

Desambiguación del Sentido de las Palabras

Uno de los primeros problemas encontrados por cualquier sistema de procesamiento del lenguaje natural es el de la ambigüedad léxica, ya sea ésta sintáctica o semántica. La resolución de la ambigüedad sintáctica de una palabra, ha sido solucionada en el procesamiento del lenguaje mediante los etiquetadores de partes del discurso (*part-of-speech taggers*), los cuales predicen con altos grados de precisión, la categoría sintáctica de las palabras un texto (ver por ejemplo [17]). Por otro lado, el problema de resolver la ambigüedad semántica es conocido generalmente como desambiguación del sentido de una palabra o *word sense disambiguation* (WSD por sus iniciales en inglés). Y este último problema, ha probado ser más difícil que la desambiguación sintáctica.

El problema fundamental, es que las palabras tienen a menudo más de un significado, algunas veces bastante o completamente diferentes. El significado de una palabra en un uso particular puede solamente ser determinado mediante la examinación de su contexto. Esto es, comunmente, una tarea trivial para los seres humanos. Tomese por ejemplo las siguientes dos oraciones, cada una con diferentes sentidos de la palabra banco:

- El bote quedó atascado en un banco de arena.
- La camioneta se estacionó al lado del banco y tres hombres enmascarados salieron de ella.

Es fácil reconocer que en la primera oración, banco se refiere a la orilla

de una rivera, mientras que en la segunda oración, se refiere a un edificio o institución bancaria. Sin embargo, concluir lo anterior ha probado ser difícil para una computadora, al grado que algunas personas creen que este problema podría nunca ser resuelto.

Bar-Hillel [18] es una persona que se hizo famosa por proclamar la siguiente frase: "*sense ambiguity could not be resolved by electronic computer either current or imaginable*".

Él usó el siguiente ejemplo, que contiene la palabra *pen* en Inglés, la cual es una palabra con varios sentidos.

> *Little John was looking for his toy box.*
> *Finally he found it.*
> *The box was in the pen.*
> *John was very happy.*

Bar-Hillel argumentó que aún si la palabra *pen* tuviese únicamente dos sentidos, "implemento para escribir" y "compartimiento", la computadora no tendría manera de decidir entre estos dos sentidos. Un análisis del ejemplo muestra que éste es un caso en donde las restricciones de selección fallan al momento de desambiguar la palabra *pen*, ya que ambos sentidos potenciales indican objetos físicos en los cuales es posible colocar cosas (tal vez un poco difícil para el primer caso), la proposición en (*in*) puede aplicarse para ambos casos. La desambiguación en este caso debería hacer uso del conocimiento del mundo, los tamaños relativos y los usos de *pen* como implemento de escritura y como compartimiento.

Sin embargo, la situación no es tan mala como lo plantea Bar-Hillel, ya que existen muchos avances en la tarea de desambiguación del sentido de las palabras y realmente se está en este momento en una etapa en donde la ambigüedad léxica en un texto puede ser resuelta con un grado razonable de precisión.

1.1. Antecedentes

Las bases de datos léxicas orientadas a la desambiguación conforman un volumen de información, básicamente de términos y relaciones de significado. El desarrollo de este recurso ha pasado por diversas fases. En la primera fase hubo necesidad de su uso en la traducción automática (*MT*), en donde se enfocó la traducción de textos técnicos y de un mismo dominio. Después MT fue

dedicada al desarrollo de diccionarios especializados [5]. Posteriormente aparecen aplicaciones en el mapeo de palabras de cualquier lenguaje a una representación semántica/conceptual común. La primer máquina implementada con este concepto fue la construida para el tesoro de Roget [6]. El sistema de Masterman(1962) utiliza una red semántica para derivar la representación de oraciones de conceptos de un lenguaje fuente [7]. Wilks(1990) usó las primitivas de Masterman para el lenguaje natural y es uno de los primeros diseños para el tratamiento de la desambiguación de sentidos [8], [9], [10]. Es fácil observar que la ambigüedad está presente en el lenguaje natural (LN), y aunque para los humanos es relativamente sencillo identificar a qué se refiere la palabra ambigua tomando como referencia su contexto, no lo es tal para los sistemas de recuperación de información. Estos últimos, obtienen documentos que presumiblemente son de interés para el usuario, de acuerdo a su consulta, sin embargo, muchas veces obtienen documentos que no tienen nada que ver con lo que se estaba esperando, debido precisamente a la ambigüedad de las palabras presentadas en la consulta. La pérdida de tiempo al recuperar esta información puede, incluso, ocasionar perdidas económicas en empresas que requieren información para la toma de decisiones, por lo que se ha experimentado en la aplicación de las bases de datos léxicas en recuperación de información (RI). Otras aplicaciones se enfocan a la creación de diccionarios, por ejemplo, Lesk(1986) [11]. En algunos casos, se tiene un conjunto de palabras y su definición (diccionario de términos), en otros casos se tiene además la relación entre cada uno de los términos (por ejemplo Wordnet), existen otras bases de datos multilingües como es el caso de EuroWordNet y otras más almacenan información acerca de la información morfológica sobre cada término. Se generaron nuevas ideas, haciendo algotirmos a partir de los algoritmos de Wilks y Lesk [64].

1.2. Aplicaciones de WSD

En el procesamiento del lenguaje se puede distinguir entre tareas finales e intermediarias: las tareas finales son aquellas que se llevan a cabo para sus propios ejemplos de uso, básicamente tareas finales son, por ejemplo, traducción automática, generación de resúmenes personalizados y extracción de información; las tareas intermediarias son llevadas a cabo para ayudar a las tareas finales, algunos ejemplos son etiquetamiento de partes del discurso, *parsing*, identificación de raíces morfológicas y desambiguación del sentidos de una palabra, éstas son

tareas en las cuales usualmente no prestamos demasiado interés.

El uso de la tareas intermediarias puede ser explorado buscando su utilidad en algunas de las tareas finales. A continuación se examinarán tres tareas en las que tradicionalmente se ha asumido que la tarea de desambiguación del sentido de una palabra podría ayudar: recuperación de información, traducción automática y *parsing.*

1.2.1. Recuperación de Información

La idea de que la tarea de desambiguación del sentido de una palabra podría ayudar en la recuperación de información, proviene de asumir que si un sistema de recuperación indexa documentos mediante los sentidos de sus palabras y si es posible determinar los sentidos correctos de una consulta, entonces los documentos no relevantes a la consulta (por contener diferentes sentidos de las palabras) no serían enviados como respuesta. Strzalkowski [19] encontró evidencia de que el NLP puede ayudar en la recuperación de información. Sin embargo, otros investigadores han encontrado que WSD ayudan muy poco a mejorar el funcionamiento de los SRI. Krovetz y Croft [20], [21] desambiguaron manualmente un *corpus* estándar de prueba de IR y encontraron que una máquina con desambiguación de sentidos perfecta podría mejorar el funcionamiento del SRI incrementando en solamente un 2 %. Sanderson [22] ejecutó experimentos similares, donde él introdujo de manera artificial ambigüedad en la colección, y encontró que el funcionamiento se incrementó únicamente para consultas con pocas palabras (menos de 5 palabras). La razón de este comportamiento es que los algoritmos estadísticos a menudo usados en recuperación de información son similares a las aproximaciones para WSD y las consultas con muchas palabras realmente ayudan a desambiguar entre sí con respecto a los documentos. Sanderson también descubrió que el funcionamiento de los SRI es insensitivo a la ambigüedad pero muy sensitivo a la desambiguación errónea.

1.2.2. Traducción Automática

En contraste, los investigadores del tema de traducción automática han argumentado consistentemente que los procedimientos efectivos de desambiguación del sentido de una palabra podrían revolucionar su campo de estudio. Hutchins y Sommers [23] han apuntado que actualmente hay dos tipos de ambigüedad léxica semántica con los cuales los sistemas de traducción deben contender: hay ambigüedad en el lenguaje fuente cuando el significado de una palabra no

es inmediatamente aparente y también hay ambigüedad en el lenguaje destino cuando una palabra no es ambigua en el lenguaje fuente pero tiene dos posibles traducciones. Brown et. al. [24] construyeron un algoritmo para WSD para un sistema de traducción Inglés-Francés. Este experimento realizó tanta desambiguación como fue necesaria para encontrar la palabra correcta en el lenguaje destino (es decir, el algoritmo solamente resolvió el primer tipo de ambigüedad en el área de traducción automática). Brown encontró que el 45 % de las traducciones fue aceptable cuando se usó la máquina de desambiguación, mientras que el 37 % fue aceptable cuando dicha máquina no se usó. Esta es una prueba empírica de que WSD es una tarea intermediaria útil para el área de traducción automática. Actualmente se afirma que WSD ocurre implícita en los algoritmos de traducción automática, y la incompatibilidad de los sistemas de WSD con la traducción efectúa una mejora poco significativa [76].

1.2.3. Parsing

Parsing o análisis sintáctico, es una tarea intermediaria usada en muchas aplicaciones del procesamiento del lenguaje y obtener un *parsing* preciso ha sido por mucho tiempo una meta en NLP. Parece ser que si fuera conocida la semántica de cada elemento léxico entonces ésta podría ayudar al *parser* en la construcción de las estructuras de frase para una oración. Considérese la oración "El niño observó al perro con el telescopio", la cual es a menudo usada como un ejemplo de un problema para adjuntar la frase preposicional. Un *parser* podría adjuntar correctamente la frase proposicional al verbo *observó* mediante el uso de conocimiento semántico del mundo, y pareciera ser que las etiquetas semánticas podrían proporcionar parte de este conocimiento. Desafortunadamente, al parecer existen pocos experimentos llevados a cabo en el uso de WSD para la tarea de *parsing*.

Se puede ver entonces que WSD podría ser benéfica para bastantes tareas importantes de NLP, aunque podría no ser tan útil como muchos investigadores han pensado. Sin embargo, la verdadera utilidad será puesta a prueba cuando existan adecuados algoritmos de WSD; sólo hasta entonces estaremos en posición de experimentar si éstos incrementan o no la efectividad de los resultados; lo cual significa que se tengan recupos lingüísticos extensos para hacer pruebas en colecciones representativas; como las que se tienen en la *web*.

1.3. Enfoques usados para WSD

Es útil distinguir entre diferentes enfoques usados en el problema de WSD. En general, se puede categorizar la mayoría de los enfoques a este problema en alguna de las siguientes tres estrategias: basadas en conocimiento, basadas en *corpus*, o híbridas. A continuación se presenta cada una de estas tres estrategias.

1.3.1. Enfoque basado en conocimiento

Bajo este enfoque, la desambiguación es llevada a cabo mediante el uso de un lexicón explícito o base de conocimiento. El lexicón puede ser un diccionario leíble por computadora (*machine readable dictionary-MRD*), un *thesaurus* o un lexicón construido manualmente (*hand-crafted*). Este es uno de los enfoques más populares a WSD y entre otros, existen algunos trabajos que han sido llevados a cabo mediante el uso de fuentes léxicas de conocimiento tales como WordNet [25], Resnik [26], Richardson [27], Sussna [28], Voorhees [29], LDOCE [30], J. Guthrie [31], y el *Thesaurus* Internacional de Roget [32].

La información en estos recursos ha sido usada de diferentes maneras, por ejemplo, Wilks y Stevenson [33], Harley y Glennon [34] y McRoy [35], todos usaron grandes lexicones (generalmente *MRDs*) y la información asociada con los sentidos (tales como etiquetamiento de POS, guías de tópicos y preferencias seleccionales) para indicar el sentido correcto. Otro enfoque es tratar el texto como un bolsa de palabras desordenada donde las medidas de similitud son calculadas buscando en la similitud semántica entre todas las palabras en una determinada ventana, sin importar su posición, tal como fue usado por Yarowsky [32].

1.3.2. Enfoque basado en Corpus

Este enfoque intenta desambiguar palabras mediante el uso de información obtenida por entrenamiento en algún *corpus*, en vez de tomarla directamente de una fuente explícita de conocimiento. Este entrenamiento puede ser llevado a cabo ya sea en un *corpus* desambiguado o en uno plano. Un *corpus* desambiguado es aquel donde la semántica de cada elemento léxico con polisemia es marcado y un *corpus* plano es el que no tiene estas marcas.

Corpora desambiguado

Este conjunto de técnicas requiere de un *corpus* entrenado que esté totalmente desambiguado. En general, se aplica algún algoritmo de aprendizaje automático para extraer ciertas características del *corpus* y usarlas para conformar una representación de los sentidos. Esta representación puede entonces ser aplicada a nuevas instancias, con la finalidad de desambiguarlas. Diversos investigadores han hecho uso de diferentes conjuntos de características, por ejemplo P. Brown [24] usó colocaciones locales tales como primer sustantivo a la izquierda y derecha, segunda palabra a la izquierda/derecha, etc. Sin embargo, un conjunto de características más común usado por Gale [37] es tomar todas aquellas palabras en una ventana alrededor de la palabra ambigua, tratando el contexto como una bolsa de palabras.

Otra aproximación es usar modelos ocultos de Markov (HMM), los cuales han demostrado ser bastante buenos en etiquetamiento de POS. Observando, por supuesto, que el etiquetamiento semántico es un problema mucho más difícil que el etiquetamiento POS, Segond [36] decidió ejecutar un experimento para ver que tan bien pueden ser desambiguadas las palabras usando técnicas que han demostrado ser efectivas en el etiquetamiento POS.

El problema general con estos métodos es su dependencia en *corpus* desambiguados, los cuales son caros y difíciles de obtener. Esto ha derivado que muchos de estos algoritmos se ejecuten sobre números muy pequeños de palabras diferentes, a menudo sobre 10.

Corpora artificial

Una consecuencia de la dificultad de obtener corpora etiquetado por sentidos, ha llevado a muchos investigadores a encontrar maneras creativas de construir corpora artificial que contenga alguna forma de etiquetamiento semántico.

El primer tipo de *corpus* artificial que ha sido usado extensamente es el *corpus* paralelo. Un *corpus* bilingüe consiste de dos corpora que contienen el mismo texto en diferentes idiomas (por ejemplo uno puede ser la traducción de otro , o ambos pueden ser producidos por organizaciones como las Naciones Unidas, quienes continuamente transcriben reuniones en diferentes idiomas). La alineación de oraciones es el proceso de tomar un *corpus* y empatar las oraciones que son traducciones de cada una y existen bastantes algoritmos para llevar a cabo esta tarea con un alto grado de éxito (por ejemplo [38], [39]). Un hábeas bilingüe que ha sido alineado viene a ser un hábeas paralelo alineado. Este es

un recurso interesante dado que consiste de muchos ejemplos de oraciones y sus traducciones. Estos corpora han sido usados en WSD (ver [24] y [39]) tomando las palabras cuyos sentidos se traducen de manera diferente a través de los idiomas. Estos últimos usaron las memorias del Parlamento Canadiense que son publicados en Frances e Inglés.

Hay dos maneras de crear corpora etiquetado artificialmente por sentidos. La primer manera es desambiguando las palabras por su significado, tal como pasa en el caso de corpora paralelo; la otra aproximación es adicionando ambigüedad al *corpus* y hacer que un algoritmo trate de resolver esta ambigüedad para regresar al *corpus* original. Yarowsky [40] usó este último método, creando un *corpus* que contenía pseudo-palabras. Estas son creadas eligiendo dos palabras (por ejemplo "cocodrilo y zapatos") y reemplazando cada ocurrencia con su concatenación ("cocodrilo/zapatos").

1.3.3. Corpora plano

A menudo es difícil obtener recursos léxicos apropiados (especialmente para textos de un sublenguaje especializado), y hemos ya notado la dificultad en obtener texto desambiguado para desambiguación supervisada. Esta carencia de recursos ha llevado a muchos investigadores a explorar el uso de corpora plano o no etiquetado, para ejecutar desambiguación no supervisada. Debería notarse que la desambiguación no supervisada no puede etiquetar términos específicos como una referencia a un concepto particular: ya que esto requeriría más información de la que está disponible. Lo que la desambiguación no supervisada puede hacer es la discriminación del sentido de una palabra, de esta manera, esta técnica agrupa las instancias de una palabra en categorías distintas sin etiquetar estas categorías a partir de un lexicón (tal como los números de sentidos de LDOCE o los synsets de WordNet).

Un ejemplo de esto es la técnica de empatamiento dinámico [41] que examina todas las instancias de un término dado en un *corpus* y compara los contextos en los cuales ocurre para palabras comunes y patrones sintácticos. Se forma así una matriz de similitud, la cual es sujeta de análisis de agrupamiento para determinar grupos de instancias de términos relacionadas semánticamente.

Otro ejemplo es el trabajo de Pedersen [42] quien compara tres diferentes algoritmos de entrenamiento no supervisado sobre 13 diferentes palabras. Cada algoritmo fue entrenado en texto que fue etiquetado ya sea con los sentidos de WordNet o LDOCE para la palabra; aunque el algoritmo no tenía acceso a los

verdaderos sentidos, sí conocía el número de sentidos de cada palabra, por lo que partió las instancias de cada palabra en el número apropiado de grupos. Estos grupos fueron entonces mapeados al sentido más cercano del lexicón apropiado. Desafortunadamente, los resultados no son muy alentadores, ya que Pedersen reporta un 65-66 % de desambiguación correcta, dependiendo del algoritmo de aprendizaje usado. Este resultado debería ser comparado con el hecho de que, en el *corpus* que Pedersen utilizó el 73 % de las instancias podría haber sido clasificado correctamente mediante la selección del sentido más frecuente.

1.3.4. Enfoque híbrido

Este enfoque no puede ser clasificado exclusivamente como basado en *corpus* o en conocimiento, sino más bien como la combinación de ambos. Un buen ejemplo de esto es el sistema de Luk [49], quien usa las definiciones textuales de los sentidos de un MRD (LDOCE) para identificar las relaciones entre los sentidos. A partir de esto él usa un *corpus* para calcular los valores de información mutua entre los sentidos relacionados para descubrir el más útil. Esto permitió a Luk producir un sistema que usa la información de recursos léxicos como una manera de reducir la cantidad de texto necesario en un *corpus* de entrenamiento.

Otro ejemplo de esta aproximación es el algoritmo no supervisado de Yarowsky [43]. Este toma un pequeño número de definiciones de los sentidos de alguna palabra como semilla (las semillas podrían ser synsets de WordNet o definiciones de algún otro lexicón) y usa éstas para clasificar los casos obvios en un *corpus*. Las listas de decisión [44] son entonces usadas para hacer generalizaciones basadas en las instancias de los *corpus* clasificadas y estas listas son entonces re-aplicadas al *corpus* para clasificar más instancias. El aprendizaje prosigue de esta manera hasta que todas las instancias del *corpus* son clasificadas. Yarowsky reporta que el sistema clasifica correctamente los sentidos el 96 % del tiempo.

En este trabajo de tesis se ha optado por experimentar con *corpus* planos y algoritmos de desambiguación no supervisados y contribuir a los enfoques basados en *corpus* e híbridos. Ya que es necesario incidir en el enfoque de *corpus* plano, latente en el PLN, cuando nos centramos en los diversos sublenguajes.

Capítulo 2

WordNet y WSD

La desambiguación del sentido de una palabra (WSD), es sin duda, uno de los problemas más importantes en el área de Procesamiento de Lenguaje Natural (PLN). Desde el primer concurso de SENSEVAL en 1998, este problema ha sido tratado de manera sistemática [59]. SENSEVAL proporciona una colección estándar de información para comparar diversos algoritmos de WSD. Este problema consiste en el reconocimiento del sentido en una palabra ambigua. Por ejemplo, dada la palabra "*interés*" en la oración "*un interés en la búsqueda filosófica*", la meta es determinar para esta oración específica, que *interés* expresa "*exitamiento mental*", evitando la posibilidad de considerar que su significado podría ser financiero o comercial (cargo por prestamo monetario). La solución a este problema podría representar un avance muy importante en sistemas que usan palabras u oraciones en Lenguaje Natural. Por ejemplo, robots buscadores de información, y otros vastos volumenes de información tales como Google, Altavista y demás.

Hata ahora, los algoritmos para WSD han sido categorizados como supervisados y no supervisados. Basicamente, los algoritmos supervisados para WSD aplican una fase de entrenamiento, usando un conjunto ejemplos positivos y negativos (oraciones resueltas), en este caso, oraciones etiquetadas con el sentido correcto para la palabra ambigua. Los algoritmos no supervisados, no deberían usar algún tipo de información extra. En la literatura no existe un acuerdo de lo que deben ser los algoritmos no supervisados en WSD. En particular en etse trabajo de tesis se aplica el término no supervisado a las técnicas que no usan fuentes de conocimiento externas. Ciertamente, los algoritmos supervisados para WSD muestran un mejor desempeño que algunos no supervisados, pero

para su ejecución se requiere de un conjunto de entrenamiento, restringiendo su uso a un dominio específico. En cambio los algoritmos no supervisados para WSD pueden ser usados para resolver problemas en cualquier dominio.

Más aún en 1997, H.T.Ng [55] habló sobre la necesidad de enfocarse de una manera diferente a la obtención de ejemplos de entrenamiento con la finalidad de ser capaz de construir algoritmos para WSD, dado que el etiquetamiento manual de un conjunto de entrenamiento lo suficientemente grande para aplicaciones normales de WSD podría llevar más de 15 años. En los últimos años ha habido importantes avances en WSD. El enrequecimiento de corpora en Internet para acumular suficientes ejemplos de contextos que contienen palabras ambiguas, y el uso de otros recursos, como diccionarios leíbles por computadora (MRD), ha llevado a compilar recursos para obtener mejores resultados (Rada Milhacea reporta alrededor de un 63 % de precisión para este tipo de métodos [54]). Notablemente, los resultados en SENSEVAL-2 estuvieron 14 % abajo con respecto a SENSEVAL-1 (para la tarea en el idioma Inglés), aún cuando se usó la misma metodología de evaluación y muchos de los sistemas fueron versiones mejoradas de los mismos sistemas que participaron en SENSEVAL-1. Esto puede ser visto como una evidencia de que las distinciones de sentidos de WordNet no estan bien enfocadas, sin embargo se necesita más investigación para confirmar esto [56].

La calidad de la ejecución de los algoritmos es una meta que demanda la conformación de fuentes léxicas para WSD en dominios específicos; lo cual es algo que WordNet no cumple. Ciertamente las bases de datos léxicas de propósito general (LDB) y los conjuntos de entrenamiento construidos para WSD podrían ayudar en esta tarea, sin embargo, el reto real es tratar con dominios específicos donde la disponibilidad de recursos es casi nula.

Así, el problema de WSD genera una fuerte demanda en la elaboración de varios recursos léxicos, tales como bases de datos léxicas, no solamente para construir colecciones de ejemplos para la fase entrenamiento de los algoritmos, sino también, para la obtención de diccionarios especializados [51], [52] que son útiles para diferentes aplicaciones.

El sistema de representación léxica WordNet merece especial atención por varias razones. En primer lugar, aunque su implementación y objetivos son muy diferentes a los del trabajo de tesis aquí presentado, ofrece numerosos puntos de convergencia con el proyecto que se prentende desarrollar en cuanto a fundamentos teóricos. Además que fue uno de los primeros intentos serios de desarrollar un lexicón multipropósito en forma de aplicación informática (ver en

Miller [12] [13] y Miller et. al [14]). Por lo que se explicará brevemente.

2.1. WordNet

2.1.1. Generalidades

WordNet es un sistema electrónico de referencia léxica, desarrollado en forma de base de datos léxica. El diseño de WordNet está en consonancia con teorías psicolingüísticas relativas a la organización de la información léxica en la mente del hablante [12], y ha servido en los últimos años para apoyar la construcción de lexicones computacionales de gran envergadura, y relacionados al concepto de reutilización.

Los objetivos primordiales de WordNet son dos:

- La validación de las teorías psicolingüísticas sobre organización léxica.
- Su previsible utilización en diversas aplicaciones que requieran acceso a información léxica.

Las diferencias con un diccionario tradicional son obvias: WordNet divide el lexicón en cinco categorías: nombres, verbos, adjetivos, adverbios y elementos funcionales. Aunque el precio que ha tenido que pagar es la considerable cantidad de información redundante, facilita enormemente el análisis de las diferencias de organización semántica que existen entre esas cinco categorías sintácticas, además, se puede buscar la forma más adecuada para cada una de ellas por separado, por ejemplo, ver Pinto *et al.* [50].

Miller [14] argumenta que cualquier diccionario impreso puede ser reducido a la proyección de los significados sobre las formas (entradas léxicas), y esto a su vez ser reducido a una matriz. Propone, por tanto, la matriz de vocabulario, donde las columnas de una matriz contendrán todas las palabras (formas léxicas *"word form*) de un idioma, mientras que las filas contendrán todos los significados (*"word meaning"*). La matriz dará acceso a la información de dos maneras: se podrá acceder a una columna, de esta forma se obtendrían todos los sentidos que una palabra puede tener en diversos contextos. También se podría acceder por una fila, de este modo se obtendrían todas las maneras posibles de expresar un determinado concepto. En el caso de que haya dos entradas en la misma columna, la forma léxica es polisémica; si hay dos entradas en la misma fila, las dos formas léxicas son sinónimas. Así, la matriz de vocabulario contempla dos de los principales problemas de la semántica léxica, la polisemia y la sinonimia.

La respuesta que WordNet propone para la representación de los conceptos, está basada en la matriz de vocabulario y se les denomina *"synonym sets"* abreviado *"synset"*, y no es más que el resultado de cruzar una fila de la matriz de un lado a otro y asignar un número arbitrario al conjunto de palabras obtenido. Este número actua a modo de identificador del concepto abstracto representado por el conjunto de elementos léxicos que lo designan. Los *"synonym sets"* no explican lo que son los conceptos, simplemente "significan" que un determinado concepto existe. Este sistema obviamente conlleva altos niveles de redundancia en cuanto representación se refiere. La sinonimia es por tanto la relación léxica primordial en WordNet, aparte, WordNet ofrece las de antonimia, superordinación (hiperonimia), subordinación (hiponimia), meronimia y relaciones morfológicas. WordNet está organizado en base a estas relaciones semánticas. Puesto que las relaciones semánticas son relaciones de significados, y los significados están representados por medio de "synsets", WordNet expresa las relaciones semánticas como punteros (*pointers*) entre "synsets".

2.1.2. Sustantivos en WordNet

WordNet versión 1.5 [15] contiene aproximadamente 57,000 formas nominales (sustantivos) organizadas en unos 48,000 significados *("synsets")*. Las definiciones de los nombres están organizadas en jerarquías semánticas, construidas en base a los términos superordinados que aparecen en las definiciones de los sustantivos, junto con los rasgos distintivos que diferencian un sustantivo de su hiperónimo. Esta relación de superordinación genera una organización semántica jerárquica que WordNet duplica por medio del uso de punteros entre *"synsets"*. Los rasgos distintivos se introducen de manera que se crea un sistema de herencia léxica en el que cada palabra hereda los rasgos distintivos de su término superordinado, creándose una jerarquía tal como lo muestra la figura 2.1.

Los sustantivos de WordNet no están estructurados en torno a una jerarquía única que contenga un término superordinado general del tipo {*entity* } que englobe a todos los demás. Al contrario que en otras jerarquías, los sustantivos se han agrupado en torno a un grupo de conceptos genéricos ("primitivos semánticos"), de forma que cada uno de ellos es el término superior de una jerarquía separada. Estas jerarquías se corresponden, según sus autores, con campos léxicos relativamente bien definidos, cada uno de los cuales cuenta con su propio vocabulario.

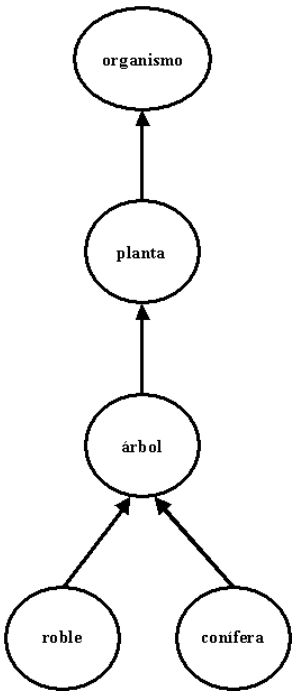

Figura 2.1: Ejemplo de jerarquías en los sustantivos

El mayor problema que esta estructuración plantea es el estado de estos primitivos. En algunos casos, se corresponden con elementos léxicos, y, en otros casos, los denominan "componentes semánticos primitivos" (*primitive semantic components*) [15], y los consideran conceptos a los que se adscribe un campo léxico y los lexemas que en él se encuentran [15]. Otro problema de este tipo de jerarquías múltiples es, por supuesto, decidir cuáles han de ser esos conceptos genéricos que actúen como términos superiores en las jerarquías. En WordNet, se han incluido los que consideran necesarios para dar cabida a todos los sustantivos del inglés.

Gran parte de la estructuración de los sustantivos de WordNet se ha generado por medio de las relaciones de hiponimia. También en este caso parece bastante confuso el uso del término "concepto", ya que a la hora de ejemplificar estos rasgos, parece que tanto los rasgos en sí, como los elementos a los que se adscriben se identifican como elementos léxicos y no como conceptos [15]. Vease un ejemplo:

> [....] a canary is a bird that is small, colorful, sings and flies, so not only must canary be entered as a hyponym of bird, but the attributes of small size and bright color must be included, as well as the activities of singing and flying. Moreover, canary must inherit from bird the fact that it has a beak and wings with feathers. In order to make all of this information available when canary is activated, it must be possible to associate canary with at least three different kinds of distinguishing features: Attributes: small, yellow Parts: beak, wings Functions: sing, fly

En este ejemplo, los rasgos distintivos que Miller propone parece que deban ser tratados como conceptos, ya que son los que distinguen *canary* de *bird*. Sin embargo, Miller identifica a continuación los atributos con los adjetivos contenidos en WordNet, las partes con los nombres y las funciones con los verbos, de modo que parece que en realidad sean elementos léxicos a los que la descripción de *canary* deba dirigir punteros.

2.1.3. Adjetivos

WordNet [15] divide los adjetivos en dos clases principales: descriptivos y relacionales, contiene aproximadamente 19,000 formas adjetivales, organizadas en unos 10,000 *synsets* o significados léxicos. También contiene un grupo cerrado de adjetivos como *former* o *alleged*, que se consideran como adjetivos de modificación de referencia (*reference-modifying adjectives*).

La antonimia es la relación semántica básica de los adjetivos descriptivos. En el caso de dos adjetivos que poseen un significado muy parecido, pero que tienen antónimos diferentes, o en aquellos casos en los que el adjetivo no tiene ningún antónimo, y no se le puede asignar el de otro adjetivo de significado similar, nuevamente representa un problema y éste radica nuevamente en la falta de una separación clara entre conceptos y unidades léxicas. No se puede equiparar la relación de antonimia que existe (o puede existir) entre conceptos con la que puede existir entre unidades léxicas. Por ejemplo, la mayoría de los antónimos de adjetivos ingleses se construyen por medio de procesos morfológicos (añadiendo un prefijo negativo al adjetivo), y las reglas morfológicas se aplican a las unidades léxicas, no a sus significados.

2.1.4. Verbos

WordNet [15] contiene más de 21.000 verbos (formas verbales) y aproximadamente 8,400 significados léxicos (*"synsets"*).

Los verbos están divididos en 15 archivos diferentes, en base a criterios semánticos. Todos estos archivos contienen verbos que denotan acciones o eventos, a excepción de un archivo que contiene verbos que se refieren a estados, aunque éstos no forman un dominio semántico ni comparten otra propiedad semántica que no sea la de referirse a estados.

Si el principio de herencia léxica servía para organizar las relaciones semánticas entre sustantivos y el de oposiciones bipolares las de adjetivos, las diferentes relaciones que organizan los verbos en WordNet se aglutinan en torno al principio de implicación léxica (*lexical entailment*).

La relación semántica de hiponimia considerada entre sustantivos, se denomina en el caso de los verbos "troponimia" (*troponymy*), ya que se considera que las distinciones de "modo" son las más importantes a la hora de diferenciar un hipónimo verbal de su hiperónimo.

Las otras dos relaciones de implicación consideradas en WordNet son la relación de oposición y la relación causal. La relación de oposición entre verbos es bastante compleja, ya que, la oposición entre verbos está basada en muchos casos en un proceso morfológico que se aplica a uno de los miembros de la oposición, planteando problemas similares a los que se plantean en referencia a los adjetivos.

Estos cuatro tipos de implicaciones se ajustan mejor para organizar unos tipos de verbos que otros. La finalidad principal de WordNet era convertirse en un reflejo computacional de la memoria léxica y no la representación del conocimiento léxico, por lo que no incluye mucha de la información que un hablante nativo posee acerca de las propiedades semánticas y sintácticas de los verbos [16].

2.1.5. Uso de WordNet en desambiguación

Desde el artículo original de Fellbaum [62], Wordnet se ha consolidado como un tesoro valioso para los investigadores de PLN. Su uso se ha expandido sobre diversas áreas de PLN y el caso particular de WSD no ha sido la excepción, ya que la estructura que maneja WordNet, permite de alguna manera enriquecer la representación de la palabra ambigua. Múltiples investigadores lo han usado con la finalidad de determinar el sentido correcto de una palabra ambigua, inte-

grándolo regularmente como una base de conocimiento externa que apoya algún determinado algoritmo. Por ejemplo, Galley y McKeown [61] usan WordNet para WSD utilizando una técnica denominada encadenamiento léxico, en donde se determina la relación que pueden tener diversas palabras a través de todo un texto. Pedersen *et al.* [64], por otro lado, usan WordNet para WSD con una técnica modificada del algoritmo de Lesk [11]. Incluso algunas empresas, como es el caso de "CL Research" [63], proporcionan versiones brevemente modificadas de WordNet para propósitos de WSD. Las aplicaciones de WSD que usan WordNet son amplias, basta mencionar que desde que Kilgarriff decide promover un evento piloto para medir la eficacia de los sistemas de desambiguación de sentidos (SENSEVAL, 1998 [59]), el número de sistemas que usan WordNet como apoyo a la tarea de WSD se ha incrementado notablemente. En el sitio "WordNet Bibliography" [65] es posible encontrar una bibliografía extensa sobre el uso de Wordnet; y si se desea observar su uso en WSD, basta buscar las palabras clave: *"sense disambiguation"*.

Como se mencionó anteriormente, las competencias de Senseval han sido el marco de referencia para la generación de nuevos sistemas para WSD. A diferncia de Senseval-1, en donde se evaluó sobre el *corpus* "Héctor", Senseval-2 y Senseval-3 ocuparon los sentidos de WordNet; ésto a conducido de algún modo a que algunos sistemas presentados en estas competencias usen WordNet como un recurso léxico de apoyo. Y a pesar de que se realizan distinciones entre sistemas supervisados y no supervisados, e incluso una pequeña distinicón para aquellos sistemas que sólo utilizan el *corpus* de prueba (completamente no supervisado), la realidad es que la mayoría de los sistemas usan fuentes de conocimiento externas tales como diccionarios o *corpora*.

Algunos de los sistemas que usan WordNet en la tarea de WSD, son presentados por Rada *et al.* en [54]. Ahí se presentan 46 sistemas en total, de los cuales 37 son supervisados y 9 son no supervisados. GAMBL [69], por ejemplo, utiliza WordNet como un lexicon de sentidos que apoya su proceso de WSD; este sistema aplica técnicas de aprendizaje basadas en memoria (*Memory-Based Learning*) para entrenar el módulo de identificación de palabras ambiguas y un algoritmo genético para optimizar las características de los contextos de las mismas palabras. Altaf *et al.* [70], por otro lado, utiliza WordNet para identificar colocaciones de palabras como un primer paso en la construcción de un árbol de análisis (*parse tree construction*) para un sitema basado en formas lógicas. Las formas lógicas son representaciones en lógica de primer orden de texto en lenguaje natural. Una forma lógica es una colección de predicados derivados de

un texto. En este caso las formas lógicas son derivadas a partir de la salida de un analizador sintáctico. Los sentidos de WordNet son utilizados también aquí para incluirlos dentro de la forma lógica. El sistema KUNLP [71] usa también WordNet para el proceso de WSD y fué aplicado en la tarea de desambiguación para el inglés; este sistema se basa en una técnica propuesta por Yarowsky [32], sin embargo, sustituye el tesoro de Roget por WordNet. Se utiliza el concepto de términos relativos a una palabra y que son basicamente todos aquellos que tienen una relación con ésta, por ejemplo, sus sinonimos, antonimos, hiperonimos o hiponimos. Este sistema es bastante similar a la aproximación basada en WordNet de Leacock *et al.* [72], obteniendo términos relativos de una cierta palabra a partir de WordNet y extrayendo frecuencias de co-ocurrencia de los relativos desde un *corpus*; la única diferencia consiste en el uso de relativos con muchos sentidos (*polysemous*) a diferencia de los relativos de un solo sentido (*monosemous*) de Leacock, quien considera la longitud entre los nodos de la jerarquía normalizada (n_1, n_2) por la profundidad (D, que es 16 en la versión 1.7 para sustantivos): $LCH(n_1, n_2) = -log(longMin(n_1 n_2)/(2 * D))$.

Resnik [26] por su parte se basa en el contenido de la información de un concepto (nodo): $CI(n) = -log(Pr(n))$. Así, la medida de relación entre dos nodos está dada en proporción a la cantidad de información que comparten; esta idea es relativa a un nodo y por eso se utiliza el "menor" nodo que subsane a ambos (mns): $Res(n_1, n_2) = CI(mns(n_1, n_2))$.

Por su parte, Lin utiliza como referencia a Resnik y Leacock, y toma una medida basada en el teorema de la similitud: la similitud de dos conceptos se mide por la cantidad de información que ellos aportan. Esto lleva a un cálculo de cantidad de información condicional:

$$Lin(n_1, n_2) = \frac{2 * CI(mns(n_1, n_2))}{CI(n_1) + CI(n_2)}.$$

Algunos otros métodos que utilizan WordNet son los basados en marcas de especificidad [75] y en densidad conceptual [73] [74]. La densidad conceptual está dada por una proporción de la cantidad de nodos en un área delimitada por concepto (nodo) y las áreas delimitadas por los diferentes sentidos de la palabra. Una fórmula que realiza lo anterior es:

$$AR(n_1, n_2) = \frac{\sum_{i=0}^{m-1} nhyp^i}{\sum_{i=0}^{h} nhyp^i},$$

donde m es el número de sentidos de las palabras del contexto o de la palabra

a desambiguar, h la altura del nodo que contiene a todos los subárboles de los sentidos considerados en m, y $nhyp$ el número de hipónimos que deben concordar con la siguiente entidad: $desc = \sum_{i=0}^{h} nhyp^i$; es decir, el número de nodos de un árbol balanceado se toma como un área de una región en la jerarquía.

Aún así, solamente unos cuantos sistemas han intentado evitar o reducir el problema de utilización de fuentes externas de conocimiento, como es el caso de aquellos que realizan agrupamiento del sentido de las palabras (*word sense clustering*) o discriminación del sentido de las palabras (*word sense discrimination*) [67].

Para concluir, el resultado de WordNet es impresionante en cuanto a la cantidad de información que contiene, sobre todo si se tiene en cuenta que toda esta información fue incluida manualmente por el grupo de lexicógrafos del proyecto. Las ventajas de contar con toda esta información en formato electrónico son muchas, aunque los mayores problemas que se plantean son prácticos ya que las operaciones realizables con el conjunto de herramientas informáticas implementadas hasta el momento son ciertamente limitadas (básicamente, ordenación y comparación de los elementos contenidos con los elementos de otros conjuntos).

La información que contienen los diferentes archivos es ciertamente valiosa, lo que hace que futuros proyectos que tomen WordNet como base puedan realmente sacar partido de ella si se integra en un sistema computacional apropiado, aunque debemos tener en cuenta también que no puede considerarse como un repositorio de conocimiento léxico detallado, sino como una interesante representación de las diferentes relaciones semánticas que existen entre elementos léxicos, en un intento de capturar la organización de la memoria léxica.

Capítulo 3

Algoritmos de Agrupamiento y Selección de Rasgos

La meta del agrupamiento consiste en reducir la cantidad de instancias mediante categorización o agrupamiento por similitud. Una de las motivaciones para usar algoritmos de agrupamiento es proveer herramientas automatizadas para ayudar en la construcción de categorías.

Los métodos de agrupamiento pueden ser divididos dentro de dos tipos básicos: agrupamiento jerárquico y particional [77]. Dentro de estos dos tipos existen muchos subtipos y diferentes algoritmos para encontrar grupos.

Un problema con los métodos de agrupamiento es que la interpretación de grupos puede ser difícil, ya que la mayoría de los algoritmos de agrupamiento prefieren ciertas formas de generación de grupos, además de que éstos, generalmente, asignarán siempre las instancias a los grupos, aún si las instancias no pertenecen a algún grupo.

Otro problema potencial es la elección del número de grupos, la cual puede ser crítica. Los mejores algoritmos obtienen la cantidad de grupos automáticamente, sin embargo, existen algunos que necesitan este número como un parámetro de entrada. Independientemente del algoritmo, se debe obtener una instancia inicial que conforme al grupo (centroide) y posteriormente podrán agregarse más instancias a los grupos. La calidad del agrupamiento depende fundamentalmente de la elección de los centroides, ya que pueden emerger grupos

ligeramente diferentes, cuando se cambia el número de centroides. La buena inicialización de los centroides de los grupos es crucial, ya que algunos grupos podrían incluso quedar vacíos si su centroide se encuentra inicialmente muy lejos de la distribución de las instancias.

Teniendo en cuenta las dificultades para hacer agrupamiento, en la elección del algoritmo a usar, se toma en cuenta que tan viable puede ser considerar un híbrido con otro algoritmo, qué tan sencilla puede ser la implementación de dicho algoritmo y qué tan bueno resulta.

En este capítulo se presentan varias técnicas que fueron implantadas con la finalidad de reportar su rendimiento en el ámbito de agrupamiento de contextos para palabras ambiguas. La primer técnica está fundamentada en el uso de un algoritmo voraz que utiliza una función de similitud basada en el valor del coseno del ángulo entre vectores representativos de cada contexto. La segunda técnica se le denomina MOD-SLC, en la que se enfoca el trabajo, y está basada en la propuesta de Hassan *et al.* [45] para fabricación de partes. Una propuesta original de este trabajo es refinar la técnica MOD-SLC mediante iteraciones continuas, usando la técnica de temple simulado. También se propuso y se implementó la técnica denominada K-means y una variante de la técnica KNN. Por último, se implementó una mezcla de algunas de las técnicas anteriores, las cuales se denominaron SLC-SLC y SLC-KNN.

La parte final del capítulo se orienta a la elección de características de los grupos encontrados. Para hacer el agrupamiento, no importando la técnica que se decida, es necesario tener los contextos de la palabra a desambiguar. Se representan los contextos de cada palabra, usando información mutua para obtener el mínimo número de palabras que describan al mismo (palabras representativas). Una vez obtenidos los grupos se procede a elegir los rasgos de cada grupo, y así generar la base de datos léxica. En algunas técnicas se calcula la eficacia asumiendo que cada grupo está compuesto de solamente una palabra (la palabra a asignar). Se revisa para cada grupo cuál obtiene la mayor eficacia y a ese grupo se asigna la palabra, como representativa del grupo; la otra forma de elegir las palabras representativas del grupo, es obteniendo la unión de todas las intersecciones por pares de contextos (solución SR). A continuación se describe a detalle cada una de las técnicas implantadas en este trabajo, así como las dos formas de selección de rasgos.

3.1. Agrupamiento

3.1.1. Algoritmo de Salton

La técnica utiliza básicamente un algoritmo de agrupamiento voraz y una función de similitud basada en el valor del coseno del ángulo entre vectores representativos de cada contexto.

El modelo de espacio vectorial fue propuesto originalmente por Gerard Salton [46], y se fundamenta en la representación de documentos mediante vectores en un espacio de términos. La descripción formal del modelo incluyendo preprocesamiento, representación y función de similitud se presenta enseguida. Posteriormente se calcula la función de similitud y se utiliza en el marco del algoritmo presentado en la figura 3.1 que corresponde a una búsqueda voraz. Se considera en la descripción las siguientes entidades.

Vectores de índice. Sea $D = \{D_1, \ldots, D_M\}$ una colección de documentos. Sea $V = \bigcup_i D$ el vocabulario de la colección, y $V_0 = [v_i]_i$ el vocabulario ordenado lexicográficamente. La representación de un texto D es el vector $D = [d_i]_{i \leq n}$, un vector está ahora formado por los contextos de una palabra, donde $d_i = 1$ si $v_i \in D$ y $d_i = 0$ si $v_i \notin D$, con $n = \#V_0$.

Asignación de pesos. Las componentes de cada vector $\overrightarrow{D_i} = [d_{i1}, ..., d_{in}]$ son ponderadas de la siguiente forma: $d_{ik} = tf_{ik} * idf_k$ donde tf_{ik} es la frecuencia del término k en D_i, e idf_k está definido como $idf_k = log_2(M) - log_2(df_k) + 1$, siendo df_k el número de documentos que usan el término k, y M el número de documentos.

Función de Similitud. Tal cual se había mencionado con anterioridad, en el caso de la representación vectorial se emplea el coseno del ángulo entre los vectores que representan a los documentos, como lo muestra la fórmula siguiente:

$$sim(\vec{D}_i, \vec{D}_j) = \frac{\sum_{k=1}^{M} d_{ik} d_{jk}}{\sqrt{\sum_{k=1}^{M} d_{ik}^2 * \sum_{k=1}^{M} d_{jk}^2}}$$

En este caso, D_i y D_j son dos contextos con los que se desea calcular su similitud. Cabe recalcar que en este trabajo, se ha modificado el concepto de manera breve, de tal forma que para cada palabra ambigua se representan sus respectivos contextos. Una vez agrupados los contextos ver figura 3.1, para una palabra x se procede a la selección de características de cada grupo $G(x) : R(x)$ obteniendo como resultado los rasgos representativos.

Entrada: *δ: Conjunto de vectores que representan a cada contexto de una palabra*
Salida: *Grupos generados por el algoritmo*

1. Sea $\delta = C_1, C_2, ..., C_n$ el conjunto de contextos de un término.
2. Mientras $\delta \neq \emptyset$ hacer
   - *a*) Tomar un $x \in \delta$
   - *b*) Crear grupo: $G(x)$
   - *c*) Sea $\delta = \delta - x$
   - *d*) Para cada $y \in \delta$
     1) Si $sim(x, y) \geq umbral$ entonces hacer
        - Añadir y al grupo: $G(x) \cup \{y\}$
        - $\delta = \delta - y$

Figura 3.1: Algoritmo Voraz

3.1.2. MOD-SLC

MOD-SLC está basada originalmente en la técnica SLC (Single Linkage Cluster), usada en biología y propuesta por Sneath [48], en la cual se utiliza el coeficiente de similitud de Jaccard para encontrar la similitud entre bacterias. Por su parte, la versión modificada de la técnica SLC (MOD-SLC), utiliza una variante en el coeficiente de similitud de Jaccard conocido como coeficiente de similitud de Jaccardian o non-Jaccardian [45], el cual es una medida del nivel de empatamiento, en donde el número de empates es dividido por una cantidad normalizada. Este coeficiente tiene un término adicional en el numerador y es básicamente el número de parejas perdidas, el cual es dividido por la normalización de términos. En nuestro caso, hemos hecho uso de la medida de Jaccardian con la finalidad de medir el grado de similitud entre contextos.

El coeficiente de similitud usado queda definido como sigue:

$$SB_{ij} = (X_{ij} + \sqrt{X_{ij} * Y_{ij}})/(X_i + X_j + X_{ij} + \sqrt{X_{ij} * Y_{ij}})$$

donde X_{ij} es el número de unos en común entre los contextos i y j en la matriz de incidencia (palabras en común), Y_{ij} es número de ceros en común (palabras que ambos contextos no tienen), X_i es el número de unos que están en el contexto i y no están en el contexto j (palabras diferentes) y X_j se define de manera similar a X_i pero para el contexto j. A continuación se presenta un

ejemplo para observar la manera en que se calcula la medida de Jaccardian.

Suponga los contextos $contx_1$, $contx_2$, $contx_3$, $contx_4$, $contx_5$, y $contx_6$, y el vocabulario formado por las palabras pal_1, pal_2, pal_3, pal_4 y pal_1, la matriz contexto-palabra queda distribuida como se muestra en la tabla 3.1. Entonces $X_{1,2} = 1$, $Y_{1,2} = 1$, $X_1 = 1$, $X_2 = 3$ y por tanto $SB_{1,2} = (1 + \sqrt{1 * 1})/(1 + 3 + 1 + \sqrt{1 * 1}) = 0{,}33$.

	pal_1	pal_2	pal_3	pal_4	pal_5	pal_6
$contx_1$	0	1	0	1	0	0
$contx_2$	1	1	1	0	1	0
$contx_3$	1	1	0	0	0	1
$contx_4$	1	1	1	1	0	1
$contx_5$	0	0	1	1	1	1
$contx_6$	1	0	0	1	1	0

Tabla 3.1: matriz contexto-palabra

El procedimiento para la generación de los grupos se describe en la figura 3.2.

3.1.3. MOD-SLC con Temple simulado

El temple simulado se ha utilizado ampliamente en la búsqueda de soluciones y es una analogía del proceso de templado de acero, en el cual se calienta el metal y después de cierta temperatura, se enfría repentinamente, para después volverlo a calentar, hasta que obtiene la templanza requerida; esto implica en algunos casos pasar de muy buenas calidades del metal a muy malas, para después volver a mejorar.

El algoritmo MOD-SLC con Temple simulado utiliza una modificación de la técnica Mod-SLC, haciendo una serie de iteraciones usando temple simulado. El algoritmo propone la obtención de una solución inicial de manera aleatoria, S_0. Se comprueba su eficacia (la cual mide la calidad de los grupos generados, $F(S_0)$) y se itera mientras la temperatura no llegue a cero. En cada iteración se calcula un vecino de la solución actual, S_1, y se compara la eficacia del mismo ($F(S_1)$) con la de la solución actual. Si la eficacia es mejor, se intercambian las respuestas y se procede con el algoritmo. Se mantiene un registro del valor óptimo para garantizar los mejores resultados. Así, la eficacia es el criterio que decide si una solución es "mejor" que otra. La fórmula de eficacia fue utilizada por Hassan *et al.* [45] en agrupamientos de partes de máquinas y se describe

Entrada: *Conjunto de contexto de una palabra ambigua*
Salida: *Conjunto de grupos generados para la palabra ambigua*

1. Generar la matriz de incidencia contexto-palabras.
2. Usando la matriz de incidencia crear la matriz de similitud de contextos(SM).
3. Calcular la similitud promedio(SA) de SM.
4. Localizar el máximo valor de similitud en SM (SB_{ij}) mayor o igual que SA. Por tanto, se encuentra un par de contextos C_i y C_j. Si no se encuentra dicho valor entonces terminar.
5. Asignar el contexto i (C_i) y el contexto j (C_j) al mismo grupo tomando en cuenta lo siguiente:
   a) Si C_i y C_j no han sido asignados a algún grupo, entonces se genera un nuevo grupo con estos dos contextos.
   b) Si C_i y C_j ya han sido asignados, pero a diferentes grupos, entonces por medio de una operación de unión, se genera un solo grupo.
   c) Si C_i o C_j no ha sido asignado, entonces el contexto no asignado se agrega al grupo del contexto ya asignado.
6. Eliminar la similitud (SB_{ij}) de SM ($SB_{ij} = 0$)
7. Regresar a paso 4.

Figura 3.2: Algoritmo Mod-SLC

más adelante en este mismo capítulo, en la sección 3.2.2.

Una de las modificaciones a este algoritmo, es la introducción de una función de penalización, en la cual, existe la posibilidad de aceptar malas soluciones bajo la hipótesis de que dichas soluciones permitirán en un futuro mejorar la solución óptima. Básicamente, este mecanismo permite escapar de óptimos locales. El mecanismo para la obtención de vecinos se describe a continuación:

Obtención de vecinos: No existe una garantía en el número de grupos generados de manera aleatoria, por lo que es necesario introducir criterios de aleatoriedad para mover contextos de grupo en grupo, dando la posibilidad incluso de crear un nuevo grupo o en su caso de eliminar uno existente. Se genera un número aleatorio para indicar el grupo a considerar, posteriormente dentro de ese grupo se elige un contexto de manera aleatoria y se elimina de dicho grupo (inciso "a" del algoritmo). En seguida se genera un número aleatorio entre 1 y el

Entrada: *Conjunto de contexto de una palabra ambigua*
Salida: *Conjunto de grupos generados para la palabra ambigua*

1. Generar la matriz de incidencia contexto-palabras.
2. Usando la matriz de incidencia crear la matriz de similitud de contextos.
3. Localizar el máximo valor de similitud (SB_{ij}) en la matriz de similitud.
4. Asignar el contexto i y el contexto j al mismo grupo
5. Eliminar la similitud (SB_{ij}) de la matriz de similitud
6. Si no están todos los contextos asignados regresar a paso 3.
7. Establecer como solución inicial S_0 el agrupamiento encontrado
8. Calcular la eficacia de la solución inicial, $F(S_0)$
9. Para $i = 1$ hasta $MAXITERACIONES$
   - *a*) Se obtiene solución vecina, S_1, resultado de intercambiar dos contextos de S_0
   - *b*) Si $F(S_0)$ es menor que $F(S_1)$ entonces $S_0 = S_1$
   - *c*) En caso contrario aceptar S_1 con un cierto umbral de probabilidad (por ejemplo: 0.3)

Figura 3.3: Algoritmo Mod-SLC con Temple Simulado

número de grupos actuales más uno y se introduce el contexto extraído dentro de este grupo. En cada iteración se verifica la eficacia global de los grupos. El procedimiento usado se describe en la figura 3.3.

3.1.4. K-means

Esta técnica utiliza un mecanismo supervisado. La idea principal de esta técnica es elegir k centroides o conjunto de pivotes iniciales que indican cuántos grupos se generan. Una desventaja de este algoritmo es que necesita conocer de antemano el número de sentidos que tiene la palabra ambigua. El procedimiento usado se describe en la figura 3.4.

3.1.5. KNN-MOD

Esta técnica se encuentra basado en el algoritmo de K-vecinos más cercanos (K-Nearest Neighbor o KNN), con modificaciones para calcular dinámicamente

Entrada: *Conjunto de contexto de una palabra ambigua, y el valor de k*
Salida: *Conjunto de grupos generados para la palabra ambigua*

1. Generar la matriz de incidencia contexto-palabras.
2. Usando la matriz de incidencia crear la matriz de similitud de contextos.
3. Localizar los menores valores de similitud en la matriz y sobre éstos elegir k contextos que formarán la base de los k grupos.
4. Para cada uno de los contextos no asignados verificar la máxima similitud a cada uno de los k grupos. Asignar el contexto al grupo con la máxima similitud.

Figura 3.4: Algoritmo K-means

el número de grupos. El algoritmo obtiene el par de contextos más similares y crea un grupo conformado por estos contextos. En seguida verifica para cada contexto no asignado el promedio de la similitud de dicho contexto contra cada uno de los contextos pertenecientes al grupo. En caso de que este promedio sea mayor o igual al promedio de similitudes global entonces este contexto se agrega al grupo. El proceso se repite generando con los restantes otro grupo, hasta haber asignado cada uno de los contextos. El procedimiento usado se describe en la figura 3.5.

3.2. Selección de rasgos

3.2.1. Información Mutua

Inicalmente se hizo un preprocesamiento de los contextos encontrados para la palabra ambigua. Dicho preproceso, por así llamarlo, consistió en la eliminación de palabras que fueran menos representativas a la palabra a desambiguar. Para lograr este objetivo, se utilizó una medida de asociación denominada Información Mutua (ver [67]), la cual es presentada a continuación:

Dadas w_1 y w_2, la primera una palabra ambigua y la segunda una palabra de algún contexto de w_1, la información mutua entre ellas es:

$$IM(w_1, w_2) = log_2 \left(\frac{N * Fr(w_1, w_2)}{Fr(w_1) * Fr(w_2)} + 1 \right)$$

donde $Fr(w_1)$ y $Fr(w_2)$ son las frecuencias de w_1 y w_2 respectivamente, en el conjunto de contextos, $Fr(w_1, w_2)$ es la frecuencia del par de palabras w_1 y

Entrada: *Conjunto de contexto de una palabra ambigua*
Salida: *Conjunto de grupos generados para la palabra ambigua*

1. Generar la matriz de incidencia contextos-palabras.
2. Usando la matriz de incidencia crear la matriz de similitud de contextos (SM).
3. Calcular el promedio global de similitudes (Umbral). Sea n el número total de contextos y NC $= \left(\frac{n^2-n}{2}\right)$ el número de valores en la matriz triangular superior de SM; se define $Umbral = 1/NC * \sum_{j>i} sim(C_i, C_j)$
4. Localizar j, i tales que $j > i$, $sim(C_i, C_j) = max_{p,q}\{sim(C_p, C_q)\}$,
5. Generar un nuevo grupo conformado por los contextos C_i y C_j encontrados en el paso anterior ($G = \{C_i, C_j\}$)
6. Sea m el número de contextos asignados al grupo G. Para cada uno de los contextos aún no asignados (C_k) hacer:
   a) $simG = 1/m * \sum_{r=1}^{m} sim(C_r, C_k)$
   b) Si $simG \geq Umbral$ entonces agregar C_k al grupo G
7. Mientras haya contextos sin asignar regresar al paso 4, en caso contrario terminar.

Figura 3.5: Algoritmo KNN-MOD

w_2 en el conjunto de contextos, y N es el número de contextos de w_1.

Una vez que se hace el preprocesamiento de los contextos, se procede a la generación de los grupos, utilizando el algoritmo de agrupamiento deseado. Cuando los grupos son creados, se procede a la elección de características que identificarán a cada grupo. Para ello, se aplicarón dos técnicas: solución por eficacia y una técnica basada en el sentido de un sintagma, denominada SR.

3.2.2. Eficacia

En nuestro caso, entendemos que una palabra es un término del vocabulario del conjunto de grupos generados por el algoritmo elegido.

La medida de la eficacia global para los grupos generados y que se usa también para la asignación de palabras a contextos se realiza mediante la siguiente

fórmula:

$$Eficacia = (e - e_0)/(e + e_1)$$

donde, e es la cantidad de unos en la matriz, e_0 es el número de elementos excepcionales (fuera de los grupos) y e_1 es el número de ceros dentro de los grupos. Esto es, la fórmula obtendrá un valor de 1 cuando no existen elementos excepcionales(EE) o cero's (Zs) dentro de los grupos generados. Por otro lado, el valor de eficacia tiende a cero, de manera proporcional al incremento de (EE y Zs). Aunque Hassan *et al.* [45] usan la fórmula de eficacia para medir la calidad de los grupos generados por su algoritmo, en este trabajo se usa para el proceso de selección de características.

3.2.3. Solapamiento de rasgos (SR)

Esta técnica obtiene características para cada grupo, mediante la unión de todas las intersecciones por pares de contextos (solución SR). Esta solución mejoró los resultados obtenidos por la fórmula de eficacia para la selección de rasgos y por tanto se eligió para las pruebas finales.

Dado un conjunto de contextos $SC = \{ctx_1, ctx_2, ..., ctx_r\}$ de un grupo G_i, la solución SR de G_i es un conjunto de términos pertenecientes al grupo que cumplan la siguiente fórmula:

$$SR = \bigcup_{j \neq k} (ctx_j \bigcap ctx_k)$$

La idea está basada en una propuesta para obtener el sentido de un sintagma. En [58] se dice que, intuitivamente se supone que el sentido de un sintagma está determinado por una combinación de propiedades de las palabras que aparecen en el sintagma. En [3] se presentan experimentos de representación de documentos basados en sintagmas para recuperación de información. En [57], se usa tambiíen una fórmula basada en esta idea, considerando las propiedades bajo el uso de un diccionario, dando la siguiente función, que asocia esas propiedades a una palabra w:

$$\Xi : V \rightarrow V^*, w \mapsto \Xi(w) = Prop(w).$$

Si $\psi(w, z) = \Xi(w) \bigcap \Xi(z)$, el sentido del sintagma $w_1, ..., w_n, w$ es generado como sigue:

$$\xi(w_1, ..., w_n, w) = \xi(w_1, ...w_n, w) \cup \bigcup_{i=1}^{n} \psi(w_i, w)$$

donde $\xi(w_1, w_2) = \psi(w_1, w_2)$

En este trabajo se toman las propiedades como el contexto de la palabra ambigua. Esta idea está basada en las relaciones de sentido que son asociadas a una palabra wd (toda palabra relacionada con wd). Asi se obtiene la fórmula SR.

Capítulo 4

Pruebas

La falta de conjuntos de entrenamiento que ayuden a la tarea de WSD y otras fuentes de conocimiento para dominios específicos ha conducido a la necesidad de experimentar con algoritmos no supervisados. En este capítulo, se presentan los resultados de un nuevo enfoque del algoritmo MOD-SLC como técnica de agrupamiento de sentidos para la desambiguación de palabras de SENSEVAL-2. Proponemos una técnica novedosa para la selección de características (nuestra principal contribución), basada en el sentido de un sintagma. Una doble ejecución de la técnica MOD-SLC fue aplicada para obtener las mejores características de cada grupo. Además no se usa ningúna fuente de información externa, y se usan los sentidos de palabras ambiguas de SENSEVAL para medir de la calidad de nuestro algoritmo.

Se ha probado con varios algoritmos de agrupamiento, tanto supervisados como no supervisados, tales como K-mean, KNN con una modificación (KNN-MOD), Slc modificado (Mod-SLC), etc. Y finalmente se decidió usar Mod-SLC, porque este algoritmo mostró el mejor comportamiento, además de ser un algoritmo totalmente no supervisado.

4.1. Resultados preliminares

Las pruebas preliminares hicieron uso de un *corpus* heterogéneo (CorpCIC) [53] [1], para obtener los contextos de los sustantivos. El paso más importante consistió en agrupar estos contextos con el fin de extraer características de cada

[1] Recurso proporcionado por el laboratorio de PLN del CIC-IPN

uno de los sentidos de la palabra ambigua "banco"; se tomó cada grupo formado como un posible sentido de la palabra.

Las primeras observaciones indicaron la importancia de dos componentes: el algoritmo de agrupamiento y el criterio de selección de características o rasgos. Se procedió a evaluar ambos componentes a fin de determinar los mejores elementos a usar en el proceso de creación de la base de datos léxica. A continuación se discute la evaluación sobre el criterio de selección de rasgos.

4.1.1. Evaluación en la selección de rasgos

Se experimentó inicialmente con los mecanismos de "eficacia" y "SR", descritos en el capítulo anterior. Una evaluación sobre el *corpus* CorpCIC, usando la palabra ambigua "banco", y el conjunto de consultas presentadas en la tabla 4.1, indican que no existen diferencias significativas para ambos procesos de selección.

Los grupos obtenidos fueron evaluados manualmente, a través del proceso de desambiguación del sentido de la palabra ambigua "banco". Se realizaron los agrupamientos mediante la técnica Mod-SLC, con las dos variantes de selección de rasgos; y posteriormente se introdujeron las consultas para determinar si los grupos generados podían desambiguar el sentido del término.

No.	Consulta
1	El dolar puede cambiarse en varias instituciones financieras, tales como las casas de cambio o el **banco**
2	Demanda a **banco** mundial ayuda monetaria para abatir pobreza de mujeres
3	Se acentúa la baja de las tasa financieras para depósitos en los **bancos**
4	Esta chica siempre toma asiento en el mismo **banco**
5	El **banco** no pudo estar cerrado sino el acreedor habría ido a otro lugar a cambiar el cheque
6	Esta evaluación cuenta con un **banco** de docencia curricular donde se brinda un seguimiento académico al profesor
7	Puesto que el **banco** rocoso se encuentra con pronunciada inclinación desde el río Troja
8	El **banco** condone parte de la deuda agrícola

Tabla 4.1: Oraciones usadas para realizar la prueba de desambiguación

La tabla 4.2 muestra los resultados obtenidos (para más detalle verse [1]). La evaluación sobre un *corpus* de mayor tamaño y que indicó el uso de la técnica *SR*, como definitiva, se presenta más adelante.

	Eficacia			SR		
Número de oración	Grupo correcto	Grupo asignado	Valor de la similitud	Grupo correcto	Grupo asignado	Valor de la similitud
1	5	5	0.04	3	3	0.43
2	6	6	0.04	3	3	0.25
3	3	3	0.01	3	3	0.50
4	2	-	0.00	2	3	0.25
5	4	4	0.01	3	3	0.17
6	-	-	0.00	-	3	0.33
7	1	5	0.02	1	3	0.20
8	4	4	0.01	3	3	0.20

Tabla 4.2: Selección de ragos por Eficacia vs. SR en CorpCIC

La evaluación de los algoritmos de agrupamiento y los experimentos finales usaron un *corpus* más extenso y de uso generalizado en los sistemas para WSD. Por tal motivo, antes de explicar la evaluación de los algoritmos de agrupamiento, se introdicirá la descripción de este *corpus*.

4.1.2. Descripción del corpus de prueba

Para los experimentos finales, se extrajo un conjunto de palabras ambiguas de un diccionario (llamado MiniDir) desarrollado por CLIC[2] específicamente para SENSEVAL, obviamente para ser usado por sistemas de desambiguación semántica. Se usó un subconjunto de palabras del MiniDir que se muestran en la tabla 4.3 con su parte de discurso, así como el número de sentidos. MiniDir no ofrece un conjunto de sentidos tan detallado como lo hace WordNet, ya que presenta en promedio cuatro sentidos para los sustantivos y adjetivos y seis sentidos para los verbos. Los criterios usados en la elaboración del MiniDir 2.1 son listados en [66]. MiniDir cuenta con 58 palabras ambiguas, y una vista general del contenido de éste se presenta en la figura 4.1.

Después de seleccionar el conjunto de palabras ambiguas se utilizó un *corpus* preprocesado de entrenamiento de SENSEVAL-2 para el español (SST), para extraer los contextos que contienen las palabras ambiguas seleccionadas. El número de contextos encontrados para cada palabra también se reporta en la tabla 4.3. El tamaño del *corpus* es aproximadamente de 41 Megabytes. Debe quedar claro que el *corpus* solo se utilizó con fines de evaluación, y no como fuente externa de conocimiento.

[2] http://clic.fil.ub.es

La fase de preprocesamiento aplicado a SST consistió en la eliminación de palabras cerradas,y caracteres no alfabéticos y conversión a minúsculas. Mientras que el preprocesamiento de los contextos encontrados además requirió de la eliminación de palabras no representativas mediante el uso del concepto de información mutua, el cual fue presentado en la sección 3.2.1.

<?xml version="1.0" encoding="ISO-8859-1"?>
<minidir lang="Spanish" version="2.0" date="2004/02/13">
<lexelt item="actuar" pos="VM">
<sense id="actuar.1" definition=Realizar actos, ejercer funciones propias de su cargo o naturaleza" used="yes">
<example text="hay que actuar">
<example text="la fiscalía actuó contra el terrorista">
<example text="desde la muerte de su padre, el hermano mayor actuó como cabeza de familia">
<example text="actuar legalmente">
<example text="los jugadores suramericanos que actúan en España">
<collocation text="actuar en defensa propia">
<synset wordnet="1.5" id="01341700v">
<synset wordnet="1.5" id="00618376v">
<sense>

<sense id="actuar.2"definition="Comportarse de una manera determinada" used="yes">
<example text="ha actuado bien">
<example text="actuó sin lógica">
<example text="tiene unos raros modos de actuar">
<example text="se empeña en actuar de forma cicatera">
<synonym text="comportarse">
<synonym text="conducirse">
<synset wordnet="1.5" id="00007021v">
<sense>
<lexelt>
:
<minidir>

Figura 4.1: Estructura general del MiniDir

A continuación se presentan experimentos con diferentes algoritmos de agrupamiento, llevados a cabo en el desarrollo de esta tesis, y usando el *corpus* de SENSEVAL.

palabra ambigua	POS	#Sentidos	#Contextos
corona	verbo	4	297
hermano	sustantivo	3	593
apoyar	verbo	4	316
apuntar	verbo	9	39
subir	verbo	5	288
tocar	verbo	13	117
vencer	verbo	7	383
verde	sustantivo	5	342

Tabla 4.3: Subconjunto de palabras ambiguas usadas en el experimento

4.1.3. Pruebas con diversos algoritmos de agrupamiento

Se evaluaron cinco algoritmos de agrupamiento con la finalidad de decidir el mejor a usar en la conformación del sistema para la creación de bases de datos léxicas que apoyen en el proceso de desambiguación del sentido de un término. A continuación se discuten los resultados obtenidos en cada uno de ellos. Como se mencionó anteriormente, la evaluación es general, y está basada en la cantidad y calidad de grupos generados por cada uno de los algoritmos.

Algoritmo voraz con representación de Salton (AVTS)

Esta técnica generó grupos que mezclaban contextos que poseían diferentes sentidos, y aunque fueron pocos los grupos generados, no se podía determinar a qué sentido correspondía cada grupo. Se observó un alto grado de complejidad en la conformación del algoritmo, derivado principalmente del tipo de representación usado y que se encuentra basado en el modelo de espacio vectorial propuesto por Salton [46]. Se observa que la mayoría de los vectores que representan a los contextos, tienen muy pocos términos en común y una gran cantidad de términos con peso cero, derivado de la poca cantidad de términos que tienen cada uno de los contextos.

K-means

La problemática principal de esta técnica consiste en la determinación del valor de k, que es precisamente, el número de grupos a generar. En principio, se debería desconocer este parámetro y la introducción de un valor implica una modalidad en la supervisión, o puede ser visto como apoyo en una fuente de

información externa . A pesar de este inconveniente, se decidio verificar su comportamiento, usando como valor de k, el número de sentidos para cada palabra ambigua reportado en el MiniDir. Las evaluaciones fueron poco alentadoras, principalmente porque el número de sentidos reportado en el MiniDir, en la mayoría de los casos no se encuentra representado dentro del *corpus* de prueba, con lo cual se fuerza a formar un determinado número de grupos, que vician el verdadero agrupamiento.

KNN-MOD

Esta técnica se comportó aceptablemente, ya que el número de grupos a generar se obtuvo automáticamente. Su único inconveniente fue la generación de un grupo en el cual se mezclaban todos aquellos contextos que no pudieron, por alguna razón, ser agrupados en su sentido correcto.

Mod-SLC

Esta técnica presentó los mejores resultados en cuanto a la calidad de los grupos. Se observó que una primera implementación generaba un solo grupo final, sin embargo, una pequeña modificación al algoritmo permitió mejorar la calidad del agrupamiento, aunque incrementando el número de grupos generados. Esta última problemática fué resuelta satisfactoriamente y se discute con mayor detalle en la siguiente sección.

MOD-SLC con Temple simulado (SLCTS)

Se utilizó el algoritmo de temple simulado, con la finalidad de obtener iteraciones de agrupamientos, evaluados mediante la fórmula de eficacia. Se buscaba reducir el número de elementos excepcionales (ver la subsección de Eficacia del capítulo anterior) y decrementar el número de ceros dentro de cada grupo. Esta técnica se consideró como supervisada ya que se deben ajustar algunos parámetros, como temperatura(lo cual implica evaluar un agrupamiento con la ayuda de conocimiento externo, en este caso la manipulación manual de la temperatura para ir obteniendo los mejores resultados), para cada *corpus* y palabra ambigua distinta a utilizar. Los resultados obtenidos fueron desalentarores, ya que se generaron muchos grupos y de mala calidad.

Observaciones generales

La tabla 4.4 muestra algunas comparaciones generales sobre los resultados obtenidos sobre la agrupación de contextos para la palabra ambigua "corona" en el *corpus* de SENSEVAL. De acuerdo al MiniDir, esta palabra tiene cuatro diferentes sentidos. Para determinar la calidad de los grupos se hizo una comparación manual de los grupos generados con respecto a los sentidos presentados en el Minidir y debido a ello se establece que tan buenos o malos resultan ser estos grupos. El tiempo de ejecución fue medido manualmente y dependiendo del tiempo observado es que se da dicha medida.

Método	Grupos Generados	Supervisado	Tiempo de ejecución	Calidad de los grupos
AVTS	8	No	Lento	Malo
K-means	4	Si	Rápido	Malo
KNN-MOD	33	No	Lento	Regular
MOD-SLC	53	No	Rápido	Bueno
SLCTS	25	Si	Lento	Malo

Tabla 4.4: Evaluación general sobre diversos algoritmos de agrupamiento

Después de evaluar cada uno de los algoritmos se determinó utilizar Mod-SLC como el fundamento del algoritmo de agrupamiento. Los resultados finales, y que conforman la terminación de este trabajo de tesis, se presentan en la siguiente sección.

4.2. Resultados finales

4.2.1. Generación de grupos que representan sentidos de una palabra

El objetivo de este trabajo es generar una base de datos léxica (BDL) que permita la desambiguación del sentido de palabras ambiguas. Sea C el *corpus* que contiene ocurrencias de la palabra ambigua, y L la lista de palabras ambiguas. El procedimiento es el siguiente:

1. Extraer para cada $x \in L$ las características según $C : Ct(x)$.
2. Para cada x
   a) Agrupar los contextos $Ct(x)$.

b) Generar los rasgos para cada grupo correspondiente al agrupamiento hecho en el paso anterior.

3. Generar la BDL con las diferentes entradas por cada $x_i \in L$:

$$x_1(S_1) : rasgos_{11}$$

$$x_1(S_2) : rasgos_{12}$$

$$:$$

$$x_2(S_1) : rasgos_{21}$$

$$x_2(S_2) : rasgos_{22}$$

$$:$$

donde S_i corresponde al i-ésimo grupo que proporcionó los rasgos para la palabra en cuestión.

4.2.2. Selección de rasgos

Se hizo una evaluación de la selección de rasgos para la palabra ambigua "corona", sobre el *corpus* de SENSEVAL, usando los criterios de "Eficacia" y SR. Se usaron los cuatros sentidos de la palabra ambigua, reportados en el MiniDir, como criterios de evaluación. Los resultados obtenidos mostraron que el uso del traslape por pares de contextos (solución SR) arrojaba los mejores rasgos de cada grupo generado. Estos resultados son posiblemente derivados de la cantidad de palabras usadas en cada prueba y del tamaño de los *corpora* (ver tabla 4.5).

	Eficacia		SR	
Sentido de corona	Valor máximo similitud	Asignación del sentido	Valor máximo similitud	Asignación del sentido
símbolo premio o autoridad	0.90	Incorrecta	0.86	Correcta
unidad monetaria	0.94	Incorrecta	0.89	Incorrecta
reino o monarquía	0.92	Incorrecta	0.87	Incorrecta
aureola	0.90	Incorrecta	0.86	Correcta

Tabla 4.5: Evaluación de las técnicas Eficacia y SR en el *corpus* SENSEVAL

4.2.3. Evaluación

Para evaluar los resultados, se utilizó una propuesta de *baseline* hecha por Ted Pedersen [68] para agrupamiento no supervisado y usada por él en WSD. Para el cálculo de la exactitud del conjunto de grupos obtenidos, se necesitó un conjunto de clases (*gold standard*); cada clase debería contener ejemplos usando un sentido de cada palabra ambigua considerada.

El agrupamiento de oraciones de sentidos conocidos $S_1, ..., S_n$ en los grupos $c_1, ..., c_m$ puede ser visto en la tabla 4.6 (ejemplo dado por Ted Pedersen).

	S_1	S_2	S_3	Total
c_1	10	30	5	45
c_2	20	0	40	60
c_3	50	5	10	65
Total	80	35	55	170

Tabla 4.6: Ejemplo del cálculo *baseline* para tres sentidos y tres grupos

Asumiendo el peor de los casos, es decir, que todos los contextos sean agrupados en c_i, podemos conformar un *baseline*. Así, $(0 + 0 + 55)/170 = 0{,}32$ sería el resultado obtenido si c_3 correspondiera a S_3 y $(0 + 0 + 80)/170 = 0{,}47$ sería el resultado obtenido si c_3 correspondiera a S_1. Lo anterior se puede observar con mayor claridad en la tabla 4.7.

	S_1	S_2	S_3	Total
c_1	0	0	0	0
c_2	0	0	0	0
c_3	80	35	55	170
Total	80	35	55	170

Tabla 4.7: Ejemplo *baseline* para c_3

El promedio de la exactitud para una palabra ambigua w puede ser calculado como sigue:

$$Exactitud(w) = \frac{1}{NS} * \sum_{i=1}^{NS} \frac{Gc_i}{Gt_i},$$

donde Gc_i es el número de grupos correctamente asignados; Gt_i, es el número total de grupos hallados para el sentido i, y NS es el número real de sentidos para la palabra w. La evaluación de nuestro algoritmo es presentado en la sigu-

iente sección. Nosotros comparamos la exactitud con respecto a *baseline* para el subconjunto de palabras tomadas de SENSEVAL-2 (ver tabla 4.3).

4.2.4. Resultados

La evaluación de los resultados para las ocho palabras ambiguas es mostrada en la tabla 4.10. La parte (a) de la tabla muestra la exactitud del agrupamiento usando doble ejecución del algoritmo MOD-SLC, con selección de características en cada una de las ejecuciones realizadas, y en la parte (b) de la tabla se muestran los resultados de exactitud del agrupamiento cuando se hace la doble ejecución de MOD-SLC eligiendo las características únicamente en la primera ejecución del algoritmo. Para ejemplificar como se obtuvieron dichos resultados se toma la palabra ambigua "corona" y se procede a la evaluación por *baseline* y exactitud, de acuerdo a las fórmulas mostradas anteriormente. En la tabla 4.8 se muestran los grupos generados para "corona" (denotados por c_i). De acuerdo al MiniDir de Senseval, esta palabra tiene 4 sentidos, los cuales son denotados por S_j. Así, en la misma tabla se presentan los resultados obtenidos en el agrupamiento. Un valor distinto de cero en el renglón i y en la columna j indica que grupo c_i fue asignado al sentido S_j. De esta misma tabla se parte para hacer la evaluación.

	S_1	S_2	S_3	S_4	Total
c_1	0	0	0	0	0
c_2	0	0	0	0	0
c_3	0	0	1	0	1
c_4	0	0	1	0	1
c_5	0	1	0	0	1
c_6	0	0	0	0	0
c_7	0	1	0	0	1
c_8	0	1	0	0	1
c_9	0	0	0	0	0
c_{10}	0	0	0	0	0
c_{11}	0	0	0	0	0
c_{12}	0	0	0	0	0
Total	0	3	2	0	5

Tabla 4.8: Agrupamiento obtenido para la palabra ambigua "corona"

Al aplicar *baseline* en el peor de los casos, todos los contextos serían asignados a un determinado c_i, en este caso se tomó a c_3 (ver tabla 4.9). Así, $(0 + 0 +$

$2 + 0)/5 = 0{,}4$ es el resultado obtenido, ya que c_3 corresponde al sentido S_3 y $(0+3+0+0)/5 = 0{,}6$ es el resultado obtenido para c_5, ya que éste corresponde a S_2. Finalmente, se suman estos resultados parciales y se dividen entre el total de sentidos proporcionados por el MiniDir para la palabra ambigua a evaluar, esto para obtener el promedio *baseline*. Por lo tanto, el resultado final es 0,25, ya que $(0{,}4 + 0{,}6)/4 = 0{,}25$ (ver valor *baseline* para la palabra "corona" en la parte *a* en la tabla 4.10).

	S_1	S_2	S_3	S_4	Total
c_1	0	0	0	0	0
c_2	0	0	0	0	0
c_3	0	3	2	0	5
c_4	0	0	0	0	0
c_5	0	0	0	0	0
c_6	0	0	0	0	0
c_7	0	0	0	0	0
c_8	0	0	0	0	0
c_9	0	0	0	0	0
c_{10}	0	0	0	0	0
c_{11}	0	0	0	0	0
c_{12}	0	0	0	0	0
Total	0	3	2	0	5

Tabla 4.9: Tabla que se genera para "corona" con la evaluación *baseline*

Un ejemplo para la evaluación de exactitud utilizando la palabra ambigua "corona" se presenta a continuación. Usando nuevamente la tabla 4.8, se verifica el número de grupos que pertenecen a un determinado sentido, por ejemplo, los grupos c_5, c_6, c_7 y c_8 pertenecen al sentido S_2, sin embargo, únicamente tres grupos fueron asignados correctamente, por lo cual la exactitud obtenida para el sentido S_2 es 3/4. Si se sabe que los grupos c_1, c_3 y c_4 corresponden al sentido S_3, los grupos c_9, c_{10}, c_{11} y c_{12} corresponden al sentido S_1, y el grupo c_2 corresponde al sentido S_4, es fácil ver que la exactitud promedio es: $(0/4+3/4+2/3+0/1)/4 = 0{,}35$ (ver valor de exactitud para la palabra "corona" en la parte *a* en la tabla 4.10).

Los resultados presentados en la tabla 4.10 permiten observar la sensibilidad del proceso de la selección de características en una base de datos léxica para palabras ambiguas. La aplicación doble del proceso de selección no solamente eliminó palabras innecesarias, sino también algunas importantes para caracterizar a dichas palabras ambiguas. De esta manera, la aplicación simple

	(a)		(b)	
palabra ambigua	Exactitud	Baseline	Exactitud	Baseline
corona	0.35	0.25	0.46	0.16
hermano	0.06	0.33	0.46	0.15
apoyar	0.32	0.30	0.43	0.27
apuntar	0.11	0.11	0.11	0.11
subir	0.1	0.1	0.16	0.07
tocar	0.038	0.076	0.076	0.038
vencer	0.012	0.14	0.51	0.03
verde	0.125	0.20	0.20	0.12

Tabla 4.10: Valores de exactitud para MOD-SLC con: (a)doble SR, (b)simple SR

de selección permitió tener una mejor representación y, por lo tanto, tendrá un mejor rendimiento en el proceso de WSD.

Conclusiones y Perspectivas

Se ha explorado un tema tanto controvertido como difícil, el agrupamiento de contextos de una palabra ambigua para extraer rasgos que conformen cada uno de sus sentidos, sin el empleo de fuentes de conocimiento externas. Se experimentó con una serie de algoritmos de agrupamiento, obteniendo resultados alentadores para el algoritmo MOD-SLC y con el método de selección de características basado en el sentido de un sintagma (SR). La mejor combinación se obtuvo con la doble ejecución del algoritmo MOD-SLC, seleccionando rasgos solamente en la primera ejecución de MOD-SLC.

El *corpus* usado en los experimentos fue tomado de SENSEVAL-2, un evento dedicado explícitamente a WSD. La evaluación de los resultados se realizó usando el concepto de *baseline*, una medida de evaluación que implica una cota inferior para algoritmos de agrupamiento. La exactitud obtenida con selección simple de rasgos, muestra que el algoritmo propuesto mejora el *baseline* en todos los casos.

El objetivo principal de este trabajo fue el de proporcionar los fundamentos en la construcción de una herramienta de apoyo en los procesos de desambiguación del sentido de palabras para el lenguaje español, ya que existen pocas herramientas de este estilo, sobre todo que tengan la característica de ser totalmente no supervisadas. El cumplimiento de este objetivo permitirá avanzar significativamente en todas las áreas del procesamiento del lenguaje natural que requieran un módulo de desambiguación del sentido de las palabras.

Avances preliminares de este trabajo de tesis fueron publicados en el año 2004 (ver [1]), mientras que los resultados finales ya han sido aceptados para ser presentados y publicados en el evento IICAI'05 en la India [2].

Mucho es el trabajo por realizar, puesto que es un recurso poco desarrollado para el lenguaje español. A continuación se enumeran algunas tareas pendientes:

1. Comprobar el desempeño de la BDL que genera la presente propuesta y aplicarla en la tarea de *lexical-sample*
2. Realizar pruebas frente a otros algoritmos de selección de rasgos
3. Utilizar diversos *corpora* para medir la eficacia de la BDL construída, por ejemplo SemCor
4. Aplicar la BDL en alguna de las tareas propias de PLN; *parsing*, recuperación de información, etc.
5. Transportar estos algoritmos a otras aplicaciones semejantes: agrupamiento de homónimos, agrupamiento de *e-mails*, etc., para conocer su impacto

Bibliografía

[1] Sofía Paniagua Rivera, Héctor Jiménez Salazar y David Pinto, "Pruebas con algoritmos de agrupamiento para generar una Base de Datos Léxica", Avances en la Ciencia de la Computación, TLH-ENC 04, PP. 304-310, 2004.

[2] Sofía Paniagua Rivera, Héctor Jiménez-Salazar y David Pinto, "An Unsupervised Method for Senses Clustering", To be published, IICAI'05, December 2005.

[3] Miguel Rodríguez H., Héctor Jiménez Salazar y David Pinto, "Un Modelo de Representación basado en Sintagmas para Recuperación de Información", Avances en la Ciencia de la Computación, TLH-ENC 04, PP. 296-303, 2004.

[4] Ide, N., and Veroni J., "Introduction to the Special Issue on Word Sense Disambiguation", PP. 1-29, Vol. 24, Number 1, Computacional Linguistics, 1998.

[5] Zernik, Uri, "Train1 vs. Train2: Tagging Word Senses in Corpus", In proceedings of Intelligent Text and Image Handling, IRAO91, PP. 567-585, Barcelona, Spain, 1991.

[6] Masterman, Margaret, "The thesaurus in sintaxis and semantics.", Mechanical Translation, 1957.

[7] Masterman, Margaret, "Semantic message detection for machine translation usin interlingua.", Stationery Office, London, PP. 437-475, 1962.

[8] Wilks, Y. and Fass D., "Preference semantics: A family history.", Report MCCS, PP. 90-194, 1990.

[9] Wilks, Y. and Stevenson M., "The grammar of sense: Is word sense tagging much more than part-of-speech tagging?", Technical report CS-96-05, University of Sheffield, Sheffield, UK, 1996.

[10] Wilks, Y. and Stevenson M., "Sense Tagging: Semantic tagging with a lexicon", 1996.

[11] Lesk, Michael, "Automated sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone.", SIGDOC Conference, PP. 24-26, 1986.

[12] Miller, G., "WORDNET: a dictionary browser", In Proceedings of the First International Conference on Information in Data, University of Waterloo Centre for the New OED, Waterloo, Ontario, 1985.

[13] Miller, G., R. Beckwith, C. Fellbaum, D. Gross and K. Miller, "Five Papers on WordNet.", CSL Report 43., Cognitive Science Laboratory., Princeton University, 1990.

[14] Miller, G., "Noun in Wordnet: A Lexical Inheritance System," in Internation Journal of Lexicography, vol. 3, 1985.

[15] Miller George (Ed.), "WordNet: An On-line Lexical Database", International Journal of Lexicography, 1990.

[16] Fellbaum, C., "English Verbs as Semantic Net", Journal of Lexicography, vol. 6, Oxford University Press, 1993.

[17] Eric Brill, "Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging.", Computational Linguistics, 21(4):543-566, December 1995.

[18] Bar-Hillel,"sense ambiguity could not be resolved by electronic computer either current or imaginable", 1964.

[19] T. Strzalkowski, "Information retrieval using robust language processing", In AAAI Spring Symposium on Representation and Aquisition of Lexical Information, pages 104-111, Stanford, 1995.

[20] Krovetz, R. and Croft, W., "Lexical ambiguity and information retrieval", ACM Transactions on Information Systems, 10(2):115-141, 1992.

[21] R. Krovetz, "Homonomy and polysemy in information retrieval", In 35th Meeting of the Association for Computational Linguistics and the 8th Meeting of the European Chapter of the Association for Computational Linguistics (ACL/EACL-97), pages 72-78, Madrid, Spain, 1997.

[22] M. Sanderson, "Word sense disambiguation and information retrieval", In Proceedings, ACM Special Interest Group on Information Retrieval, pages 142-151, 1994.

[23] J. Hutchins and H. Sommers,"Introduction to Machine Translation", Academic Press, 1992.

[24] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, "Word sense disambiguation using statistical methods", In Proceedings of the 29th Meeting of the Association for Computational Linguistics (ACL-91), pages 264-270, Berkley, C.A., 1991.

[25] Agirre, E. and Rigau, G., "Word sense disambiguation using conceptual density", In Proceedings of COLING'96, 1996.

[26] Resnik, P., "Using information content to evaluate semantic similarity in a taxonomy", In Proceedings of IJCAI, 1995.

[27] Richardson, R. and Smeaton, A., "Using wordnet in a knowledge-based approach to information retrieval", In Proceedings of the BCS-IRSG Colloquium, Crewe, 1995.

[28] Sussna, M., "Word sense disambiguation for free-test indexing using a massive semantic network", Proceedings of the 2nd International Conference on Information and Knowledge Management. Arlington, Virginia, USA, 1993.

[29] Voorhees, E., "Using WordNet to Disambiguate Word Senses for Text Retrieval", SIGIR-93, 1993.

[30] Cowie, J. and Lehnert, W., "Information extraction", Communications of the ACM, 39(1):80-91.

[31] J. Guthrie, L. Guthrie, Y. Wilks and H. Aidinejad, "Subject-Dependent Co-Occurrence and Word Sense Disambiguation", ACL-91, pp. 146-152, 1991.

[32] David Yarowsky, "Word-sense disambiguation using statistical models of Roget's categories trained on large corpora", In proceedings of COLING-92,1992.

[33] Y. Wilks and M. Stevenson, "The Grammar of Sense: using part-of-speech tags as a first step in semantic disambiguation", To appear in Journal of Natural Language Engineering, 4(3), 1997.

[34] A. Harley and D. Glennon, "Sense tagging in action: Combining different tests with additive weights", In Proceedings of the SIGLEX Workshop "Tagging Text with Lexical Semantics", pages 74-78. Association for Computational Linguistics, Washington, D.C., 1997.

[35] Melcuk, I. A., "Dependency Syntax: Theory and Practice", State University of New York Press, Albany, 1988.

[36] Segond, F., Schiller, A., Grefenstette, G., and Chanod, J., "An experiment in semantic tagging using hidden markov model tagging", In Vossen, P., Adriaens, G., Calzolari, N., Sanfilippo, A., and Wilks, Y., editors, Proceedings of the ACL/EACL'97 Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources, 1997.

[37] Gale, Church and Yarowsky, "A Method for Disambiguating Word Senses in a Corpus", Computers and the Humanities, 26, pp. 415-439, 1992.

[38] R. Catizone, G. Russell, and S. Warwick, "Deriving translation data from bilingual texts", In Proceedings of the First International Lexical Acquisition Workshop (AAAi-89), Detriot, Michigan, 1989.

[39] Gale, Church and Yarowsky, "Using Bilingual Materials to Develop Word Sense Disambiguation Methods", In Proceedings of TMI-92, pp. 101-112, 1992.

[40] D. Yarowsky, "One sense per collocation", In Proceedings ARPA Human Language Technology Workshop, pages 266-271, Princeton, NJ, 1993.

[41] Radford, I. , Ananiadou, S. & Tsujii, J., "Adding structural constraints to lexically based context matching techniques", Int. Joint Conf. on Artificial Intelligence (IJCAI-97), Nagoya, Japan, 1997.

[42] T. Pedersen and R. Bruce, "Distinguishing word senses in untagged text", In Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, Providence, RI, August 1997.

[43] Yarowski, David, "Unsupervised word sense disambiguation rivaling supervised methods", In proceedings of the 33rd Annual Meeting, pp. 189-196, Cambridge, MA, Association for computational Linguistics, 1995.

[44] R. Rivest, "Learning decision lists. Machine Learning", 2(3):229-246, 1987.

[45] Hassan M. S., Reda M. S. A. A., Araby I. M., "Formation of Machine Groups and Part Families: A modified SLC Method and Comparative Study", *Integrated Manufacturing Systems*, pp. 123-137, 2003.

[46] Salton G., Wong A., and Yang C.S., "A Vector Space Model for Automatic Indexing", Communications of the ACM, 18:11, Pag. 613-620, November 1975.

[47] Salton G., "Automatic Text Processing", Addison Wesley Publishing Company, 1989.

[48] Sneath, P.H., "Some Thoughts of Bacterial Classification", Journal of General Microbiology, Vol. 17, pp. 184-200, 1957.

[49] A. Luk, "Statistical sense disambiguation with relatively small corpora using dictionary definitions", In Proceedings of the 33rd Meetings of the Association for Computational Linguistics (ACL-95), pages 181-188, Cambridge, M.A., 1995.

[50] David Pinto, Fernando Tellez, "Identificación de Términos Multipalabras", Segundo Congreso de Computación, FCC-BUAP, 2004.

[51] German Rigau, "Desambiguación autómatica del sentido de las palabras", en Tecnologías del Texto y del Habla, M. Antonia, Martín y Joaquim, Llisterri (Eds.). Edicions Universitat de Barcelona, Fundación Duques de Soria, 2004.

[52] Grigori Sidorov, Alexander Gelbukh, "Word sense disambiguation in Spanish explanatory dictionary", Proc. of TALN-2001 (Tratamiento automático de lenguaje natural), Tours, France, pp 398-402, Julio 2001.

[53] Alexander Gelbukh, Grigori Sidorov, and Liliana Chanona hernández, "Compilation of a Spanish representative corpus", Lecture Notes in Computer Science N 2276, Springer-Verlag, pp 285-288, 2002.

[54] Rada Mihalcea, Timothy Chklovski and Adam Killgariff, "The Senseval-3 English Lexical Sample Task", In Procedings of ACL/SINGLEX Senseval-3, Barcelona, Spain, July 2004.

[55] Hwee Tou Ng, "Getting Serious about Word Sense Disambiguation", SIGLEX,1997.

[56] Phillip Edmonds, "SENSEVAL: The Evaluation of word sense disambiguation systems", ELRA Newsletter. Vol. 7 No. 3, 2002.

[57] Héctor Jiménez Salazar, Guillermo Morales Luna, "Domain membership Degrees and Classification methods", Computación y Sistemas, vol. 5 No. 4 pp 288-295, México, 2002.

[58] Josefina García F, "Estructura conceptual y comunicación", Dimesión Antropológica, Año 2 vol. 3 pag 75-84, México, 1995.

[59] Adam Kilgarriff,"SENSEVAL: An Exercise in Evaluating Word Sense Disambiguation Programs", Proceedings of the International Conference on Language Resources and Evaluation (LREC), Granada, Spain, pp. 581-588, 1998.

[60] Davide Buscaldi, Manuel Montes y Gomez and Paolo Rosso,"Web-based WSD using Adjective-Noun pairs", Workshop on Lexical Resources and the Web for Word Sense Disambiguation. IX Ibero-American Conference on Artificial Intelligence IBERAMIA 2004, Puebla, Mexico, November, 2004.

[61] Michel Galley, Kathleen McKeown,"Improving Word Sense Disambiguation in Lexical Chaining", In the proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August 2003.

[62] Fellbaum, C. "WordNet: An electronic lexical database", Cambridge, Massachusetts: MIT Press,1998.

[63] CL Research, http://www.clres.com/, Ultima revision: Junio 2005.

[64] Banerjee, Satanjeev and Ted Pedersen, "An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet", In: Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING-02), Mexico City, February, 2002.

[65] "Wordnet bibliography", http://engr.smu.edu/ rada/wnb/, keywords="sense disambiguation".

[66] Castelló, G., N. Artigas, M.A. Martıy M. Taulé, Guía Diccionario CLIC-The Xtracct2-WP-03/Barcelona: CLiC-Thera, 2003.

[67] Chris Manning and Hinrich Schütze, "Foundations of Statistical Natural Language Proccesing", MIT Press. Cambridge, MA: May 1999.

[68] Ted Pedersen, "A Baseline Methodology for WSD", Proc. of 3rd. Int Conf. on Intelligent Text Proccesing and Computational Linguistics, CICLing, México, 2002.

[69] Bart Decadt, Véronique Hoste, Walter Daelemans, Antal van den Bosch,"GAMBL, Genetic Algorithm Optimization of Memory-Based WSD", SENSEVAL-3: 3rd. wordshop on the evaluation of systems for the semantic analysis of text, ACL, 2004.

[70] Altaf Mohammed, Dan Moldovan, and Paul Parker, "SENSEVAL-3 Logic Forms: A system and possible improvements", 3rd. wordshop on the evaluation of systems for the semantic analysis of text, ACL, 2004.

[71] Hee-Cheol Seo, Hae-Chang Rim, and Soo-Hong Kim,"KUNLP system in SENSEVAL-3", 3rd. wordshop on the evaluation of systems for the semantic analysis of text, ACL, 2004.

[72] Claudia Leaccok, Martín Chodorow, and George A. Miller, "Using corpus statistics and WordNet relations for sense identification", Computational Linguistic, 24(1): 147-145, 1998.

[73] E. Agirre and G. Rigau, "A proposal for Word sense disambiguation using conceptual distance", Proceeding of international conference on recent advances in NLP, 1995.

[74] P. Rosso, F. Masulli, D. Buscaldi, F. Pla, A. Molina, "Automatic Noun Disambiguation", Lecture Notes in Computer Science, vol. 2588, Springer Verlang, 2003.

[75] A. Montoyo, "Método basado en marcas de especificidad para WSD", procesamiento en lenguaje natural, revista No. 26, 2000.

[76] M. Carpuat, D. Wu, "Word Sense Disambiguation vs. Statistical Machine Translation", 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005). Ann Arbor, MI: Jun 2005.

[77] Enciclopedia online en Internet, "Clasificación clásica de los algoritmos de agrupamiento", http://en.wikipedia.org/wiki/Cluster_analysis, última revisión: Junio 2005.