Capítulo 1. Introducción

Una de las principales barreras que tienen que superar los sistemas de Procesamiento de Lenguaje Natural (PLN) es la que han impuesto las propias barreras geográficas entre los países, con numerosos idiomas o lenguajes naturales diferentes. Cada vez es más frecuente que investigadores y desarrolladores de sistemas de PLN se vean involucrados en tareas que implican nuevos sistemas en lenguajes extranjeros, que muchas veces son sencillamente desconocidos o bien para los cuales no existen recursos lingüísticos automáticos suficientes o, si existen, son sistemas privados patentados por compañías y no están disponibles al público. Se calcula que en la actualidad existen en todo el mundo unos 200 idiomas con importancia suficiente como para que su tratamiento automático sea interesante y por tanto ya se esté llevando a cabo en mayor o menor grado. Evidentemente, entre las causas de la aparición de este fenómeno está la interconexión creciente entre personas e instituciones de países diferentes y hasta hace poco distantes debida a la enorme difusión de la Web y de las tecnologías de la información en general.

El llamado *cuello de botella de la adquisición de conocimiento* (en adelante CBAC, en inglés *knowledge acquisition bottleneck*) consiste en la necesidad de codificar o compilar conocimiento lingüístico en un determinado idioma para permitir que un determinado sistema de PLN automático funcione correctamente en ese idioma. Esta codificación se suele realizar casi exclusivamente por parte de personal especializado en lingüística (por ejemplo lexicógrafos) para garantizar un cierto nivel de calidad. El coste de este proceso en términos de tiempo invertido y en términos puramente económicos es evidentemente muy elevado.

En este sentido, podría utilizarse la palabra compilación (de conocimiento lingüístico) para referirse a la elaboración de diccionarios, tesauros y bases de datos léxicas en general y la palabra codificación para referirse al etiquetado manual de corpus de texto de entrenamiento.

Resulta evidente que el etiquetado de un corpus de texto para el entrenamiento de un sistema de PLN supervisado constituye un ejemplo clásico del CBAC cuando se intenta aplicar ese sistema a un idioma diferente. La aplicación a un dominio diferente dentro del mismo idioma también se puede considerar un caso del CBAC.

La compilación de bases de datos léxicas automáticas es un caso ligeramente diferente pero también se puede incluir en el CBAC cuando por ejemplo se intenta trasladar una de ellas a otro idioma. Resulta sorprendente lo complicada que puede llegar a ser esta adaptación, incluso cuando se dispone de especialistas y de recursos automáticos lingüísticos en ambos idiomas.

El problema se ve agravado por el hecho de que en muchas de las tareas más importantes de la Lingüística de Corpus (LC) los sistemas más eficaces sean precisamente los que utilizan aprendizaje supervisado, es decir, corpus de entrenamiento etiquetados manualmente.

Todo esto ha llevado a muchos investigadores a la búsqueda de métodos para resolver o por lo menos aliviar en cierta medida el problema. Comprobado que un sistema completamente no supervisado suele ser factible pero raras veces eficaz, se ha recurrido a los llamados sistemas semisupervisados o débilmente supervisados, entre los que destacan los llamados algoritmos de autoarranque (en inglés *bootstrapping*). La idea que hay detrás de los sistemas débilmente supervisados consiste en utilizar tan poca información de entrenamiento, llamada muchas veces semilla, que el sistema se libere del CBAC, pero a la vez esta información sea suficiente como para superar con creces a los algoritmos no supervisados y en muchos casos alcanzar a los plenamente supervisados. Para lograr esto, los algoritmos llamados de autoarranque hacen uso de un proceso que se realimenta a sí mismo a partir de unas semillas mínimas pero que crece muy rápidamente hasta llegar a etiquetar toda la información necesaria casi sin intervención manual externa.

Desde este punto de vista, el interés de los algoritmos semisupervisados estaría fundamentalmente en lograr alcanzar o aproximarse a la precisión de los algoritmos plenamente supervisados, pero sin padecer el problema del CBAC. Sin ser este bajo ningún concepto un objetivo insuficiente, un análisis más detallado de la naturaleza de los corpus de texto general podría llevar incluso a potenciar la utilidad de los algoritmos semisupervisados, al menos para ciertas aplicaciones concretas.

Si se analiza cuidadosamente la naturaleza de los corpus de texto general se puede llegar a la conclusión de que existe un factor adicional que contribuye a fomentar el problema del CBAC en una aplicación clásica como puede ser la Desambiguación del Sentido de las Palabras (en adelante, DSP, en inglés, *Word Sense Disambiguation*, WSD), a saber, las fluctuaciones de dominio entre corpus diferentes (en el mismo idioma) y dentro de un mismo corpus de tamaño suficiente. Desde el punto de vista de los algoritmos supervisados, estas fluctuaciones hacen que si el corpus de entrenamiento etiquetado no

coincide en dominio con el corpus donde se lleva a cabo la tarea de desambiguación, la eficacia del algoritmo, medida como precisión, disminuye considerablemente.

Este problema no es exclusivo de los algoritmos plenamente supervisados, sino que afecta en igual medida a los algoritmos semisupervisados. Sin embargo, la naturaleza especial del algoritmo de autoarranque de éstos hace que puedan resultar lo suficientemente flexibles como para poder adaptarse a los cambios de dominio de forma más eficaz que los supervisados. En este punto el interés de los algoritmos semisupervisados rebasaría la neutralización del CBAC para llegar a poder superar en precisión a los algoritmos plenamente supervisados.

Sin embargo, como cabría esperar, no todo serían ventajas para el aprendizaje débilmente supervisado: los algoritmos de autoarranque tienen, debido precisamente a su naturaleza especial, una serie de problemas exclusivos de su clase. Estos problemas se pueden resumir en dos y, a su vez, uno de ellos se puede considerar causa del otro. El problema principal sería, en una tarea de DSP, la proliferación de ocurrencias de la palabra bajo desambiguación que pueden resultar mal etiquetadas en una fase temprana del algoritmo. Debido al funcionamiento del algoritmo de autoarranque, que se realimenta a sí mismo, estos errores iniciales pueden provocar el 'descarrilamiento' temprano del algoritmo y su fracaso final, consistente en la consecución de una precisión bastante baja. Este problema está provocado, a un nivel más cercano a la codificación del algoritmo, por el llamado problema del ajuste inicial de parámetros del algoritmo. Estos parámetros iniciales del algoritmo, de número muy reducido, se deben fijar inicialmente de forma relativamente precisa, y si se logra esto, el algoritmo puede funcionar de forma óptima. El problema es que estos parámetros dependen de otras entradas del algoritmo, como puede ser la palabra bajo desambiguación. En otras palabras, diferentes palabras bajo desambiguación requieren diferentes valores de los parámetros iniciales y, además, estos valores no se conocen en general a priori. Es decir, si el algoritmo se quiere ejecutar para más de una palabra, lo cual sería perfectamente normal, y se hace un ajuste global de parámetros, muchas de las palabras bajo desambiguación trabajarían con parámetros mal ajustados, y la precisión de desambiguación total media final no sería óptima.

El objetivo de este trabajo es pues doble: por un lado, y debido a que el problema de las fluctuaciones de dominio afecta de forma más virulenta a los sistemas semisupervisados que a los plenamente supervisados, lograr un nuevo algoritmo de autoarranque que consiga que un sistema semisupervisado se libere realmente del problema del CBAC, igualando, como mínimo, la precisión de los sistemas supervisados; y por otro lado, aprovechar las características especialmente apropiadas de los algoritmos de autoarranque en relación con las variaciones de dominio para conseguir que ese nuevo algoritmo propuesto pueda incluso superar la precisión de los sistemas plenamente supervisados sobre texto general en condiciones prácticas; para lo cual el nuevo algoritmo debe tratar, además de con los cambios de dominio, con los problemas exclusivos de los sistemas de autoarranque.

Este documento está organizado de la siguiente forma. En el capítulo 2 se realiza una exposición del estado del arte en términos generales en relación con el algoritmo de Yarowsky. Se analiza el tipo de conocimiento lingüístico que utiliza este algoritmo, y se le describe desde el punto de vista de los diferentes tipos de algoritmos de DSP: como algoritmo basado en conocimiento, como algoritmo no supervisado, como algoritmo semisupervisado y como algoritmo supervisado. También se analiza el estado del arte de la DSP en relación a los cambios de dominio, en cuanto a la evaluación de estos sistemas y en cuanto a las aplicaciones reales de PLN que la utilizan. En el capítulo 3 se analiza con cierto detalle el problema del CBAC y los métodos más utilizados en la actualidad para combatirlo, incluyendo los algoritmos de autoarranque (en inglés *bootstrapping*). El capítulo 4 trata sobre la importante cuestión en DSP que consiste en determinar el nivel de granularidad óptimo de los sentidos de las palabras bajo desambiguación y hace especial énfasis en la necesidad de reducir ese nivel hasta una distinción de sólo dos sentidos en las aplicaciones reales de PLN. En el capítulo 5 se hace una descripción pormenorizada del algoritmo de Yarowsky standard tal como apareció por primera vez en el artículo de 1995 y se presentan los resultados de precisión obtenidos por ese algoritmo. El capítulo 6 analiza la fuerte dependencia de las colocaciones, que son la principal fuente de conocimiento lingüístico que utiliza este algoritmo, de las fuentes de dominio del corpus, es decir, que si hay fluctuaciones de dominio en el corpus, lo cual es muy normal, también habrá variaciones en las colocaciones alrededor de la palabra objetivo. En el capítulo 7 se describe brevemente el sistema informático que se ha utilizado para la realización de los experimentos originales del trabajo. El capítulo 8 se dedica a analizar las correspondientes dependencias del dominio del propio algoritmo de Yarowsky, algo que sucede lógicamente de lo descrito en los dos capítulos anteriores. E1 capítulo 9 realiza una investigación a fondo de la distribución estadística de las fuentes de dominio en los corpus de texto general y de texto periodístico y establece importantes diferencias entre la naturaleza de uno y otro tipo de corpus de texto y en el comportamiento de los algoritmos de DSP en ellos. En el capítulo 10 se introduce un nuevo algoritmo de autoarranque semisupervisado que libera a la DSP del problema del CBAC en corpus de texto general, logrando como mínimo la misma precisión que los algoritmos supervisados que sí padecen ese problema, e incluso superando esa precisión, al menos potencialmente. Finalmente en el capítulo 11 se resumen las principales conclusiones de este trabajo y se indican futuras líneas de investigación que surgen con naturalidad a partir de los resultados obtenidos.

Capítulo 2. Estado actual de la Desambiguación del Sentido de las Palabras

Este capítulo trata la situación de las distintas líneas de investigación que en la actualidad se dedican al estudio del algoritmo de Yarowsky, con especial atención a la relación entre la Desambiguación del Sentido de las Palabras (DSP) y las variaciones de dominio entre los corpus de texto, con la evaluación de los sistemas de DSP y con la utilización de estos sistemas en aplicaciones de PLN.

El contenido del capítulo está organizado de la siguiente forma. En la sección 2.1 se hace una descripción sistemática de las fuentes de conocimiento lingüístico (CL) que se utilizan en general en DSP, tanto desde un punto de vista más abstracto como desde el punto de vista más cercano a la implementación concreta de los sistemas. La descripción de esas fuentes de CL en el caso concreto del algoritmo de Yarowsky se realiza en el capítulo 5. En las secciones 2.2 a 2.5 se considera el algoritmo de Yarowsky en relación con cada uno de los tres tipos generales de algoritmos de DSP (supervisados, no supervisados y basados en conocimiento). En el caso concreto de la sección 2.4 se considera el algoritmo como un caso de método de DSP semisupervisado, que es la categoría donde se le clasifica con mayor exactitud. De hecho el algoritmo de Yarowsky es el pionero y uno de los principales representantes de este tipo de métodos de DSP. En la sección 2.6 se revisan las principales metodologías que se utilizan en la actualidad para representar los dominios y temas en los corpus de texto general. La sección 2.7 está dedicada a los métodos de evaluación de los sistemas de DSP, prestando atención a la evaluación en las tres primeras ediciones del certamen Senseval. Finalmente la sección 2.8 revisa el estado del arte de los sistemas de DSP en aplicaciones reales de PLN.

2.1 El conocimiento lingüístico que utiliza el algoritmo de Yarowsky

2.1.1. Conocimiento lingüístico para la DSP

Desde el inicio de la investigación en DSP se han observado diversos fenómenos lingüísticos que han sido utilizados por los sistemas de desambiguación. Estos fenómenos lingüísticos se denominan a veces *knowledge sources (KS)* en la literatura en inglés. Nosotros los denominaremos *conocimiento lingüístico (CL)*, siguiendo a [Agirre y Stevenson 2006].

El CL describe los fenómenos lingüísticos explotados en DSP desde un punto de vista más bien teórico, a diferencia de los *rasgos (features)* utilizados en las descripciones a nivel algorítmico de los sistemas reales. No existe una correspondencia biunívoca entre ambas descripciones, es decir, el sistema de rasgos de una aplicación real puede en realidad estar usando varias categorías de CL y a la inversa.

Existen tres tipos básicos de CL: sintáctico, semántico y pragmático/tópico [Stevenson y Wilks 2001]. Los fenómenos de tipo sintáctico se basan en la función de las palabras dentro de la estructura gramatical de las oraciones. Los fenómenos semánticos están relacionados con propiedades de las cosas a las que hacen referencia las palabras. Los pragmático/tópicos tienen que ver con el papel de las palabras en un contexto amplio, es decir, con el sentido común o el tópico del texto considerado.

CL sintáctico

Categoría gramatical (Part of speech, KS 1)

La categoría gramatical de las palabras (sustantivo, adjetivo, verbo, adverbio, etc.) puede ser una herramienta poderosa para la desambiguación del sentido de las palabras. Esto ocurre sobre todo cuando una determinada palabra tiene diferentes sentidos que pertenecen cada uno a diferentes categorías. Sin embargo, como se verá en el capítulo 4 en el caso de los homógrafos es muy frecuente que diferentes sentidos de la palabra pertenezcan a la misma categoría gramatical [Durking y Manning 1989], con lo que esta información no resulta útil del todo en la desambiguación. Véase también [Wilks y Stevenson 1998] para un estudio sobre el uso la categoría gramatical junto con otras propiedades en la desambiguación.

Morfología (KS 2)

La estructura morfológica de una palabra también puede ser útil en el descarte de determinados sentidos de las palabras. Por ejemplo en la frase *los tiempos que corren*, el hecho de estar utilizando la forma plural de *tiempo* descarta los sentidos de tiempo

meteorológico y de tiempo físico. Sin embargo, la morfología por sí misma no es una herramienta muy poderosa de desambiguación y la mayoría de las veces formas de palabras idénticas, es decir sin variación morfológica, tienen más de un sentido, tanto a nivel de granularidad fina (polisemia fuerte) como de granularidad gruesa (polisemia débil o de nivel homográfico).

Colocaciones (Collocations, KS3)

Las colocaciones se pueden definir como "cualquier ocurrencia conjunta estadísticamente significativa de palabras" [Sag et al. 2002]. Por ejemplo en la frase *tiempo borrascoso* el hecho de que el homónimo *tiempo* esté sucedido por *borrascoso* desambigua el sentido del homónimo descartando el sentido físico en favor del meteorológico. Las colocaciones son una herramienta de desambiguación muy útil incluso para el caso de los homónimos. En particular, como se verá en el capítulo 5, son el CL constituyente de la propiedad *one-sense-per-collocation* (un sentido por colocación) que es la más importante utilizada por el algoritmo de Yarowsky.

Subcategorización (KS 4)

La información de subcategorización de una palabra consiste en los argumentos y adjuntos, desde un punto de vista sintáctico, que puede tener. Esto puede servir claramente para desambiguar su sentido como por ejemplo en el caso del verbo *pegar*; es transitivo en el sentido de 'pegar con pegamento' e intransitivo en el sentido de la 'agresión física'.

CL semántico

Frecuencia de los sentidos (KS 5)

La información sobre la distribución a priori de los sentidos de una palabra puede ser útil [McCarthy et al. 2004]. Por ejemplo, de los 4 sentidos de la palabra *people* en Wordnet 1.6 [Fellbaum 1998] sólo uno de ellos supone el 90% de las ocurrencias de la palabra en el corpus etiquetado Semcor [Miller et al. 1993]. Esta información puede servir para implementar un algoritmo básico de desambiguación, que consiste en asignar siempre el sentido más frecuente. Este algoritmo se suele utilizar como línea de base en las métricas de evaluación formal de los sistemas de DSP (véase la sección 2.7.4).

Asociaciones semánticas entre las palabras (KS 6)

Estas asociaciones semánticas son relaciones entre los significados de los sentidos de las palabras. Se pueden clasificar en dos grandes tipos: relaciones paradigmáticas y sintagmáticas [Kriedler 1998].

Relaciones paradigmáticas (KS 6a). La hiperonimia es una relación de este tipo. Consiste en la relación entre un sentido de una palabra y otro sentido de otra palabra más general respecto de la primera. Por ejemplo en la frase *El plátano es un tipo de árbol común en Canarias* existe una relación de hiperonimia entre *plátano* y *árbol*. Esta relación sirve para desambiguar plátano descartando el sentido de 'fruto' y favoreciendo el sentido de 'planta'. Otra relación de este tipo es la meronimia. En este caso hay una relación de parte a todo entre dos sentidos; por ejemplo la que hay entre *hoja* y *planta* que serviría para descartar el sentido de 'hoja de papel' y determinar el de 'hoja vegetal'.

Relaciones sintagmáticas (KS 6b). Estas relaciones describen asociaciones entre palabras respecto a ciertas dependencias sintácticas entre ellas. Por ejemplo en la frase *El niño pegó un sello* el hecho de que el verbo *pegar* tenga ese objeto directo sirve para determinar el sentido de 'pegamento' frente al de 'agresión'. Este tipo de CL puede considerarse como un caso especial pero más restrictivo de colocación (KS 3) en el que se exige además de la coocurrencia que haya cierta relación sintáctica. También puede considerarse como un tipo de subcategorización (KS 4). Se incluye entre el CL semántico para enfatizar que existe una relación semántica entre estas palabras (*pegar* y *sello* en el ejemplo).

Preferencias selectivas (KS 7)

Los verbos y los adjetivos aceptan vocablos de determinados tipos semánticos para ocupar el lugar de sus argumentos [Cruse 1986]. Si sabemos esta información y los diferentes tipos semánticos de una palabra que ocupe este lugar como argumento, podemos desambiguar los sentidos del verbo y de su argumento. Por ejemplo, el verbo *blandir* acepta argumentos de los tipos semánticos 'arma' o 'ataque'. Por su parte, la palabra *argumento* puede pertenecer a los grupos semánticos 'función-matemática' o 'discusión'. Si nos encontramos con la frase *Alguien blandió un argumento contra él* podemos desambiguar el sentido de blandir ('ataque') y el de argumento ('discusión'). Si tenemos la frase *blandió el filo* podemos desambiguar el sentido de blandir ('arma') y el de filo ('arma') frente a, por ejemplo, 'límite'.

Todo esto significa que se deben manejar los posibles tipos semánticos de los argumentos de una palabra (verbo o adjetivo) y los tipos semánticos de las propias palabras que van a ocupar esos huecos. Esto hace que esos tipos semánticos estén organizados en jerarquías [Wilks 1975].

Las relaciones sintagmáticas (KS 6b) vistas anteriormente se diferencian de las preferencias selectivas en que los huecos pueden ser ocupados por conjuntos de palabras concretas en lugar de por grupos semánticos abstractos que generalizan sobre los grupos de palabras.

Roles semánticos (KS 8)

Son una serie de roles importantes que siempre aparecen en cualquier oración, como por ejemplo 'experimentador', 'tema', etc. [Fillmore 1971] Normalmente el experimentador es el sujeto, pero puede ocurrir que no: en *Las malas noticias le van a consumir a Pedro*, es el objeto indirecto el que juega el rol de experimentador. Este hecho puede ayudar a desambiguar el sentido de *consumir*, junto con sus preferencias selectivas.

CL pragmático/tópico

Dominio (KS 9)

El conocimiento del dominio puede ser muy útil para desambiguar una palabra en un texto. Por ejemplo, si se sabe que un texto versa sobre el dominio *deportes* y aparece la palabra *campo* es probable que el sentido sea 'estadio' en lugar de 'campo de cultivo'. Se supone que el dominio es una etiqueta extraída de una lista obtenida externamente del texto. Esta lista podría ser por ejemplo una lista de metadatos de noticias, una lista de códigos de un diccionario, o podría ser producida automáticamente por un sistema de categorización de texto.

Asociación de palabras por temas (Topical word association, KS10)

Normalmente las palabras que aparecen juntas en un texto están relacionadas por un tema (en inglés topic). Al mismo tiempo, no siempre se dispone de información explícita sobre el dominio o tema de un texto (KS 9). Sin embargo, sí se puede saber qué parejas de sentidos de palabras tienden a aparecer en textos del mismo tema (KS 10). Este CL es distinto de las asociaciones paradigmáticas (KS 6a) ya que las palabras relacionadas no tienen por qué pertenecer al mismo tipo ontológico. También es diferente de KS 6b porque en éste es necesario que haya una relación sintáctica y además las palabras no tienen por qué pertenecer al mismo dominio (tema). Lo mismo ocurre con KS 3 (colocaciones): éstas no tienen por qué estar asociadas con ningún tema en particular.

Pragmática (KS 11)

En algunos casos es necesario utilizar el sentido común y razonar con él para poder resolver una ambigüedad. Sin embargo, estos casos extremos son poco frecuentes y en la mayoría de los casos es posible desambiguar utilizando CL más básico.

2.1.2 La codificación del conocimiento lingüístico mediante rasgos

La sección anterior describe los tipos de conocimiento lingüístico (CL) utilizado en sistemas de DSP desde un punto de vista más bien teórico. Sin embargo los sistemas

DSP reales necesitan codificar a nivel algorítmico ese conocimiento, y ese conocimiento tiene que ser extraído de alguna forma de un recurso real. Existen básicamente tres tipos de recursos utilizados para extraer ese conocimiento: corpus (etiquetados con sentidos o no etiquetados), Diccionarios en Formato Electrónico DFE (Machine Readable Dictionaries, MRDs) y Bases de Conocimiento Léxico BCL (Lexical Knowledge Bases, LKBs).

El conocimiento lingüístico adquirido de estos recursos se codifica en forma de rasgos (features). Los rasgos codifican uno o más tipos de conocimiento lingüístico (CL). A continuación aparece una lista de 11 rasgos obtenidos de estos recursos y su relación con los tipos de CL vistos en la sección anterior. Los 11 rasgos están clasificados en tres tipos diferentes según el rango del contexto alrededor de la palabra objetivo que utilizan: rasgos específicos de la palabra objetivo, rasgos locales y rasgos globales.

Rasgos de la palabra objetivo

1.- La forma de la palabra objetivo. Este rasgo puede en parte codificar la categoría gramatical (KS 1) y la morfología (KS 2).

2.- La categoría gramatical de la palabra objetivo. Este rasgo codifica directamente KS 1. Este rasgo es fácil de obtener en DFEs y BCLs. En contexto (corpus etiquetado o no etiquetado con sentidos) puede obtenerse también fácilmente utilizando varios etiquetadores (*part of speech taggers*) como el etiquetador Brill [Brill 1995].

3.- La distribución de los sentidos de la palabra objetivo. Este rasgo codifica directamente KS 5. Se puede obtener en teoría directamente de un corpus etiquetado con sentidos. Sin embargo, este enfoque adolece del problema CBAC (Cuello de Botella de la Adquisición de Conocimiento, en inglés KAB, *Knowledge Acquisition Bottleneck*) y además es muy variable con el corpus utilizado. Muchos DFEs y BCLs intentan incluir este rasgo, pero la falta de recursos reales y fiables hace que sus resultados sean también subjetivos.

Rasgos locales

4.- Los patrones locales. Este es uno de los rasgos más usados por los sistemas de DSP. Codifica varios tipos de CL: colocaciones (KS 3), subcategorización (KS 4) y asociaciones semánticas sintagmáticas (KS 6b). Su forma depende del alcance y del contenido. El alcance puede ser: n-gramas alrededor de la palabra objetivo, palabra n a la izquierda o a la derecha de la palabra objetivo o palabra n a la izquierda o a la derecha de la palabra objetivo con cierta propiedad. El contenido puede ser palabras completas, lemas de palabras, categorías gramaticales de palabras o una combinación de cualquiera de ellos.

Este rasgo precisa en condiciones normales del uso de un corpus etiquetado con sentidos por lo que casi todos los sistemas que lo usan son supervisados.

5.- La subcategorización. Este rasgo codifica el CL que lleva el mismo nombre (KS 4) y se suele obtener de un corpus etiquetado con sentidos utilizando un parser (analizador sintáctico) robusto [Lin 1993] [Carroll y Briscoe 2001]. Por ejemplo, de *El desafortunado escalador cayó-1 en una grieta* podemos inferir que este sentido de *cayó* admite un sujeto pero ningún objeto directo o indirecto.

6.- Las dependencias sintácticas. Este rasgo codifica las relaciones semánticas sintagmáticas (KS 6b). Esta información puede obtenerse, para un sentido de una palabra dado, de un corpus etiquetado con sentidos y analizado sintácticamente con un parser. [Lin 1997] [Yarowsky y Florian 2002].

7.- Las preferencias selectivas. Este rasgo codifica el CL que lleva el mismo nombre (KS 7). Las preferencias selectivas aparecen incluidas en algunos DFEs y BCLs, al menos en parte. Se han propuesto métodos para obtenerlas automáticamente de corpus etiquetados con sentidos sólo [Resnik 1997] [McCarthy et al. 2001] o etiquetados con sentidos y analizados sintácticamente [Agirre y Martinez 2001b].

Rasgos globales

8.- La bolsa de palabras. Este rasgo consiste en una lista de palabras (o bigramas) y sus frecuencias dentro de una ventana grande alrededor de la palabra objetivo. Codifica parcialmente las asociaciones de palabras semánticas y tópicas (KS 6b y KS 10) así como la información de dominio (KS 9). Se puede obtener de un corpus sin ningún tipo de procesamiento.

9.- La relación con palabras del contexto. Este rasgo se obtiene de las definiciones de las palabras en un diccionario. De estas entradas se obtienen palabras que suelen aparecer en los contextos de la palabra objetivo cuando se usa en un determinado sentido. Este rasgo codifica parcialmente KS 6b y KS 10 así como KS 9. Su utilización fue propuesta por primera vez en 1986 por Lesk [Lesk 1986].

10.- La similitud con palabras del contexto. Este rasgo codifica las relaciones paradigmáticas (KS 6a). Las relaciones paradigmáticas entre la palabra objetivo y las de su contexto se pueden obtener de las taxonomías de Wordnet, por ejemplo, y utilizar para medir la similitud entre sus sentidos [Patwardhan et al. 2003].

11.- Los códigos de dominio. Este rasgo codifica la información de dominio (KS 9). El dominio más probable de un sentido de una palabra puede encontrarse en algunos recursos léxicos. Por ejemplo el diccionario LDOCE usa una lista de 100 códigos de dominio y 246 subdominios para indicar el dominio de cada sentido. Otros recursos como el Roget´s International Thesaurus [Chapman 1977] indican esta información

clasificando todas las palabras (y sus sentidos) en un total de unas 970 categorías. En realidad determinar este código, dada una palabra objetivo en un contexto (o corpus), es casi equivalente a desambiguarla (si la fuente léxica de códigos asigna uno diferente para cada sentido).

2.2. El algoritmo de Yarowsky como método de DSP basado en conocimiento

Existen tres categorías generales de métodos de DSP, cada una determinada por la estrategia que adopta el sistema para llevar a cabo la desambiguación. Una de estas categorías es la formada por los llamados métodos basados en conocimiento. Las otras dos están formadas por los llamados métodos basados en corpus y, dependiendo de si estos corpus están etiquetados o no, reciben el nombre de métodos supervisados y de métodos no supervisados. Normalmente los corpus etiquetados de los métodos supervisados se utilizan como forma de entrenamiento del sistema. El algoritmo de Yarowsky es un caso muy importante de una cuarta categoría conocida en general como de métodos basados en corpus semisupervisados, debido a la pequeña cantidad de entrenamiento que precisan, lo cual es una de sus principales ventajas.

Los métodos basados en conocimiento forman una categoría aparte de métodos de DSP y son aquellos que utilizan una fuente externa (distinta del propio corpus objetivo, sea éste etiquetado o no) de conocimiento léxico, ya sea un Diccionario en Formato Electrónico (DFE) o una Base de Conocimiento Léxico (BCL) [Mihalcea 2006].

Los métodos basados en conocimiento adolecen del problema de CBAC, pero de una forma diferente y menos dramática que los métodos supervisados basados en corpus. Así, este problema les afecta relativamente poco desde el punto de vista del tamaño y el dominio, aunque en mucha mayor medida desde el punto de vista del idioma. Esto hace que, en comparación con los métodos supervisados, se puedan aplicar a una tarea de *todas las palabras* (*all words*) empleando terminología de Senseval, mientras que aquellos sólo se puedan aplicar a aquellas palabras para las que esté disponible un corpus etiquetado con sentidos. Como contrapartida, la eficacia de los métodos supervisados suele ser algo más alta que la de los métodos basados en conocimiento.

2.2.1 Métodos basados en conocimiento que utilizan el rasgo 9

Estos métodos se basan en el solapamiento entre el contexto de la palabra objetivo y los contextos de varios sentidos de esa palabra en sus definiciones tal como aparecen en un diccionario. El primer algoritmo de este tipo fue propuesto por Lesk en 1986 [Lesk 1986]. Su funcionamiento básico es el siguiente: dada una palabra ambigua (con varios sentidos en su entrada en un diccionario) que aparece en un corpus; para cada palabra que aparece en su contexto en el corpus, si esa palabra aparece en la entrada del diccionario correspondiente a un sentido, entonces añadir un valor (posiblemente ponderado) al peso de ese sentido sobre la palabra objetivo; al final seleccionar el sentido con el peso más alto.

Nótese que el esquema conceptual del algoritmo se puede aplicar a métodos supervisados (basados en corpus) sin más que cambiar las entradas de los sentidos en el diccionario por los contextos de las apariciones de la palabra objetivo en el corpus etiquetado.

2.2.2 Métodos basados en conocimiento que utilizan los rasgos 6 y 10

Estos métodos se basan en una medida de la similitud semántica entre dos conceptos a partir de su distancia en una red semántica como Wordnet según diferentes métricas [Butadinsky y Hirst 2001]. Normalmente se calcula la distancia o similitud entre todos los posibles sentidos de la palabra objetivo y las palabras de su contexto. Para reducir la carga computacional que puede llegar a ser muy alta si se calcula la distancia entre muchos pares de palabras, estos métodos suelen utilizar un contexto muy pequeño, o bien utilizar palabras relacionadas sintácticamente (rasgo 6).

La medida de la similitud semántica también puede realizarse en un contexto global mucho más amplio. Para ello pueden utilizarse las estructuras conocidas como cadenas léxicas (lexical chains). Una cadena léxica es una secuencia de palabras semánticamente relacionadas que crea un contexto y contribuye a la continuidad y la coherencia de un discurso [Halliday y Hasan 1976]. Este tipo de estructuras de significado se utilizan ampliamente en Procesamiento del Lenguaje Natural (PLN), por ejemplo en sumarización de textos, clasificación de textos o en DSP. Las cadenas léxicas son independientes de la estructura gramatical del texto, y pueden llegar a ser muy largas.

El algoritmo básico que siguen las cadenas léxicas es el siguiente:

1. Seleccionar las palabras candidatas del texto. Suelen ser palabras sobre las que podemos establecer una medida de similitud semántica. Casi siempre son los sustantivos.
2. Para cada palabra candidata, en orden, y para cada sentido posible de ella, encontrar la cadena léxica que maximice (y supere un umbral) una medida de similitud semántica entre ese sentido y los de cada cadena léxica ya iniciada.
3. Si existe esa cadena léxica, la palabra candidata se inserta en la cadena, y si no, se crea una nueva cadena con esa única palabra.

Al añadir una palabra a una cadena léxica estamos decidiendo entre uno de sus posibles sentidos, es decir, la estamos desambiguando.

2.2.3 Métodos basados en conocimiento que utilizan el rasgo 7

El rasgo 7 (preferencias selectivas) es uno de los primeros utilizados en DSP. Las preferencias selectivas son relaciones entre tipos de palabras que representan conocimiento conceptual tópico. Por ejemplo COMER-COMIDA y BEBER-LIQUIDO.

Estas relaciones pueden servir para descartar posibles sentidos que no estén de acuerdo con el sentido común representado por ellas. En la frase *María bebió tinto* se puede descartar inmediatamente el sentido de 'color oscuro' de la palabra 'tinto' y determinar el sentido 'bebida alcohólica'.

Las preferencias selectivas se pueden medir mediante conteo de frecuencias de las palabras de la relación, es decir, el número de veces que ocurren en una determinada relación en un corpus. La relación que las conecta suele ser sintáctica. Esta medida puede sofisticarse utilizando probabilidades condicionales. Estas medidas se pueden utilizar para aprender preferencias selectivas a partir de corpus no etiquetados, con lo cual la distribución de sentidos de las palabras es desconocida, y se supone uniforme; o bien pueden utilizarse corpus etiquetados con sentidos, en cuyo caso sí se conoce la distribución de los sentidos. Además, las medidas se pueden utilizar para aprender preferencias selectivas entre palabras concretas, entre clases semánticas o entre palabras concretas y clases semánticas enteras. Finalmente, se pueden utilizar preferencias selectivas entre clases semánticas no aprendidas a partir de corpus, sino obtenidas de taxonomías creadas manualmente como diccionarios.

Los resultados experimentales muestran que las preferencias selectivas por sí solas no suelen superar la línea de base marcada por el sentido más frecuente (por ejemplo, en Senseval-2 esta marca estaba en 57% para la tarea de todas las palabras). Estos resultados, junto con una exhaustividad (*recall*) baja indican que por ahora las preferencias selectivas pos sí solas necesitan poder aprenderse mejor.

2.2.4 Métodos heurísticos (rasgos 3, 4 y 8)

Distribución del sentido más frecuente (rasgo 3)

Los diferentes sentidos de una palabra suelen presentar una distribución "*Zipfiana*": hay un sentido dominante (más frecuente) y el resto de sentidos tienen menores frecuencias descendentes [Zipf 1949]. Por tanto un criterio muy simple de desambiguación es asignar a todas las ocurrencias de la palabra el sentido más frecuente.

Este método trivial tiene aplicación al menos como línea de base (*baseline*) para comparar otros sistemas en teoría más efectivos. Según [Gale et al. 1992] "los sistemas razonables deben superar esta línea de base".

Otros inconvenientes de este criterio como método *per se* serían: primero, que hacen falta estadísticas de distribución de todas las palabras en el idioma objetivo, algo que actualmente sólo ocurre con el inglés; segundo, cualquier cambio de dominio en el corpus objetivo puede alterar significativamente estas distribuciones [Martínez y Agirre 2000]. Véase también la sección 1.6.

Existe un método propuesto por McCarthy et al. [McCarthy et al. 2004] de este tipo que evita estos inconvenientes sin necesitar la existencia de corpus etiquetados con sentidos. El método funciona de la siguiente forma: dado un corpus con múltiples ocurrencias de una palabra objetivo, se utiliza el método distribucional de [Lin 1998] para obtener una serie de palabras relacionadas contextual y sintácticamente con esa palabra objetivo. Se toman las *k* palabras más próximas a ella para caracterizar el dominio en que ocurre. Después se usan las medidas de [Jiang y Conrath 1997] y [Banerjee y Pedersen 2003] para determinar el grado de similitud semántica entre la palabra objetivo y sus vecinas. El sentido de la palabra objetivo juzgado más similar al conjunto de palabras que representa al dominio se considera el sentido predominante en ese dominio. Se asigna a la palabra objetivo en todos los contextos en que ocurre en el corpus dado. Este sistema obtuvo en Senseval-2 una precisión cercana a la línea de base para la tarea de 'todas las palabras' (*all-words*), lo cual es un muy buen resultado, teniendo en cuenta que sólo dos sistemas participantes en ese certamen obtuvieron precisión por encima de la línea de base. Este método no necesita corpus etiquetados pero sí algún tipo de BCL como WordNet en el idioma inglés. Por lo tanto no es un método no supervisado, pero en la sección 2.3 lo revisaremos como complemento de ese tipo de métodos.

Un sentido por colocación (rasgo 4)

Esta propiedad fue introducida por Yarowsky en [Yarowsky 1993] y dice que los sentidos de las palabras tienden a tener las mismas *colocaciones* de palabras en su contexto. Una colocación es una palabra en el contexto cercano de la palabra objetivo: ventana de +/- k palabras a su alrededor, en posición adyacente hacia la izquierda (-1), hacia la derecha (+1), dos hacia la izquierda (-2), dos hacia la derecha (+2), etc.
Esta propiedad es más fuerte para colocaciones más cercanas a la palabra objetivo, y es cada vez más débil a medida que la colocación es más lejana.

En su trabajo original Yarowsky [Yarowsky 1993] consideró palabras muy polisémicas de sólo dos sentidos (homógrafos). Sus *test* sobre corpus etiquetados manualmente arrojaron unos resultados de precisión muy alta, 97%.

Sin embargo, experimentos más recientes de Martínez y Aguirre [Martínez y Agirre 2000] han dado resultados de precisión mucho más bajos, bajo dos puntos de vista: primero, desde el punto de vista del número de sentidos considerados; en concreto más de dos sentidos, como en la base léxica inglesa WordNet; y segundo, utilizando corpus realistas en los que haya variaciones o cambios de dominio. Como se verá en el capítulo 8, ambos aspectos influyen negativamente en el grado de precisión alcanzado y en la eficacia final del algoritmo de Yarowsky, como usuario de esta propiedad.

Un sentido por discurso (rasgo 8)

Esta propiedad fue introducida por Gale et al. [Gale et al. 1992]. La propiedad dice que una palabra tiende a tener un único sentido en un discurso, o documento, o nosotros

diríamos también, dominio. Esta propiedad sirve para determinar el sentido de una palabra en todo un 'discurso' (ya sea documento, u otra unidad en la que no haya un cambio de dominio). Esta propiedad también la utiliza el algoritmo de Yarowsy [Yarowsky 1995] descrito en el capítulo 5 y la metodología de *bootstrapping* propuesta en el capítulo 10.

En el trabajo original [Gale 1992] Gale sólo consideró ambigüedades de dos sentidos y obtuvo una probabilidad de que dos ocurrencias en el mismo discurso de una palabra tuvieran el mismo sentido del 98%. La misma probabilidad o precisión fue calculada por Yarowsky en [Yarowsky 1995] también sobre ambigüedades de dos sentidos y obtuvo un resultado de 99.8% (precisión) y 50.1% (aplicabilidad: si una palabra aparece en un discurso, cuál es la probabilidad de que aparezca más de una vez).

Krovetz [Krovetz 1998] ha hecho un experimento parecido pero sobre ambigüedad múltiple (más de dos sentidos) y obtuvo unos resultados de precisión bastante más bajos: algo menores de 70%.

2.3 El algoritmo de Yarowsky como método de DSP no supervisado

Los métodos de DSP basados en conocimiento de la sección anterior utilizan una fuente de conocimiento externa (distinta del propio corpus objetivo), tal como un diccionario, tesauro o base de conocimiento léxica. Los métodos supervisados (sección 2.5) utilizan una fuente externa de conocimiento de distinto tipo: una parte del corpus objetivo u otro corpus están etiquetados con los sentidos de sus palabras y se utilizan para entrenar el sistema. Tanto los métodos de un tipo como los del otro están sujetos al problema conocido como CBAC: ambas fuentes de conocimiento externas necesitan ser construidas y mantenidas manualmente por personas. Esto hace que el proceso sea lento y caro.

Los métodos de desambiguación no supervisados son aquellos que no utilizan ningún tipo de fuente de conocimiento externa salvo el corpus objetivo no etiquetado (como mucho otro corpus paralelo en otro idioma y también no etiquetado). Naturalmente, estos métodos no sufren el problema de CBAC. Estos métodos eliminan la dependencia de recursos manuales externos de dos formas diferentes: la primera, conocida como enfoque *distribucional* distingue los sentidos de la palabra objetivo basándose en la hipótesis de que palabras con el mismo sentido tienden a tener a su alrededor contextos similares [Harris 1968][Millar y Charles 1991]; la segunda, conocida como *equivalencia de la traducción* se basa en la utilización de *corpus paralelos*; estos son corpus del mismo contenido pero traducidos en diferentes lenguas; cuando una palabra polisémica se traduce a otra lengua es muy frecuente que diferentes sentidos se traduzcan a diferentes palabras en el lenguaje destino, lo cual sirve como criterio desambiguador [Pedersen 2006].

Otra característica importante de los métodos no supervisados es que no utilizan ningún tipo de inventario predeterminado de posibles sentidos de la palabra objetivo. Esto puede significar una gran ventaja, ya que tales inventarios, de existir, pueden no ser muy útiles o incluso contraproducentes, ya que la naturaleza y el número de sentidos a discriminar puede variar entre muchos tipos de aplicaciones.

Una tercera ventaja de estos métodos es que la creación automática (es decir no supervisada) de un corpus etiquetado, si es de alta calidad, puede servir de entrada para el entrenamiento de un método supervisado.

El algoritmo de Yarowsky se considera a veces como método "no supervisado", porque no utiliza ninguna fuente de conocimiento externo ni tampoco ningún corpus entero etiquetado a la manera de los algoritmos supervisados *standard*. Sin embargo, aplicando estrictamente el calificativo "no supervisado", el algoritmo de Yarowsky ha de considerarse "supervisado", ya que se autoarranca a partir de un número pequeño de ejemplos *etiquetados*, es decir, es un método (mínimamente) supervisado.

Los métodos no supervisados distribucionales identifican palabras que aparecen en contextos similares sin tener en cuenta ningún tipo de inventario de sentidos previo. Por ejemplo Schütze [Schütze 1998] utiliza dos pasos: primero discrimina posibles sentidos de la palabra objetivo dividiendo sus contextos en clusters o grupos que comparten características de la distribución del contexto similares; en la segunda etapa etiqueta cada cluster con una glosa que describe el significado de la palabra objetivo en los contextos de ese cluster.

Desde este punto de vista, estos métodos se pueden considerar como un intento de automatizar la labor que normalmente hacen los lexicografistas. En la segunda etapa, el lexicografista debe escribir una definición que describa un cluster compuesto de varios contextos en que aparece la palabra objetivo. Sin duda, esto requiere el uso de su sentido común (*real-world knowledge*) además del contenido de los contextos.

Una forma de resolver este importante problema sería, en lugar de intentar escribir una definición más o menos formal del significado del *cluster*, identificar un conjunto de palabras relacionadas con el contenido y significado del cluster. Por ejemplo, un conjunto de palabras para un cluster de contextos de la palabra objetivo *línea* podría estar formado por las palabras *teléfono, llamada* y *comunicando*. Aunque no sea tan sofisticada como una definición, sí que indica el significado del cluster y si se lograra automatizar se obtendría un método de DSP automático independiente del lenguaje y resolvería el problema de CBAC.

Sin embargo, ese proceso todavía no se ha logrado automatizar, y lo razonable para etiquetar los clusters producidos por un sistema no supervisado distribucional sería utilizar las fuentes de conocimiento (que normalmente se utilizan en los métodos basados en conocimiento). Como vimos en la sección 2.2.4 el método de McCarthy et

al. [McCarthy et al. 2004] podría utilizarse aquí para etiquetar (paso 2) los clusters previamente obtenidos. Se utilizan determinadas medidas [Jiang y Conrath 1997][Banerjee y Pedersen 2003] de la similitud semántica entre palabras y el sentido de la palabra objetivo juzgado más similar al conjunto de palabras que representan el dominio se considera su significado en ese dominio. Los experimentos de McCarthy et al. en Senseval-2 lograron una precisión algo por debajo de la línea de base del sentido más frecuente en la tarea de todas las palabras (*all words*), algo que ocurrió con otros dos sistemas en la misma prueba.

En cualquier caso, la utilización de una fuente de conocimiento externo (WordNet en este caso) hace que el sistema deje de ser no supervisado y esté afectado por el problema de CBAC, por lo menos en la portabilidad a un lenguaje extranjero.

2.4 El algoritmo de Yarowsky como método de DSP semisupervisado

El algoritmo de Yarowsky forma parte de la categoría de algoritmos semisupervisados, porque necesita una cantidad muy pequeña de entrenamiento (etiquetado previo), lo que los convierte en una solución muy interesante, ya que combinan la eficacia de los métodos supervisados, pero carecen prácticamente de su principal inconveniente que es la necesidad de etiquetar manualmente los corpus que se van a utilizar como entrenamiento. Este problema se trata en detalle en el capítulo 3.

El algoritmo de Yarowsky se puede considerar como un algoritmo de metaaprendizaje, en el que se puede intercalar *cualquier* algoritmo de Aprendizaje Automático AA (*Machine Learning*, ML) supervisado. En el algoritmo original este papel lo desempeña el clasificador conocido como Lista de Decisión (véase el capítulo 5). Esta propiedad, que es la razón por la cual se trata de un algoritmo semisupervisado, lo convierte en un método muy atractivo, por lo que le han seguido análisis, variaciones, optimizaciones y aplicaciones.

Blum y Mitchell [Blue y Mitchell 1998] proponen un algoritmo semi-supervisado llamado *co-training* (*coentrenamiento*) que genera iterativamente dos clasificadores, en vez de sólo uno, y utiliza cada uno de ellos para optimizar el otro.

Abney [Abney 2002] demostró que el coentrenamiento de Blum y Mitchell y el algoritmo de Yarowsky se basan en supuestos de independencia estadística distintos, llamados *view independence* y *precision independence* respectivamente, y que además son completamente diferentes, de forma que el algoritmo de Yarowsky no es un caso especial del coentrenamiento. Nigam y Ghani [Nigam y Ghani 2000] y Ng y Cardie [Ng y Cardie 2003] compararon los dos métodos en experimentos con modelos equivalentes y aportan resultados que favorecen al algoritmo de Yarowsky.

Abney [Abney 2004] analiza el algoritmo de Yarowsky matemáticamente y demuestra que algunas variantes del algoritmo propuestas por él mismo optimizan bien la *similitud* (*likelihood*) o bien una función objetivo muy parecida a la que denomina función *K*.
Eisner y Karakos [Eisner y Karakos 2005] demuestran que a veces es posible eliminar el último resquicio de supervisión en los métodos de autoarranque (*bootstrapping*). Para ello, se pueden probar muchas semillas candidatas, no supervisadas, y quedarse con aquella que produzca la salida más plausible. Los autores llaman a esta técnica *strapping* y aportan resultados mejores que el método estándar de seleccionar semillas (supervisadas) a mano propuesto por Yarowsky originalmente.

Traupman y Wilensky [Traupman y Wilensky 2003] describen tres intentos de mejorar la precisión del algoritmo de Yarowsky. En el primero utilizan la salida del clasificador producida en una iteración como entrada de entrenamiento para la siguiente. En el segundo preprocesan los corpus de entrenamiento y test con un Etiquetador de Categorías Gramaticales ECG (*Part of Speech tagger, POS tagger*) y utilizan las etiquetas obtenidas para filtrar sentidos posibles y así optimizar el poder predictivo de los contextos de la palabra objetivo. En el tercer experimento sustituyen la suposición habitual de que los sentidos de una palabra estén distribuidos uniformemente por la de que tienen una distribución más realista, que obtienen de las frecuencias de uso de los sentidos tal como aparecen en un diccionario. Los resultados de los experimentos dan al segundo como el más beneficioso, logrando sobrepasar la precisión normal del algoritmo; el tercer experimento aporta sólo una ligera mejora sobre el algoritmo estándar, y el primer experimento no produce resultados positivos, e incluso perjudica a la precisión normal.

Sarkar [Sarkar 2008] utiliza una versión modificada del algoritmo de Yarowsky original para construir un sistema de Traducción Automática Estadística TAE (*Statistical Machine Translation*, SMT). Compara su sistema con un sistema de TAE supervisado basado en frases de referencia, sobre un conjunto de prueba del corpus EuroParl como el utilizado en la *SMT shared task 2006*. El sistema supervisado de referencia se entrena con 25 000 pares de oraciones alineadas en inglés y francés. El francés funciona como idioma fuente y el inglés como idioma objetivo. Sarkar demuestra que un sistema semisupervisado que utiliza un conjunto adicional de 500 oraciones francesas no etiquetadas junto con 25 000 pares de oraciones alineadas de los dos idiomas produce una mejora en la marca *Bleu* casi equivalente a doblar el número de oraciones alineadas en el sistema supervisado de referencia de 25 000 a 50 000 pares. La marca *Bleu* es una medida de la precisión del sistema de TAE frente a un número de 4 a 10 traducciones manuales, es decir, hechas por traductores profesionales, para cada oración.

Otras aplicaciones de algoritmos análogos al de Yarowsky se utilizan en análisis sintáctico (*parsing*), aprendizaje de morfología (*morphology learning*), predicción del género gramatical (*gramatical gender prediction*), reconocimiento de entidades (*named entity recognition*) y deducción de léxico bilingüe (*bilingual lexicon induction*) [Smith 2006].

2.5 El algoritmo de Yarowsky como método de DSP supervisado

Los métodos supervisados utilizan una fuente de conocimiento "externa" de naturaleza distinta a la utilizada por los métodos basados en conocimiento (sección 2.2). En este caso el sistema utiliza un corpus (posiblemente distinto del corpus objetivo) etiquetado con los sentidos de la palabra objetivo. Este corpus etiquetado se utiliza como entrenamiento de un sistema de aprendizaje automático (*Machine Learning*, ML) [Màrquez et al. 2006].

Los sistemas supervisados obtienen generalmente mejores resultados que los no supervisados y que los basados en aprendizaje, como se ha puesto de manifiesto experimentalmente, por ejemplo, en las competiciones Senseval.

Sin embargo, como contrapartida estos métodos sufren el problema de CBAC (necesidad de producción y mantenimiento manual de corpus grandes etiquetados) lo que representa todavía un problema serio para ellos.

2.5.1 Aprendizaje automático para la clasificación

El problema de la DSP puede verse como un problema de clasificación: las ocurrencias de la palabra objetivo en el corpus se clasifican en varios grupos, cada uno de los cuales representa uno de los posibles sentidos de esa palabra.

Los problemas de clasificación se han estudiado extensamente en la disciplina conocida como Aprendizaje Automático (*Machine Learning* ML). A continuación se resume el marco conceptual en que se desarrollan estos problemas dentro del ML.

El objetivo de un método de aprendizaje supervisado para la clasificación es inducir a partir de un conjunto de entrenamiento S, una aproximación h de una función desconocida f que se aplica a un espacio de entrada X y cuya salida pertenece a un espacio desordenado $Y = \{1,\ldots, K\}$.

El conjunto de entrenamiento está formado por m ejemplos $S=\{(\mathbf{x}^1, y^1),\ldots, (\mathbf{x}^m, y^m)\}$, donde $\mathbf{x}$ pertenece a X y $y = f(\mathbf{x})$. El componente $\mathbf{x}$ de cada ejemplo suele ser un vector $\mathbf{x} = (x_1, \ldots, x_n)$ donde cada componente, llamado *rasgo* (*feature*) o atributo representa la información o las propiedades que describen el ejemplo, y puede tener valores discretos o reales. Los valores del espacio de salida Y asociados con cada ejemplo se llaman clases o categorías. Por tanto, cada ejemplo de entrenamiento queda completamente descrito por una serie de pares atributo-valor y una etiqueta de categoría.

Dado un conjunto de entrenamiento S el algoritmo de aprendizaje debe inducir un clasificador llamado h, que es una función aproximadora de f, de entre un conjunto de

posibles funciones *H*, también llamado espacio de hipótesis. Los algoritmos de aprendizaje varían dependiendo de tres aspectos: el espacio de hipótesis que utilizan (funciones reales, partición de dominios por hiperplanos paralelos al eje, funciones radiales, etc.), el lenguaje de representación elegido (árboles de decisión, conjuntos de probabilidades condicionales, redes neuronales, etc.) y el criterio de elección de la mejor hipótesis de entre varias compatibles con el conjunto de entrenamiento (simplicidad, margen máximo, etc.).

Dado un conjunto de vectores **x** nuevos, se utiliza *h* para predecir los correspondientes valores de *y*, esto es, clasificar los nuevos ejemplos de la forma posible más parecida a como lo haría *f*, es decir, produciendo el menor número posible de errores. Naturalmente, el número exacto de errores cometidos no se puede minimizar directamente, ya que se desconoce *f*. El procedimiento que se sigue es minimizar el número de errores en el conjunto de entrenamiento. Este principio inductivo se conoce como *minimización empírica del riesgo* y produce una estimación adecuada del número de errores reales, siempre que haya suficientes ejemplos de entrenamiento, es decir, *S* sea suficientemente grande.

Si el conjunto de entrenamiento *S* no es suficientemente grande se corre el riesgo de sobreestimar los datos de entrenamiento si se fuerza a cero el error de entrenamiento y, como consecuencia, la generalización (aproximación de *f*) no será buena. Este riesgo es mayor si hay ejemplos muy excepcionales en el entrenamiento, o si hay ruido (ejemplos de entrenamiento mal clasificados). El riesgo de sobrestimación también está relacionado con la complejidad de la hipótesis *h*. Esta complejidad se puede medir teniendo en cuenta la expresividad del conjunto de hipótesis *H* usando la dimensión VC de Vapnik-Chervonenkis [Vapnik 1998]. En la práctica se establece un compromiso entre complejidad de *h* y error cometido.

2.5.2 Principales metodologías de DSP supervisado

Métodos probabilísticos

Estos métodos estiman un conjunto de parámetros probabilísticos que expresan distribuciones probabilísticas condicionales o conjuntas entre categorías y contextos expresados mediante rasgos. Estos parámetros se usan para, dado un nuevo ejemplo, calcular la categoría que tiene la probabilidad condicional máxima de ser la correcta.
El algoritmo más simple de este tipo es el Naïve Bayes [Duda et al. 2001]. Utiliza la fórmula inversa de Bayes y supone la independencia condicional de los rasgos dada la categoría. Pese a su simplicidad se ha usado en muchas investigaciones sobre DSP [Gale et al. 1992][Leacock et al. 1993][Pedersen y Bruce 1997][Escudero et al. 2000b] y en muchos artículos [Mooney 1996][Ng 1997a][Leacock et al. 1998] en los que se afirma que produce resultados de precisión standard (al nivel del estado del arte).

El algoritmo Naïve Bayes tiene el problema de la suposición de independencia condicional. [Bruce y Wiebe 1994] proponen un modelo más complejo que considera ciertas características dependientes entre sí. Pero el modelo es demasiado complejo por el elevado número de parámetros a estimar y como consecuencia necesita también un gran número de ejemplos de entrenamiento. [Pedersen y Bruce 1997] proponen un método automático para reducir su complejidad.

El método de Entropía Máxima [Berger 1996] presenta una forma de combinar flexiblemente información estadística de muchos tipos. Ha sido aplicado a muchos problemas de PLN y parece prometedor en DSP [Suárez y Palomar 2002].

Métodos basados en la similitud de los ejemplos

Estos métodos utilizan una métrica de similitud de dos formas: bien comparando el vector de un nuevo ejemplo con los vectores prototipo (vectores agregados) aprendidos de cada sentido, o bien buscando los ejemplos aprendidos más próximos al nuevo ejemplo y eligiendo el sentido más frecuente entre ellos.

Normalmente se utiliza como métrica el coseno del ángulo que forman los vectores en un modelo de Espacio Vectorial (*Vector Space Model*, VSM). Este modelo permite aplicar un gran conjunto de rasgos de muchos tipos y ponderar los pesos de cada rasgo [Yarowsky 2001]. [Leacock 1993] comparó favorablemente el modelo VSM con el método Naïve Bayes en DSP.

El algoritmo de aplicación de la métrica del modelo VSM más extendido se llama algoritmo de los k vecinos más próximos (k-Nearest Neighbor, k-NN). Consiste en calcular los k vectores ejemplo etiquetados más próximos al vector nuevo que se quiere clasificar, y aplicar una "media" de sus sentidos para obtener la predicción. Según [Ng 1997a] éste es el algoritmo de aprendizaje óptimo para DSP. Para [Daelemans et al. 1999] los métodos basados en la similitud de los ejemplos en general, son los mejores en PLN porque no aplican ningún tipo de generalización a los datos y, como consecuencia, no dejan de tener en cuenta las excepciones. El algoritmo k-NN también se puede utilizar para integrar varias fuentes de conocimiento diferentes [Stevenson y Wilks 2001].

Métodos basados en reglas de decisión

Estos métodos utilizan reglas de decisión asociadas con cada sentido de la palabra objetivo. Así, dada una instancia de la palabra objetivo (ejemplo o contexto), el sistema selecciona una o varias reglas satisfechas por los rasgos del ejemplo nuevo y le asigna un sentido conforme a estas reglas.

El método conocido como *lista de decisión* (*decision list*, *DL*) es el que utiliza el algoritmo de Yarowsky, y consiste en una lista ordenada de reglas de la forma

(*condición*, *clase*, *peso*). El peso representa el grado de asociación entre la condición y la clase (sentido), y se calcula mediante una función basada en el corpus de entrenamiento. Para clasificar un ejemplo nuevo se recorre la lista ordenada (en orden decreciente de pesos), y se elige la clase (sentido) correspondiente a la primera regla (máximo peso) que satisface la condición.

En el algoritmo de Yarowsky (capítulo 5) la condición de la lista de decisión corresponde a un rasgo (basado en las colocaciones de las palabras del contexto de la palabra objetivo) y los pesos se calculan mediante una medida (*log-likelihood measure*) de la plausibilidad de que dado el valor del rasgo, el ejemplo nuevo se refiera al sentido 'clase' de la regla.

Otro método basado en reglas de decisión es el *árbol de decisión* (*decision tree*, *DT*). Un árbol de decisión es un árbol n-ario que clasifica recursivamente el conjunto de entrenamiento. Los nodos internos representan un conjunto de rasgos básicos, las ramas representan reglas y las hojas representan sentidos. Este método ha sido utilizado ampliamente en ML en general, pero poco en DSP. En [Mooney 1996] se utilizó comparativamente con otros algoritmos de ML aplicados a DSP y los resultados fueron desfavorables para esta estructura debido a una serie de razones.

Métodos basados en la combinación de reglas

Estos métodos se basan en la combinación de un conjunto de reglas *homogéneas* que aprende y combina un *único* algoritmo de aprendizaje. El más conocido es el algoritmo AdaBoost.

Este algoritmo de aprendizaje se basa en combinar linealmente muchas reglas simples y no necesariamente muy precisas llamadas *reglas débiles* para dar lugar a un sistema clasificador robusto con tasa de error en el conjunto de entrenamiento arbitrariamente baja. Las reglas débiles se entrenan secuencialmente y se mantiene una distribución de pesos ponderados de forma que las reglas se concentren en los ejemplos que fueron más difíciles de clasificar por el conjunto de reglas precedentes. El método se ha empleado en varios problemas de PLN [Schapire 2003] y está especialmente indicado para cuando el algoritmo débil es un algoritmo inestable, como los árboles de decisión. Además varios experimentos [Escudero et al. 00a, 00c, 2001] indican que este método puede superar a muchos otros en DSP, entre otros los Naïve Bayes, similitud de ejemplos y listas de decisión.

Clasificadores lineales y métodos basados en núcleos

Un clasificador lineal es un hiperplano en un espacio de rasgos de dimensión n que puede representarse mediante un vector de pesos $\mathbf{w}$ y una distancia al origen b. Los pesos del vector representan la importancia del rasgo correspondiente en la regla de

clasificación. Existen muchos algoritmos para entrenar estos clasificadores: Perceptron, Widrow-Hoff, Winnow, Exponentiated-Gradient, Sleeping experts, etc.

Los clasificadores lineales se han usado ampliamente en Recuperación de la Información (RI) debido a su utilidad como clasificadores de texto. Sin embargo, hasta 2000 su uso en DSP fue más bien bajo, y además dos experimentos [Mooney 1996] [Escudero et al. 2000c] produjeron resultados muy desalentadores. A partir de esa fecha, la aplicación de métodos basados en *núcleos* (sobre todo Support Vector Machines, SVM [Boser et al. 1992]) ha dado buenos resultados en DSP ([Murata et al. 2001][Lee y Ng 2002][Strapparava 2004][Lee et al. 2004][Agirre y Martinez 2004a][Cabezas et al. 2004][Escudero et al. 2004], los cuatro últimos en Senseval-3). También han dado buenos resultados en DSP otros métodos basados en núcleos [Carpuat et al. 2004][Wu et al. 2004][Popescu 2004][Ciaramita y Johnson 2004].

Un método basado en núcleos es un algoritmo de aprendizaje no lineal que utiliza *funciones de núcleo* para no tener que hacer aplicaciones explícitas no lineales de los valores de los rasgos de entrada. Esto se puede hacer si los vectores que representan los ejemplos aparecen solamente en productos escalares tanto en el aprendizaje como en la regla de clasificación. Esto permite el aprendizaje de funciones no lineales, representadas por una aplicación no lineal de los rasgos de entrada en un espacio de rasgos multidimensional, en el que rasgos nuevos se pueden expresar como combinaciones de rasgos básicos, utilizando aprendizaje lineal standard. Los métodos basados en núcleos incrementan la expresividad de los clasificadores lineales al aprender funciones no lineales que describen las singularidades de los datos de cada aplicación y el conocimiento de fondo, pero de un modo flexible y eficiente.

2.6 DSP y dominio

El dominio[1] en que ocurre una palabra ambigua es un aspecto muy importante en la tarea de DSP. El dominio se puede considerar como una instanciación de una noción más general conocida como *espacio semántico* (*semantic space*) y que se podría definir como "una representación matemática de un cuerpo grande de texto" [Landauer et al. 1998]. Así, el espacio semántico puede ser especificado por un dominio (por ejemplo biomedicina), un subdominio (anatomía), una tarea concreta (transplante de corazón) o una organización (biomedicina en Aventis) [Buitelaar et al. 2006].

En el capítulo 4 se indica que una de las formas para determinar los posibles sentidos de una palabra ambigua era su comportamiento durante la traducción de la ocurrencia de la palabra objetivo a otro idioma en un corpus paralelo. También se indicó que otra forma

[1] La palabra "dominio" ("domain", [Peh and Ng 1997][Cucchiarelli and Velardi 1998]) se refiere aquí a "un trozo de texto semánticamente coherente que trata sobre un tema concreto". Muchas investigaciones se refieren a lo mismo con otros términos: "subject" [Guthrie et al. 1991], "discourse" [Gale et al. 1992], "topic" [Agirre et al.2001].

de hacer lo mismo era identificar en qué dominio se producía la aparición de la palabra en el corpus: si una palabra dada ocurre en un dominio determinado, podemos saber que tiene en ese caso cierto sentido, y puede tener otro muy diferente si la palabra aparece en otro dominio diferente.

Este aspecto está muy relacionado con el hecho de que parece prácticamente imposible definir un corpus completamente genérico, que represente cualquier dominio. También podríamos expresarlo diciendo que se podría reducir toda la DSP de una palabra objetivo a determinar en qué dominio concreto aparece. Esto sería aún más cierto si la subtarea de DSP de esa palabra objetivo estuviera orientada a una tarea de Recuperación de la Información (IR).

Todo esto quiere decir que la caracterización o modelización del dominio podría ser una de las primeras prioridades para la DSP.

2.6.1 Enfoques para la caracterización del dominio

Códigos de dominio (*subject codes*)

Un espacio semántico o dominio puede estar indicado en un diccionario mediante los llamados códigos de dominio o, en inglés, subject codes. Por ejemplo, en el diccionario LDOCE (*Longman Dictionary of Contemporary English*) [Procter 1978], hay códigos de dominio como MD ('Medical Domain') o ML ('Meteorology'). Naturalmente, estos códigos pueden indicar qué sentidos de una palabra se usan en qué dominios.

Los códigos de dominio se pueden usar para detectar el dominio de un texto con sólo contar su frecuencia entre todas las palabras no vacías [Walter y Amsler 1986]. Nótese que esta técnica está sujeta al problema del CBAC al menos al nivel de utilización de una base de datos léxica externa.

Los códigos de dominio también se pueden usar para construir modelos de contexto de un dominio [Guthrie et al. 1991], teniendo en cuenta todas las palabras incluidas en las definiciones y oraciones de muestra de todas las palabras de un diccionario que tengan en común ese mismo código de dominio.

Por todo esto, puede resultar interesante identificar el espacio semántico o dominio de los *synsets*[2] de WordNet más explícitamente utilizando códigos de dominio. Esto se ha hecho en WordNet Domains [Magnini y Cavaglià 2000][3], donde los *synsets* de WordNet se han anotado manualmente con al menos una etiqueta de dominio,

[2] Los *synsets* (synonym sets) son grupos de palabras con significados similares en WordNet.
[3] http://wndomains.itc.it

utilizando un conjunto de unas doscientas de estas etiquetas organizadas jerárquicamente.

En la Tabla 2.1 se muestra la distribución de dominio de los synsets de WordNet 1.6 considerando 43 etiquetas de dominio disjuntas correspondientes a un nivel intermedio de la jerarquía (por ejemplo se usa Sport y no Volleyball ni Basketball).

Número synsets	**Dominio**	**Número synsets**	**Dominio**	**Número synsets**	**Dominio**
36820	Factotum	1771	Linguistics	532	Publishing
21281	Biology	1491	Military	511	Tourism
4637	Herat	1340	Law	509	Computer-science
3405	Psychology	1264	History	493	Telecommunication
3394	Architecture	1103	Industry	477	Astronomy
3271	Medicine	1033	Politics	381	Philosophy
3039	Economy	1009	Play	334	Agriculture
2998	Alimentation	963	Anthropology	272	Sexuality
2975	Administration	937	Fashion	185	Body-care
2472	Chemistry	861	Mathematics	149	Artisanship
2443	Transport	822	Literature	141	Archaeology
2365	Art	746	Engineering	92	Veterinary
2225	Physics	679	Sociology	90	Astrology
2105	Sport	637	Commerce		
2055	Religión	612	Pedagogy		

Tabla 2.1. Distribución de los *synsets* de WordNet 1.6 entre etiquetas de dominios disjuntos de WordNet Domains en un nivel intermedio de la jerarquía. Los *synsets* de WordNet son conjuntos de palabras sinónimas.

La metodología que se utilizó para anotar WordNet Domains fue manual, siguiendo criterios léxico-semánticos que a su vez se ayudaron de las relaciones conceptuales ya existentes previamente en WordNet.

La información aportada por los dominios es complementaria a la ya existente en WordNet:

- Los dominios pueden incluir synsets de diferentes categorías sintácticas: por ejemplo, el dominio Medicine incluye sentidos de sustantivos como *doctor-1* y *hospital-1* o de verbos como *operate-7*.
- Pueden incluir sentidos de diferentes subjerarquías de WordNet. Por ejemplo, Sport contiene sentidos como *athlete-1*, derivado de *life-form-1*, *game-equipement-1* derivado de *physical-object-1*, *sport-1* derivado de *act-2* y *playing-field-1* derivado de *location-1*.
- Los dominios pueden agrupar varios sentidos de una palabra en uno sólo correspondiente a un dominio; es decir, colapsar varios sentidos en uno más general que los subsuma, reduciendo la granularidad de WordNet. Esto está evidentemente relacionado con el nivel de granularidad necesario para las aplicaciones de PLN, y confirma la teoría de que podría ser suficiente

determinar el dominio para desambiguar las palabras, al menos en aplicaciones como IR (Recuperación de la Información). Como ejemplo, en la Tabla 2.2 están los diez sentidos de la palabra *bank* tal como aparecen en WordNet, y se ve que colapsando sentidos mediante etiquetas de dominio, el número de sentidos se puede colapsar a 3 ó 4. Este número de sentidos sigue siendo muy alto, pero hay que tener en cuenta que la frecuencia de algunos de ellos es despreciable.

Sentido	**Synset y glosa**	**Dominios**
1	*depository, financial institution, bank, banking concern, banking company* (a financial institution)	Economy
2	*bank* (sloping land)	Geography, Geology
3	*bank* (a suply or stock held in reserve)	Economy
4	*bank, bank building* (a building)	Architecture, Economy
5	*bank* (an arrangement of similar objects)	Factotum
6	*savings bank, coin bank, money box, bank* (a container)	Economy
7	*bank* (a long ridge or pile)	Geography, Geology
8	*bank* (the funds held by a gambling house)	Economy, Play
9	*bank, cant, camber* (a slope in the turn of a road)	Architecture
10	*bank* (a flight maneuver)	Transport

Tabla 2.2. Sentidos de la palabra *bank* y sus *synsets* en WordNet y los dominios correspondientes en WordNet Domains.

Otra base de datos léxica muy conocida y utilizada que se puede interpretar bajo la perspectiva de los códigos de dominio es el *Roget's International Thesaurus*. Es un tesauro creado en el siglo XIX para el idioma inglés en el que todas las palabras están clasificadas en 'categorías' que reflejan el concepto representado por la palabra. Hay unas 1042 categorías, y se pueden interpretar como códigos de dominio.

Bajo esta perspectiva, se puede considerar el algoritmo de Yarowsky de 1992 [Yarowsky 1992] como un algoritmo de DSP basado en códigos de dominio. Este algoritmo se basa en entrenar un clasificador mediante las categorías de Roget's de las palabras de su contexto.

El método colecciona palabras que aparecen normalmente en el contexto de cada categoría Roget's. Para cada palabra objetivo, entrena un clasificador para separar las dos o más categorías a las que pertenece, basándose en las palabras del contexto de cada categoría. Dado un contexto nuevo, se le aplica el clasificador.

El algoritmo de Yarowsky de 1992 logra un 92% de precisión en ambigüedades de nivel homográfico, lo cual, sin llegar a la precisión del algoritmo de 1995, es un resultado bastante importante. Sin embargo, la principal desventaja frente a éste es que sufre el problema de CBAC, al utilizar una base de datos léxica externa.

[Stevenson y Wilks 2001] adaptaron el algoritmo de Yarowsky de 1992 utilizando los códigos de dominio del LDOCE en vez de las categorías del Roget´s. Además en lugar de suponer una distribución uniforme de las categorías como hace ese algoritmo,

estimaron la probabilidad a priori de cada código de dominio como la proporción de sentidos en el LDOCE a la cual se asigna un determinado código. Compararon la precisión de este método con otro que en lugar de códigos de dominio utilizaba palabras de definiciones de diccionario y con otro que usaba preferencias selectivas (ver sección 2.1) y los resultados fueron de 79%, 65% y 44% respectivamente, lo cual indica la utilidad de este tipo de información de dominio.

En [Escudero et al. 2000] se probó un sistema supervisado añadiéndole rasgos de dominio obtenidos de WordNet Domains, y utilizando probabilidades a priori para cada dominio (la frecuencia del dominio en WordNet Domains). El sistema se probó en Senseval-2 y obtuvo una mejora sistemática del 3% en sustantivos sobre el algoritmo sin los rasgos de dominio.

En [Magnini et al. 2002] se presenta un sistema que utiliza exclusivamente información de WordNet Domains denominado Domain Driven Disambiguation (DDD). La idea básica de este sistema es que la desambiguación de una palabra w en un contexto t puede ser un proceso de comparación entre el dominio del contexto y los dominios de los sentidos de la palabra.

El algoritmo consta de tres pasos. En el paso (1) se calcula el llamado vector de dominio (*domain vector*, DV) del contexto t de la palabra w. El DV es un vector en un espacio multidimensional en el que cada dominio representa una dimensión del espacio. El valor de cada componente es la relevancia del dominio correspondiente con respecto al objeto descrito por el vector. En el paso (1) se utiliza un contexto de +/- 50 palabras alrededor de la palabra w. En el paso (2) se construye un DV para cada sentido posible de la palabra objetivo. Si se usa una versión no supervisada del algoritmo, se utilizan las asociaciones entre sentidos y dominios que hay en WordNet Domains. Si la versión del algoritmo es supervisada, se obtienen los DV aplicando el paso (1) a cada ejemplo de entrenamiento. En el paso (3) se elige el sentido de w cuyo DV maximiza la similitud con los DV del contexto a desambiguar, utilizando como medida el producto escalar de DVs.

El sistema DDD se probó en la tarea de todas las palabras en Senseval-2, y obtuvo una precisión de 75% y un *recall* de 36%. El resultado bajo de *recall* se debe al hecho de que sólo un subconjunto de los sentidos de palabras de un documento están relacionados realmente con el dominio del contexto. Esta observación es importante, y otorga al DDD el valor de poder predecir la clase de palabras que se pueden desambiguar por un sistema que utilice este tipo de fuente de conocimiento.

Firmas de dominio (topic signatures) y variación de dominio (topic variation)

Los modelos de contexto de dominio de [Guthrie et al. 1991] mencionados en la sección anterior pueden interpretarse como *firmas de dominio* (*topic signatures*) del dominio o tópico en cuestión. De hecho una firma de dominio se puede construir incluso sin

necesidad de usar códigos de dominio procedentes de un diccionario o de una base de datos léxica externa: se puede generar (semi)-automáticamente a partir de un recurso léxico y después validarla en un corpus de un dominio concreto [Hearst y Schütze 1993].

Una idea semejante a éstas consiste en construir dominios sobre sentidos de palabras coleccionando documentos que correspondan a ese sentido. Este conjunto de documentos representaría el dominio, y a partir de él se podría extraer la firma de dominio correspondiente al sentido de la palabra objetivo.

Por ejemplo se pueden formular consultas sobre sentidos de palabras o synsets de WordNet a buscadores de la Web [Agirre et al. 2000] [Agirre et al. 2001]. Estas consultas o querys están formadas por combinaciones booleanas de palabras clave extraídas del synset o de la glosa de WordNet, hiperónimos, hipónimos, etc. Los documentos devueltos por el buscador constituyen un conjunto de documentos correspondientes al dominio del sentido o del synset de la palabra, y a partir de él se elabora una lista de las palabras más relevantes del sentido, que sería la firma de dominio.

Como ejemplo, la Tabla 2.3 muestra las diez primeras palabras más relevantes de tres sentidos de la palabra *boy* usando el sistema interfaz web de WordNet 1.6 de firma de dominio [Agirre y Lopez de Lacalle 2004]. Cada una de estas listas correspondería en cierta medida a la firma de cada sentido de la palabra en diferentes dominios.

Sentido 1 (*male child, boy, child*)		**Sentido 2 (‘*informal referente to a man*’)**		**Sentido 3 (*son, boy*)**	
Puntuación	**Palabra**	**Puntuación**	**Palabra**	**Puntuación**	**Palabra**
43.14	*male*	35.00	*exboyfriend*	72.14	*mamma’s*
33.47	*sonny*	31.89	*womaniser*	70.76	*esau*
29.41	*ball*	24.23	*womanizer*	69.65	*man-child*
27.85	*laddie*	23.71	*ex-husband*	54.17	*mama’s*
25.57	*schoolboy*	23.30	*eunuch*	50.59	*offspring*
21.26	*sirrah*	19.98	*galoot*	49.56	*male*
18.91	*ploughboy*	18.84	*divorced*	29.85	*jnr*
16.35	*adult*	18.02	*philanderer*	27.76	*mother’s*
14.23	*altar*	16.88	*strapper*	12.73	*female*
13.61	*bat*	13.98	*geezer*	6.35	*chromosome*
29.41	*ball*	24.23	*womanizer*	69.65	*man-child*

Tabla 2.3. En orden de revancia se muestran las diez primeras palabras para tres sentidos de la palabra *boy* en WordNet 1.6. Las palabras se extraen del *synset* correspondiente al sentido y la puntuación indica la relevancia de la palabra para el dominio del sentido, según los documentos devueltos por una consulta que incluya esa palabra lanzada por un buscador en la Web.

El concepto de firma de dominio está relacionado con la idea de la definición de qué es exactamente un espacio semántico o dominio. La propia idea de dominio supone la existencia de un sentido dominante, dado un dominio. Esto sólo es estrictamente cierto en el caso de la polisemia fuerte o de nivel homográfico (ver capítulo 4): por ejemplo la

palabra *sentencia* tendrá un sentido muy claro en un dominio de lenguaje judicial y otro muy distinto y claro en un dominio de lingüística. Sin embargo, en el caso de la polisemia débil (en inglés *aspect polisemy*) las firmas de dominio se solaparán (esto se puede ver claramente en los sentidos 1 y 3 de la Tabla 2.3.

Esta distinción entre polisemia fuerte y débil se ha puesto claramente de manifiesto en la prueba de la hipótesis *un sentido por discurso* de [Gale 1992] (véase sección 2.2.4) que trabaja con polisemia fuerte y en su extensión a la polisemia débil en [Krovetz 1998]. En este último trabajo se prueba que la hipótesis es mucho menos cierta para más de dos sentidos, es decir, para polisemia débil como la que se usa en WordNet.

Una forma de definir un espacio semántico se puede derivar de los resultados de algoritmos desambiguadores cuando hay variaciones de dominio, como se discute ampliamente en los capítulos 6 y 8. Los resultados expuestos en esos capítulos muestran que se puede definir un espacio semántico o dominio como un trozo más o menos grande de texto (un corpus, un texto, un trozo de texto, etc.) con una distribución homogénea de sentidos y de las colocaciones correspondientes.

Sintonización de dominio (domain tuning)

Como se pudo comprobar en la sección anterior, los sentidos que WordNet asigna a muchas palabras, como por ejemplo la palabra *boy* de la Tabla 2.3, no encajan con los dominios derivados empíricamente para esa palabra. Este fallo refleja un problema importante y bastante habitual, que es el de estar utilizando un inventario de sentidos general (es decir, no adaptado a un dominio) como es en este caso WordNet.

Para resolver este problema es necesario sintonizar el inventario de sentidos general al dominio concreto con que se esté trabajando. Esto implica elegir sólo aquellos sentidos que sean más apropiados al dominio [Basili et al. 1997][Cucchiarelli y Velardi 1998][Turcato et al. 2000][Buitelaar y Sacaleanu 2001], así como ampliar el inventario de sentidos con términos y sentidos nuevos, específicos del dominio [Buitelaar y Sacaleanu 2002][Vossen 2001].

En las dos próximas secciones se van a revisar dos enfoques para la sintonización de un inventario de sentidos general (WordNet) a dominios concretos. En el primer caso se definen empíricamente los *sysnsets* de nivel alto de WordNet más apropiados, y se eliminan los sentidos de nivel más bajo. En el segundo caso se define el conjunto de synsets más apropiado en el nivel más bajo de la jerarquía. Por ello, nos referimos a ellos como sintonización arriba-abajo (*top-down*) y abajo-arriba (*bottom-up*) respectivamente.

Sintonización de dominios arriba-abajo

Este método fue propuesto en [Cucchiarelli y Velardi 1998] y sostiene que si un inventario de sentidos adaptado a un dominio es equilibrado, esto es, tiene una distribución equilibrada de palabras a sentidos, y tiene un nivel adecuado de abstracción, es decir compromiso entre ambigüedad y generalidad, puede derivarse automáticamente aplicando los siguientes cuatro criterios: generalidad, poder discriminatorio, cobertura de dominio y ambigüedad media.

Categoría	Synset de nivel alto
C_1	*person, individual, someone, mortal, human, soul*
C_2	*instrumentality, instrumentation*
C_3	*written communication, written language*
C_4	*message, content, sublect matter, substance*
C_5	*measure, quantity, amount, quantum*
C_6	*Action*
C_7	*Activity*
C_8	*group action*
C_9	*Organization*
C_{10}	*psychological feature*
C_{11}	*Posesión*
C_{12}	*State*
C_{13}	*Location*

Tabla 2.4. Categorías del dominio financiero basadas en el corpus del WSJ. Cada categoría se corresponde con un grupo de *synsets* de WordNet.

El método empieza aplicando el algoritmo de Hearst y Schütze de 1993 [Hearst y Schütze 1993] para determinar empíricamente un conjunto equilibrado de synsets de WordNet para el dominio específico, donde cada conjunto de *synsets* representa una categoría o sentido. Este algoritmo utiliza una función recursiva para agrupar *synsets* en categorías siguiendo la estructura jerárquica de WordNet. El algoritmo dispone de límites inferiores y superiores de los tamaños de las categorías que se van formando.

Una vez que las categorías se han formado se aplica una función de medida para decidir qué conjuntos de categorías son más relevantes para el dominio de trabajo. Se aplican medidas para los cuatro criterios mencionados antes. Por ejemplo, la generalidad de un *synset* de nivel alto C_i se puede expresar como $1/DM(C_i)$, donde $DM(C_i)$ representa la distancia media entre el conjunto C_i y los *synsets* del nivel más alto.

La Tabla 2.4 ilustra el conjunto de categorías seleccionadas para el dominio financiero del corpus Wall Street Journal (WSJ). Dado este conjunto de synsets de alto nivel, sólo

aquellos sentidos que son subsumidos por ellos se mantienen como sentidos del inventario sintonizado para este dominio. Por ejemplo, la Tabla 2.5 muestra cómo sólo 5 de 16 sentidos se mantienen para la palabra objetivo *stock*. La Tabla 2.6 muestra cómo se eliminan los otros sentidos.

Sentido	Jerarquía de synsets para el sentido	Synset del nivel alto para el sentido
1	*capital* > *asset*	*posession* (C_{11})
2	*support* > *device*	*instrumentality* (C_2)
4	*document* > *writing*	*written communication* (C_3)
5	*accumulation* > *asset*	*possession* (C_{11})
6	*ancestor* > *relative*	*person* (C_1)

Tabla 2.5. Sentidos de la palabra objetivo *stock* obtenidos por sintonización de dominio del corpus WSJ. De las 13 categorías para el dominio financiero de la Tabla 2.4, sólo se mantienen las 5 que subsumen alguno de los sentidos de *stock*.

Sentido	Jerarquía de synsets para el sentido
3	*stock, inventory* > *merchandise, wares* > ...
7	*broth, stock* > *soup* > ...
8	*stock, caudex* > *stalk, stem* > ...
9	*stock* > *plant part* > ...
10	*stock, gillyflower* > *flower* > ...
11	*malcolm stock, stock* > *flower* ...
12	*lineage, line of descent* > ... > *genealogy* > ...
14	*lumber, timber* > ...

Tabla 2.6. Sentidos de la palabra objetivo stock descartados por sintonización de dominio del corpus WSJ. Estos 8 sentidos no se ven subsumidos por ninguna de las categorías para el dominio financiero de la Tabla 2.4.

Sintonización de dominios abajo-arriba

En este caso, en lugar de propagar la propiedad de un *synset* de alto nivel respecto a un dominio hacia abajo en la jerarquía de WordNet, se asigna la propiedad respecto a un dominio a los sentidos de las palabras directamente, es decir, desde abajo hacia arriba.

[Buitelaar y Sacaleanu 2001] utilizan un método que determina la propiedad respecto a un dominio de los *synsets* basándose en la propiedad respecto al dominio de los sinónimos que componen el *synset* a través de su ocurrencia conjunta en un corpus específico de ese dominio.

En la Figura 2.1 se puede ver una representación algorítmica de este sistema de DSP basado en sintonización de dominios de abajo hacia arriba.

(1) para cada palabra W_1 en WordNet *WN*
(2) para cada *synset S* al que pertenece W_1
(3) para cada sinónimo W_2 del *synset S*, calcular su propiedad o relevancia respecto al corpus específico al dominio *DC*
(4) calcular la propiedad respecto al dominio de *S* contando todos los W_2
(5) asignar el *S* con mayor propiedad respecto al dominio a cada aparición de W_1 en el DC

Figura 2.1. El método de sintonización de dominios abajo-arriba de [Buitelaar y Sacaleanu 2001] en forma algorítmica.

[Buitelaar y Sacaleanu 2001] utilizan una medida de la propiedad o relevancia de un término respecto a un dominio basada en la medida *tf.idf* que se utiliza en los modelos vectoriales de Recuperación de la Información (IR) [Salton y Buckley 1988]. En concreto, dado un término *t*, un corpus específico de un dominio *d* y teniendo en cuenta *N* corpus específicos de dominio diferentes, la expresión de la relevancia de *t* respecto a *d* viene dada por:

$$relevancia(t \mid d) = \log(tf_{t,d}) \log \mid N / df_t \mid$$

donde *tf* y *df* son la frecuencia del término y la frecuencia del corpus, respectivamente. Nótese que en vez de contar la frecuencia entre documentos como en IR, aquí se cuenta la frecuencia entre corpus específicos de dominios.

Una vez calculada la relevancia del término, se calcula la relevancia de todo el *synset* sin más que sumar la relevancia de cada término de éste.

Para medir la corrección del método [Buitelaar y Sacaleanu 2001] comprueban dos cosas: 1) cómo de bien se seleccionan conceptos específicos de un dominio y 2) cómo de bien se selecciona el mejor *synset* (o sentido) correspondiente de los términos específicos del dominio. En cuanto al primer punto obtienen una precisión de entre el 80% y el 90% en experimentos con tres dominios: financiero, fútbol y médico. Con respecto al segundo punto obtienen diferentes resultados para cada dominio: en el dominio médico, de 24 términos 12 tenían al menos un sentido específico de ese dominio, de los cuales todos fueron determinados correctamente; para el dominio financiero fueron 5 correctos de 6 con algún sentido específico de ese dominio de entre un total de 17 términos; y para el dominio de fútbol los resultados fueron 5 correctos de 6 de entre 8 términos en total. Por tanto, los resultados indican que este método produce una selección correcta de sentidos específicos de dominio con bastante precisión.

Un método similar propuesto por [McCarthy et al. 2004], revisado en la sección 2.2.4), produjo una precisión del 64% en granularidad de sentidos a nivel de WordNet en la

tarea de todas las palabras de Senseval-2, lo cual es un muy buen resultado en relación con el resto de métodos no supervisados de esa competición.

Nótese que aunque ambos métodos son no supervisados en el sentido de no necesitar corpus etiquetados, sí que utilizan fuentes de conocimiento externas, es decir, bases de datos léxicas como WordNet, con lo que esto implica en relación al problema del CBAC (*Knowledge Acquisition Bottleneck*).

2.7 La evaluación de los sistemas de DSP

En esta sección se va a revisar el estado del arte en la evaluación de sistemas de DSP automáticos, basándonos fundamentalmente en el artículo de 2006 de Palmer, Ng y Dang [Palmer et al. 2006], poniendo el énfasis en su relación con el problema de la granularidad de sentidos y del inventario de sentidos en general, como se hace en otras partes de este trabajo.

Como se verá en el capítulo 4, probablemente la decisión más importante a la hora de diseñar un ejercicio de evaluación de sistemas de DSP sea la elección del inventario de sentidos. El inventario de sentidos en este contexto suele ser un léxico computacional o un diccionario procesable automáticamente que divide cada palabra objetivo del corpus objetivo en una serie 'digital' de sentidos. En otros contextos, como los sistemas no supervisados (sección 2.3), el inventario de sentidos puede juzgarse contraproducente *per se* y llegar a ser inexistente a priori.

Una característica muy fuerte de los inventarios de sentidos, y que en cierto modo pasa desapercibida a primera vista, es su gran diversidad: las entradas para palabras polisémicas (sobre todo polisemia débil o *aspect polisemy*) pueden ser muy diferentes entre unas bases de datos y otras, y dependen mucho entre otros factores del dominio del corpus objetivo o de su tamaño y cobertura. Este hecho está muy relacionado con la falta o imposibilidad de la existencia de un 'diccionario universal' de sentidos, que fuera soporte de un 'inventario universal' o simplemente de un 'corpus universal'. Nótese además la relación de este hecho con los problemas derivados del CBAC.

Dentro de la consideración de un inventario de sentidos *a priori* y sin pretensiones de ser 'universal' tal como se diseñaría en una aplicación real o bien en el diseño de un ejercicio práctico de evaluación de sistemas automáticos de DSP, uno de los criterios de diseño más importantes para la adecuación del inventario a la aplicación práctica o para la calidad del etiquetado de sentidos en el corpus de entrenamiento o de test en el ejercicio de evaluación, es la elección del número de sentidos o granularidad de sus distinciones. Este tema se trata en esta sección en su relación con la evaluación de sistemas automáticos de DSP. En el capítulo 4 se discute desde el punto de vista de la DSP en general.

2.7.1 Definición de tareas de evaluación

Se distinguen dos tipos de evaluación de sistemas de DSP [Ide y Véronis 1998]. En la evaluación *in vitro*, se define una tarea de DSP independientemente de cualquier aplicación de PLN real. Los sistemas se evalúan frente a medidas de eficacia (*benchmarks*) construidas especialmente para esa tarea. Por el contrario, en la evaluación *in vivo* se mide la eficacia de un componente de DSP en términos de su contribución a la eficacia total de una aplicación de PLN real y concreta.

Aunque a primera vista parezca más realista el segundo tipo de evaluación, ya que al final lo que se pretende es mejorar el rendimiento de aplicaciones de PLN reales, de hecho casi nunca se han intentado evaluaciones de ese tipo. Por el contrario casi siempre se ha considerado la DSP como una tarea de clasificación aislada, como pueda ser la asignación de categorías gramaticales (*part of speech tagging*). La razón fundamental es la mayor facilidad de definición e implementación del sistema de evaluación. Evidentemente es mucho más problemático adaptarse a diferentes aplicaciones reales, y además éstas no se pueden comparar directamente entre sí desde el punto de la eficacia (de su sistema de DSP, sea éste explícito o no).

Dentro de los sistemas de evaluación aislados, se puede utilizar un marco de *tarea de muestra léxica* (*lexical sample task*) o de *tarea de todas las palabras* (*all words task*). En la tarea de todas las palabras los sistemas deben etiquetar todas las palabras (no vacías) en un corpus de texto o discurso. Esta tarea es diferente a la tarea de etiquetar con categorías gramaticales (*part of speech tagging*) en un aspecto muy importante: en la tarea de POS *tagging* el conjunto de clases (sustantivo, adjetivo, verbo, etc.) es el mismo para todas las palabras, mientras que en la tarea de todas las palabras cada palabra (o al menos cada lema) tiene un conjunto de clases o etiquetas distinto.

El hecho de que se deban etiquetar todas las palabras restringe fuertemente la elección del inventario de sentidos, debido a que es necesario usar un diccionario muy general y de acceso público. En cambio, en la tarea de muestra léxica, se selecciona del léxico cuidadosamente una muestra de palabras y de sus apariciones en un corpus. Los sistemas deben etiquetar sólo estas muestras y a partir de sus apariciones en trozos cortos de texto. Por lo tanto, sólo se necesitan las entradas en un diccionario de las palabras muestra, con lo que la elección del diccionario es mucho más flexible. También es evidente que esta tarea es mucho más adaptable a un dominio determinado.

El tipo de tarea (muestra léxica o todas las palabras) tiene influencia en la metodología de etiquetado manual de los sentidos de las palabras, tanto para entrenamiento como para test. Cuando se puede etiquetar todas las apariciones u ocurrencias de cada palabra en el corpus a la vez, el proceso de etiquetado es mucho más rápido y fiable que cuando se etiquetan todas las palabras consecutivas del texto. El primer método se llama etiquetado objetivado (*targeted tagging*) y encaja mejor en la tarea de muestra léxica

que en la de todas las palabras. Esto hace que existan más sistemas de evaluación para la primera tarea que para la segunda. Además, un sistema diseñado para competir en la tarea de todas las palabras también puede evaluarse en la tarea de muestra léxica, mientras que lo mismo no ocurre a la inversa. En cambio, la tarea de todas las palabras presenta la ventaja de aproximarse mejor a la evaluación de la eficacia de un sistema de PLN real, como por ejemplo un sistema de traducción automática (MT) en el cual evidentemente es necesario traducir *todas* las palabras (no vacías) de un corpus.

2.7.2 Uso de corpus para la evaluación

En ambos tipos de tareas se utiliza un corpus etiquetado de sentidos. En la tarea de muestra léxica el corpus suele estar formado por un número relativamente alto de oraciones en las que aparecen con naturalidad muchas ocurrencias de cada palabra objetivo de la muestra léxica. Todas estas apariciones están etiquetadas de alguna forma con un sentido asignado manualmente.

En un sistema supervisado, una parte de las ocurrencias etiquetadas se reserva para el entrenamiento y otra para el test. La parte reservada para el test se puede utilizar también para el test (y sólo para el test) de sistemas no supervisados (tanto en el sentido estricto del término como en el sentido de sistemas que utilizan fuentes de conocimiento externo, pero no un corpus etiquetado de entrenamiento) que se quieran comparar muy directamente con el sistema supervisado.

La elección de la proporción entre el número de ejemplos de entrenamiento y el de ejemplos de test depende de los objetivos de la evaluación. Si el objetivo es medir la mejor estimación de la eficacia del sistema se puede utilizar una proporción grande de datos de entrenamiento, en una relación de 10:1 ó 5:1. Pero si se quiere una estimación más realista de la eficacia del sistema, la proporción debe ser mucho más baja, del orden de 2:1, debido simplemente al número relativamente bajo de datos etiquetados disponibles en la realidad, como consecuencia del problema del CBAC. Este problema llega a su máxima expresión en la tarea de todas las palabras, ya que no existe ningún corpus etiquetado manualmente en todas sus palabras no vacías de dimensiones suficientes para una evaluación mínimamente realista de un sistema en ese tipo de tarea. Por tanto, esa tarea se suele reservar para todos los sistemas excepto los supervisados (en sentido estricto).

2.7.3 La puntuación en los sistemas de evaluación

En general un sistema de evaluación dispone de una parte del corpus etiquetado que se dedica a la evaluación (test) del algoritmo que se está probando. Normalmente esa parte etiquetada consiste en una serie de ejemplos que contienen ocurrencias en el corpus de

la palabra objetivo, a los que se les ha adjudicado un sentido o etiqueta, que se supone es el sentido correcto en cada ejemplo.

Dado un sistema de evaluación como este, el método empleado para puntuar la eficacia de un algoritmo bajo prueba es inmediato: cada vez que el algoritmo acierta en el sentido de una aparición de la palabra objetivo en un ejemplo, es decir, su predicción coincide con la etiqueta, se le asigna una puntuación de 1; y cada vez que falla, es decir, su predicción no coincide con la etiqueta, se le asigna una puntuación de 0.

Este enfoque básico puede complicarse de varias formas. Una de ellas ocurriría cuando el sistema bajo prueba no asigna una única predicción de sentido a una aparición dada de la palabra objetivo, sino que asigna varias etiquetas diferentes, cada una con una probabilidad asociada. En este caso el mecanismo de puntuación sería adjudicar la probabilidad del sentido correcto, es decir, si la etiqueta correcta es *c* dada la palabra *w* y su contexto la puntuación sería:

$$puntuación = \Pr(c \mid w, contexto(w))$$

En el caso de que hubiera más de un sentido correcto se sumarían las probabilidades de cada sentido correcto:

$$puntuación = \sum_{t=1}^{C} \Pr(c_t \mid w, contexto(w))$$

donde C sería el número de sentidos correctos.

Si el inventario de sentidos está organizado de forma jerárquica se distinguen tres métodos de puntuación según el nivel aplicado de granularidad de los sentidos: granularidad fina, granularidad gruesa y granularidad mixta. Cuando se utiliza un nivel de granularidad fino, sólo puntúan las coincidencias de etiquetas, que además se refieren a ese tipo de granularidad. En cambio, en el nivel de granularidad gruesa, tanto los sentidos predichos por el sistema bajo prueba como las etiquetas que definen los sentidos ideales se propagan hacia arriba a través de la jerarquía hasta un nivel alto (de granularidad gruesa) y si los sentidos originales se juntan en el mismo sentido del nivel alto se recibe una puntuación de 1. Si no coinciden la puntuación será de 0.

En un sistema de puntuación de granularidad mixta, al sistema bajo prueba se le permite predecir cualquier sentido en cualquier nivel de la jerarquía, y lo mismo ocurre con las etiquetas asignadas a los sentidos ideales por las personas que elaboran el corpus de test. Cuando el sistema predice un sentido descendiente del correcto se puntúa 1. Cuando predice un sentido ascendiente del correcto se puntúa entre 0 y 1 de forma inversamente proporcional al número de descendientes. Cuando el sentido predicho no es predecesor ni sucesor del correcto se puntúa 0.

2.7.4 La línea de base del sentido más frecuente

La precisión obtenida por el sistema bajo prueba se suele comparar con una *línea de base* (en inglés *baseline*) que suele indicar una eficacia mínima del algoritmo. Naturalmente también se compara con la precisión obtenida por otros algoritmos bajo el mismo corpus de prueba.

La línea de base más común es la llamada línea de base del sentido más frecuente [Gale et al. 1992], que indica la precisión obtenida por un sistema que asignara como sentido de todas las apariciones de la palabra objetivo el sentido más frecuente de ésta en el corpus. Esta línea de base indica las prestaciones mínimas de un sistema de DSP y nos indica hasta qué punto el esfuerzo extra generado por el sistema bajo prueba merece la pena.

2.7.5 Acuerdo entre anotadores (inter-annotator agreement)

El *acuerdo entre anotadores* (*inter-annotator agreement, ITA*) es, al menos en teoría, un límite superior en cuanto a la precisión de los sistemas de DSP automáticos, suponiendo que éstos no pueden superar a las personas en esa tarea sobre los mismos datos, o sobre datos comparables [Gale et al. 1992]. El ITA se calcula comparando las anotaciones o etiquetados de dos o más personas entrenadas con los mismos criterios sobre los mismos datos. El acuerdo se calcula como porcentaje de coincidencias totales sobre el total de etiquetados; o bien, si se permiten etiquetados múltiples, puede contarse el solapamiento. Existen medidas más sofisticadas que miden el acuerdo entre anotadores eliminando las coincidencias que se puedan deber al azar, como el coeficiente *kappa* [Cohen 1960][Bruce y Wiebe 1999][Ng et al. 1999], aunque no están definidas para etiquetas múltiples.

[Kilgarrif y Rosenzweig 2000] han propuesto una medida de precisión máxima alternativa al ITA llamada replicabilidad. Consiste en repetir todo el proceso de obtención de las etiquetas ideales y comparar el acuerdo entre los dos conjuntos de etiquetas ideales. El problema de este método es su gran coste de producción, por lo que normalmente se utiliza el ITA.

El ITA depende de muchos factores, entre ellos la elección de palabras objetivo, la calidad del inventario de sentidos, la calidad de los ejemplos seleccionados y la adecuación del inventario de sentidos al corpus a etiquetar. Un nivel de ITA alto confirma la calidad del etiquetado. Además, un inventario de sentidos de alta calidad con sentidos bien diferenciados y apropiado al corpus, facilita la obtención de un nivel alto de ITA. La inspección de los desacuerdos entre anotadores puede ayudar a revisar y mejorar el inventario de sentidos, haciendo las distinciones de sentidos más claras y definidas.

[Gale et al. 1992] propusieron utilizar la precisión del sentido más frecuente y el ITA como límites mínimo y máximo de precisión de sistemas de DSP. En su trabajo utilizaron ambigüedades de dos sentidos y estimaron esos límites en 75% y 96.8% respectivamente. Naturalmente los límites son bastante más bajos para ambigüedades de más de dos sentidos.

2.7.6 La evaluación en las competiciones Senseval

La iniciativa Senseval[4] es el primer ejercicio abierto de evaluación de sistemas de DSP. Empezó en 1997 y está auspiciada por ACL-SIGLEX (Association for Computacional Linguistics' Special Interest Group on the Lexicon). El formato de las evaluaciones está basado en los formatos de la DARPA y se basa en suministrar a los participantes datos etiquetados manualmente, tanto para entrenamiento como para prueba (test) y con una métrica de evaluación predefinida. El fin último de Senseval es la profundización en la comprensión de la semántica léxica y la polisemia. En las tres ediciones que ha habido hasta el momento se han puesto de manifiesto algunas cuestiones importantes, entre las que destacan la dificultad del etiquetado de sentidos y el problema del nivel de granularidad de éstos.

Senseval-1

Senseval-1 [Kilgarrif 1998][Kilgarrif y Palmer 2000] fue el primer ejercicio de evaluación de sistemas de DSP automáticos en el idioma inglés y se celebró en 1998[5]. Como inventario léxico se utilizó el léxico Hector [Atkins 1993]. Este léxico fue desarrollado conjuntamente por DEC y Oxford University Press y utiliza una metodología orientada al corpus y entradas de diccionario jerárquicas tradicionales. De este léxico se seleccionó aleatoriamente una muestra de treinta y dos palabras objetivo, con subconjuntos muestra según categoría gramatical (sustantivo, verbo y adjetivo), frecuencia y número de sentidos.

Utilizando el corpus Hector (un corpus predecesor del British Nacional Corpus, BNC) se extrajeron ejemplos que contuvieran las palabras objetivo y un grupo de lexicógrafos profesionales etiquetaron esas palabras con sentidos según el léxico Hector. Se permitió la discusión y revisión de entradas confusas y se alcanzó un ITA de algo más del 80%. Hubo una participación de veinticuatro sistemas de los tres tipos: basados en fuentes de conocimiento externo, supervisados y no supervisados (en sentido estricto).

La metodología de evaluación fue propuesta por [Melamed y Resnik 2000] y utilizó dos métodos: uno basado en coincidencias totales de sentidos de granularidad fina y otro

[4] http://www.senseval.org
[5] Un ejercicio para los idiomas francés e italiano (Romanseval) se celebró en paralelo [Segond 2000] [Calzolari and Corazzari 2000].

basado en coincidencias parciales de sentidos de granularidad más gruesa. Se utilizaron varios algoritmos sencillos como líneas de base: RANDOM (asignación aleatoria de sentidos con distribución uniforme), COMMONEST (línea de base del sentido más frecuente), y algunas variantes del algoritmo sencillo de LESK [Lesk 1986] (véase la sección 2.1.2).

El orden relativo de los sistemas fue independiente de la métrica empleada y el mejor sistema obtuvo una precisión de 77.1% en la métrica de granularidad fina y de 81.4% en la de granularidad gruesa. La precisión más alta considerando sólo verbos fue de 70.5% y la polisemia media de éstos (número medio de sentidos) fue de 7.79, por tanto un nivel alto de granularidad fina. En relación con las líneas de base utilizadas, resultó que ningún sistema logró una eficacia más alta que la mejor línea de base (una de las variantes del algoritmo de LESK) [Kilgarriff y Rosenzweig 2000].

Senseval-2

Aunque Senseval-1 demostró que los sistemas automáticos de DSP podían funcionar bien, dados un inventario de sentidos y unos datos de entrenamiento adecuados, el léxico utilizado (Hector) se consideró un léxico muy reducido y además no era de acceso público ilimitado.

En Senseval-2 [Edmonds y Cotton 2001] se incluyeron tareas de DSP en 10 idiomas diferentes[6] y se intentó utilizar como inventario de sentidos las WordNet existentes en ese momento por considerarse un recurso asequible y gratuito. Se incluyeron tareas de *muestra léxica* y de *todas las palabras* en casi todos los idiomas y en algunos los dos tipos de tareas. A continuación se describen los dos tipos de tareas para el idioma inglés.

La tarea de todas las palabras en inglés

Esta tarea utilizó un texto de 5000 palabras obtenido de tres artículos del Penn Treebank II [Palmer et al. 2001] pertenecientes a géneros diferentes. Las personas encargadas del etiquetado fueron dos estudiantes de lingüística, y una tercera persona corrigió y adjudicó los sentidos finales. Los participantes recibieron sólo los datos de prueba (test), de forma que los sistemas supervisados (en sentido estricto) tuvieron que utilizar otros datos etiquetados de entrenamiento (por ejemplo Semcor); los sistemas supervisados (en sentido laxo) utilizaron bases de datos externas, como frases en definiciones de diccionarios. Se utilizó una línea de base siguiendo la estrategia simple de anotar cada palabra con el primer sentido de WordNet (los sentidos de WordNet suelen estar ordenados de mayor a menor frecuencia) cuya categoría gramatical coincidiera con la

[6] Vasco, chino, checo, danés, holandés, inglés, estonio, italiano, japonés, coreano, español y sueco, aunque el chino y el danés no estuvieron preparados a tiempo, y no hubo participantes en el holandés.

correspondiente del Penn Treebank (este corpus contiene información sintáctica). Esta línea de base logró una precisión del 57%, mientras que el mejor sistema, de la Southern Methodist University, logró una marca del 69%. El límite máximo representado por el ITA se estimó en un 80%.

La tarea de muestra léxica en inglés

Esta tarea en Senseval-2 fue preparada por la Universidad de Pennsylvania (datos de entrenamiento y test para verbos) y la Universidad de Brighton (datos de entrenamiento y test para sustantivos y adjetivos) [Kilgarrif 2001][Palmer et al. 2001]. Los datos se extrajeron en su mayoría del corpus Penn Treebank II (parte Wall Street Journal). Cuando no hubo suficientes ejemplos de alguna palabra objetivo se añadieron del British Nacional Corpus (BNC). Las palabras objetivo se obtuvieron de WordNet 1.7. Se eligieron 73 sustantivos, adjetivos y verbos, con un total de entre 75 y 300 ejemplos de cada uno, dependiendo del número de sentidos. La relación entre datos de entrenamiento y de test fue de 2:1, tanto para hacer una evaluación realista (no demasiados ejemplos de entrenamiento) como para equilibrar la evaluación entre sistemas supervisados y sistemas no supervisados.

Se eligieron 29 verbos de entre los más polisémicos de la tarea de todas las palabras para la tarea de muestra léxica. La polisemia media de estos verbos fue de 16.28 sentidos, un número bastante elevado. Dos lingüistas anotaron independientemente los ejemplos del corpus y un tercer lingüista resolvió conflictos entre ellos para crear las etiquetas de sentidos ideales. El ITA para los verbos fue de un 71%, más bajo que en Senseval-1, y para sustantivos y adjetivos fue de un 85.5%, dado el carácter menos polisémico de éstos.

Dado el carácter más polisémico de los verbos, la atención se centró en ellos, y un grupo de personas los agrupó para aplicar una métrica de evaluación de granularidad más gruesa. El ITA de los verbos subió al 82% utilizando estos grupos.

Como líneas de base se utilizaron las mismas que en Senseval-1, pero a diferencia de este certamen, en Senseval-2 la mitad de sistemas lograron marcas más altas que la mejor línea de base (una variante de LESK, con un 45.5% de precisión). El mejor sistema en la tarea de muestra léxica para verbos obtuvo un 57.6% [Palmer et al. 2001]. En la tarea de muestra léxica en total las marcas más altas fueron de 64.2% (granularidad fina) y 71.3% (granularidad gruesa) obtenidas por un sistema de la Johns Hopkins University. En la Tabla 2.7 se puede ver un resumen de los resultados de Senseval-2. La tarea AW se refiere a la tarea *All Words*, en la que se desambiguan todas las palabras no vacías del corpus, y LS se refiere a la tarea *Lexical Simple*, en la que se hace una selección de palabras a desambiguar. La columna *Línea base* se refiere a la referencia del porcentaje del sentido más frecuente.

Idioma	Tarea	Sistemas	Lemas	Ejemplos	ITA (%)	Línea base	Marca
Inglés	AW	21	1082	2473	75	57	69/55
Estonio	AW	2	4608	11504	72	85	67
Vasco	LS	3	40	5284	75	65	76
Inglés	LS	26	73	12939	86/71	48/16	64/40
Italiano	LS	2	83	3900	21	-	39
Japonés	LS	7	100	10000	86	72	78
Coreano	LS	2	11	1733	-	71	74
Español	LS	12	39	6705	64	48	65
Sueco	LS	8	40	10241	95	-	70

Tabla 2.7. Resumen de los resultados obtenidos en Senseval-2. El ITA en la tarea LS (lexical sample) del idioma inglés es para sustantivos y adjetivos a la izquierda de la barra y para verbos a la derecha. En las columnas 'línea de base' y 'marca' cuando hay barra a la izquierda es para sistemas supervisados y a la derecha para sistemas no supervisados. Cuando no hay barra es para sistemas supervisados.

En general, los sustantivos y adjetivos tuvieron menor polisemia (4.9), mayor ITA (85.5%) y mejor precisión (64% para granularidad fina) [Yarowsky et al. 2001] que los verbos. Los resultados para la tarea de muestra léxica (LS) de los otros idiomas son similares, aunque no se puede hacer una comparación directa, porque los criterios no son completamente análogos.

Los inventarios de Senseval-1 y Senseval-2 y el nivel de polisemia

En los dos primeros certámenes de Senseval se utilizaron inventarios de sentidos muy diferentes. En Senseval-2 se utilizó WordNet como inventario de sentidos por primera vez y anteriormente a este hecho había dudas sobre si esta base de datos léxica cumplía con el requisito necesario de ofrecer distinciones de sentidos claras y consistentes. Los resultados de ambas competiciones tanto en ITA como en precisión ofrecen una oportunidad para comparar la influencia de las características diferenciales de ambos inventarios.

En principio tanto el ITA como la precisión fueron más bajos en términos absolutos en Senseval-2 que en Senseval-1, lo cual podría ser interpretado como prueba de lo fundamentado de esas dudas. Sin embargo, un análisis de una de las características más importantes de un inventario de sentidos, como es el nivel de polisemia, puede explicar estos resultados. En concreto, un grupo de palabras tan importante como son por ejemplo los verbos, tienen un nivel medio de polisemia del doble en Senseval-2 que en Senseval-1, de 16.28 frente a 7.79. Está demostrado que un nivel alto de polisemia tiene efectos negativos en el etiquetado manual (ITA) y en el etiquetado automático (precisión), aunque no está tan correlacionado con este efecto como lo está el nivel de entropía [Palmer et al. 2001].

Otra diferencia importante entre los dos inventarios está en la cantidad de datos de entrenamiento: de media hubo la mitad de ejemplos de entrenamiento para los verbos en Senseval-2 que en Senseval-1. Sin embargo, un estudio comparativo entre palabras con

nivel de polisemia similar, aunque diferente cantidad de ejemplos, de los dos ejercicios, arrojó unos resultados de precisión muy similares [Palmer et al. 2006]. Con lo que se puede concluir que *el nivel de polisemia de un inventario afecta negativamente a la precisión de un sistema de DSP*. Podemos estar seguros de que el nivel más bajo de ITA se debe también a este motivo, y que el sistema de doble etiquetado por especialistas asegura que hay una correlación consistente entre los datos etiquetados y los sentidos de WordNet. También podemos asegurar que el número diferente de ejemplos de entrenamiento no fue un factor decisivo.

2.7.7 Causas de los desacuerdos en el ITA

La dificultad de lograr ejemplos etiquetados con sentidos con alta precisión es un asunto bastante tratado en la literatura [Kilgarriff 1997][Hanks 2000]. El hecho de que incluso especialistas como son los lexicógrafos tomaran decisiones respecto a los sentidos tan diferentes pudo verse ya en Senseval-1, donde estuvo disponible una correspondencia o relación (mapping) entre los dos inventarios, Hector y WordNet 1.6. No sólo los mismos lemas tenían a menudo un número de sentidos diferentes. También es evidente el uso de criterios de decisión muy diferentes para distinguir sentidos. Y además los criterios usados en todos los casos no por ser diferentes son menos válidos, por lo que no se puede afirmar que un criterio sea el 'bueno' y otro el 'equivocado'.

En Senseval-2, donde el ITA varió desde 28.8% (*train*) hasta 90.8% (*serve*), se eligieron 50 ejemplos de algunos verbos, como categoría gramatical más polisémica, distribuidos uniformemente según la frecuencia ideal de los sentidos; estos ejemplos se volvieron a etiquetar por dos especialistas y se estudiaron detenidamente los desacuerdos producidos. Se identificaron al menos cuatro causas claras de errores de etiquetado: subsumisión de sentidos, entradas de diccionario deficientes, usos vagos y sentido común [Fellbaum et al. 2001].

Subsumisión de sentidos

Se detectaron varios desacuerdos en el verbo *develop* derivados de la posible elección entre un sentido más general o más específico, un problema clásico de la lexicografía [Fellbaum et al. 2005]. Dos sentidos frecuentemente confundidos de esa palabra se refieren a la creación de entidades nuevas, bien "products, or mental or artistic creations" (sentido 1, creación física), o bien "a new theory of evolution" (sentido 2, creado por un acto mental). El 25% de los desacuerdos con esta palabra surgió de decidir cuál de esos dos sentidos se debería aplicar a una frase como "develop a better way to introduce crystallography techniques". Ambos sentidos valen, dependiendo de si 'ways' se considera como 'cosas' o como 'teorías'. Como en la definición del sentido 1 se menciona explícitamente 'mental creations' además de otros tipos de creaciones, se puede considerar que el sentido 1 subsume al sentido 2. Además, estos sentidos más

generales aportan la flexibilidad necesaria para absorber usos nuevos, que constituyen otro problema importante del etiquetado de sentidos.

Entradas de diccionario deficientes

Otra fuente de desacuerdos se puede deber a diversas deficiencias del inventario manejado por los especialistas: falta de entradas, entradas redundantes, palabras ambiguas en las glosas o incluso ejemplos contradictorios en éstas. Un caso especialmente frecuente de este tipo de errores se debe a la imposibilidad práctica de que un inventario general abarque sentidos nuevos, inusuales o específicos de un determinado dominio. Como ejemplo de ambigüedad, el 16% de los desacuerdos debidos a la palabra *develop* se debieron a la confusión entre decidir si *understanding* en la frase "develop a much better understanding of …" representaba un atributo (sentido 3) o una característica física (sentido 4). Además, en este caso ninguno de los dos sentidos es suficientemente general como para subsumir al otro.

Usos vagos

Este tipo de errores se debe a que en muchos contextos una palabra se puede referir a varios sentidos a la vez. Un caso típico son los juegos de palabras, pero ocurre frecuentemente en otros casos. Por ejemplo, en Senseval-1, la palabra *onion* (cebolla), que normalmente tiene los sentidos de comida y de planta, se refiere a ambos en la frase "planning, harvesting y marketing onions" [Krishnamurthy y Nicholls 2000]. En Senseval-2 la frase "he played superbly" se utilizaba en un contexto en el que claramente se tocaba música, pero la palabra *play* podía referirse al instrumento (sentido 3) o a la melodía (sentido 6) o incluso a los dos.

Sentido común

Esta es una de las fuentes de desacuerdo más difíciles de resolver. No se debe a diferencias sintácticas o semánticas entre los argumentos de la palabra objetivo, sino al sentido común ó conocimiento de la realidad sobre la que trata el contexto de ésta. Por ejemplo, en Senseval-2, el 58% de los desacuerdos sobre la palabra *develop* se debieron a este tipo de causas. Entre ellos varios se referían al *development* de *cancer tumors*. Si se considera que un tumor se desarrolla espontáneamente, como un movimiento religioso, estaríamos ante el sentido 5, pero si se desarrolla como un producto de crecimiento natural o evolución, como una flor, estaríamos ante el sentido 10. Claramente es necesario un gran conocimiento de la realidad, en este caso la medicina, para poder distinguir los dos sentidos. En este caso es evidente que un sentido más general que subsumiera ambos resolvería el problema.

2.7.8 La granularidad: los grupos de sentidos en WordNet

Como se ha mencionado en la sección 2.7.6, en Senseval-2 se formaron grupos de verbos de WordNet manualmente, para efectuar una evaluación de granularidad más gruesa, y este sistema de evaluación tuvo un impacto considerable tanto en las marcas de precisión como en las de ITA.

La razón de ser básica de estos grupos se deriva del hecho de que muchas veces es mejor no intentar asignar un determinado sentido muy específico (granularidad fina) sino elegir un grupo de sentidos menos específico (granularidad más gruesa).

Aunque WordNet aporta una gran cantidad de información de herencia como hiperónimos y conjuntos de sinónimos (synsets), estas jerarquías no se avienen con naturalidad a la formación de jerarquías de sentidos para la formación de grupos de éstos que puedan ayudar a realizar una evaluación de granularidad gruesa [Lin 1998][Mihalcea y Moldovan 2001]. La razón de este hecho queda de manifiesto observando la variación de los hiperónimos en la mayoría de grupos de Senseval-2. En la Tabla 2.8 se puede ver cómo uno de los grupos de la palabra *play* contiene tres de los sentidos de WordNet, todos sobre la producción de música con instrumentos musicales; a pesar de la similitud de los tres sentidos, cada uno tiene un hiperónimo de WordNet diferente.

Sentido	Glosa de WordNet	Hiperónimo
3	Play (music) on an instrument	*perform*
6	Play a melody	*recreate*
7	Perform music (on a musical instrument)	*sound*

Tabla 2.8. Muestra los hiperónimos de tres sentidos diferentes de WordNet de la palabra *play* que quedan agrupados dentro del mismo grupo de sentidos efectuado manualmente para Senseval-2. Este grupo engloba tres sentidos, pero el resto de sentidos queda englobado en otros gupos diferentes.

Los grupos se hicieron después de que los corpus se hubieran terminado de etiquetar, es decir, los especialistas no etiquetaron con los grupos ya formados sino con los sentidos básicos. En otras palabras, los grupos se utilizaron exclusivamente como medio de evaluación. Aunque estrictamente algunos sentidos originales pudieran pertenecer seguramente a más de un grupo, se siguió el esquema más sencillo posible de agrupamiento, en el que no se permite el solapamiento de grupos. Además los grupos fueron formados por personas distintas de los etiquetadotes, y sin ningún tipo de referencia a apariciones de las palabras en los corpus etiquetados. Los grupos se formaron por dos personas independientemente siguiendo criterios sintácticos y semánticos comunes y las discrepancias entre ellos se discutieron y se adjudicaron por un tercer especialista [Fellbaum et al. 2001].

Criterios sintácticos

La estructura sintáctica se utilizó de dos formas diferentes para realizar los grupos. En la primera forma, se hizo uso de la propiedad frecuente de la sintaxis de ser reflejo de la semántica, para distinguir distintos sentidos de una misma palabra. En concreto, es muy frecuente que cuando un verbo sufre diferencias grandes en su forma de subcategorización, los sentidos a los que se refiere también sufran diferencias grandes, suficientes para incluirlos en grupos de sentidos distintos. Por ejemplo la frase *John left the room* frente a la frase *Mary left her daughter-in-law her pearls in her will* [Palmer, Ng y Dang 2006]. Como consecuencia, se puede aplicar un filtro sintáctico sencillo para discriminar entre los sentidos (y entre los grupos de sentidos). La segunda forma trabaja casi a la inversa: estructuras sintácticas similares pueden ser indicativos de que varios sentidos se pueden agrupar juntos, indicando que los cambios de sentido son muy leves [Levin 1993].

Criterios semánticos

Los criterios semánticos son más variados: se juntaron varios sentidos en un grupo cuando todos eran versiones especializadas del mismo sentido más general; se mantuvieron sentidos separados en grupos diferentes cuando había diferencias entre la semántica de sus argumentos (abstracto frente a concreto, animal frente a humano, animado frente a inanimado, tipos de instrumentos diferentes, etc.); cuando había diferencias en el tipo y número de argumentos (muchas veces coincidiendo con el criterio de subcategorización sintáctico revisado antes); cuando había diferencias entre causa y efecto; cuando había diferencias en el tipo de acontecimiento (abstracto, concreto, mental, emocional, etc.); cuando había un dominio especializado, etc. En la Tabla 2.9 se ilustran los cuatro grupos más importantes de la palabra *develop*, sin incluir otros tres referidos a dominios muy específicos: ajedrez, cine y matemáticas.

Análisis de los resultados de las agrupaciones

Como se mencionó antes el ITA para los verbos de WordNet 1.7 pasó de ser de un 71.3% sin las agrupaciones manuales a un 82% con dichas agrupaciones. Estas marcas suponen sólo un cambio en el sistema de puntuación de la evaluación, ya que los datos se etiquetaron previamente a la formación de los grupos. Para compararlos con una línea de base, y probar que la mejora en el ITA no se debe únicamente a la disminución del número de sentidos posibles de cada palabra, se formaron grupos nuevos aleatoriamente y con el mismo número de ellos por palabra objetivo. El ITA nuevamente mejoró, pero sólo hasta un 74% desde el 71%, lo que confirma la coherencia de las agrupaciones manuales.

En estudios posteriores se repitió el etiquetado de los mismos datos pero utilizando los grupos ya hechos, lo que produjo un incremento del ITA hasta un 89%, y además la velocidad del etiquetado manual se cuadruplicó [Weischedel y Palmer 2004]. [Palmer, Ng y Dang 2006] afirman que los grupos muestran evidentemente una coherencia semántica, pero que, al perderse diferencias de sentidos, "queda por ver si las agrupaciones pueden ser efectivas en aplicaciones de PLN".

Grupo	Sentido	Glosa de WordNet	Hiperónimo
1 nuevo (abstracto)	1	Products, or mental creations	Create
	2	Mental creations: "new theory"	Create
2 nuevo (propiedad)	3	Personal attribute: "a passion for…"	Change
	4	Physical characteristic: "a beard"	Change
3 nuevo (espontáneo)	5	Originate: "new religious movement"	Become
	9	Gradually unfold: "the plot …"	Occur
	10	Grow: "a flower developed…"	Grow
	14	Mature: "the child developer…"	Change
	20	Happen: "report the news as it…"	Occur
4 mejora	6	Resources: "natural resources"	Improve
	7	Ideas: "ideas in your thesis"	Theorize
	8	Train animate beings: "violinists"	Teach
	11	Civilize: "developing countries"	Change
	12	Make, grow: "develop the grain"	Change
	13	Business: "develop the market"	Generate
	19	Music: "develop the melody"	Complicate

Tabla 2.9. Muestra los cuatro grupos de sentidos principales desarrollados manualmente para Senseval-2 a partir de WordNet 1.7 para la palabra *develop*. Como se puede observar, se ha pasado de un total de 16 sentidos inicialmente a sólo 4, después del agrupamiento.

Lenguaje	Tarea	Sistemas	Lemas	Ejemplos	ITA (%)	Línea base	Marca
Inglés	AW	26	-	2081	62	62/-	65/58
Vasco	LS	8	40	7362	78	59	70
Catalán	LS	7	27	6721	93	66	85
Inglés	LS	47	57	-	67	55/-	73/66
Italiano	LS	6	45	7584	89	18	53
Rumano	LS	7	39	11532	-	58	73
Español	LS	9	46	12625	83-90	67	84

Tabla 2.10. Muestra un resumen de los principales resultados de Senseval-3. Las tareas son: AW, *all words* (todas las palabras), LS *lexical sample* (muestra léxica). La línea de base de referencia es la del sentido más frecuente. Las marcas con barra significan sistema supervisado a la izquierda y sistema no supervisado a la derecha. Si no hay barra es sistema supervisado.

2.7.9 El certamen Senseval-3

Este certamen [Mihalcea y Edmonds 2004] es el último celebrado hasta el momento. Tuvo lugar en 2004 a la vez que ACL-2004. Esta vez se incluyeron 14 tareas diferentes y se presentaron 160 sistemas por 55 equipos participantes. Para el etiquetado de sentidos de los corpus de trabajo se siguieron los protocolos que se usaron en Senseval-2. Se añadió una tarea nueva de etiquetado semántico que requería el etiquetado de roles semánticos para lo que se utilizó FrameNet [Baker et al. 2003]. También se añadió una tarea nueva de desambiguación de glosas de WordNet. En la Tabla 2.10 se puede ver un

resumen de los resultados más importantes. Como en la Tabla 2.7, AW se refiere a la tarea *All Words* y LS se refiere a la tarea *Lexical Sample*. La línea de base también es la del sentido más frecuente.

En este certamen, la tarea de muestra léxica del idioma inglés intentó evitar los altos costes representados por el etiquetado de corpus por parte de especialistas lingüísticos, utilizando los resultados del proyecto OMWE [Chklovski y Mihalcea 2002], revisado en la sección 3.1.1. Se utilizó una parte de los datos etiquetados producidos por este proyecto que incluyeron unos 12.000 ejemplos etiquetados de 59 palabras objetivo. El ITA obtenido por este procedimiento fue relativamente bajo, 67% [Mihalcea et al. 2004], frente al ITA de 85.5% obtenido en Senseval-2 [Kilgarriff 2001]. Este hecho se puede explicar porque el etiquetado fue realizado por usuarios voluntarios de la web, en lugar de por especialistas en lingüística. Sin embargo todavía no hay explicación para la marca sorprendentemente alta de precisión lograda en esta tarea, de un 72%. El hecho de que los sistemas automáticos puedan superar a los etiquetadores humanos se explica porque los sentidos ideales se adjudican *después* de que se haya calculado el ITA. Nótese que el porcentaje de 'acuerdo' logrado por los sentidos ideales es de un 100% por definición.

En Senseval-3 se presentaron bastantes técnicas nuevas de DSP. De entre ellas, las que obtuvieron los mejores resultados fueron las técnicas supervisadas de aprendizaje automático (*machine learning*, ML) que utilizan agregaciones de varios rasgos (en inglés features) como las SVMs (Support Vector Machines). Sin embargo, una de las principales conclusiones de Senseval-3, continuando con la tradición de las anteriores ediciones, es la importancia del inventario de sentidos: cuando la calidad de éste es baja, el ITA tampoco es alto y le acompañan unas marcas de precisión de los sistemas también bajas. Esto también está relacionado con el problema de la granularidad de los sentidos del inventario (véase capítulo 4). En un panel de discusión del certamen se llegó a la conclusión generalizada de que la tarea de DSP *in vitro* tal como se había venido desarrollando en las Senseval había tocado techo y no daría lugar a líneas de investigación realmente nuevas. Quizás por ello, en otro panel se acordó que las aplicaciones de PLN jugarían un papel importante en Senseval-4: se habló de la DSP como selección léxica en MT y como equivalencia de sentidos en IR. Se espera que de esta manera se arroje luz sobre las cuestiones vigentes, entre ellas la granularidad del inventario de sentidos.

2.8 Los sistemas de DSP en aplicaciones de PLN

Para esta revisión de las aplicaciones prácticas en PLN de los sistemas de DSP nos hemos basado en el artículo de 2006 de Resnik [Resnik 2006].

Al tratar sobre el papel de la DSP en el Procesamiento del Lenguaje Natural (PLN) es útil la distinción entre *aplicación* y *tecnología coadyuvante* (*enabling technology*). Así,

una tecnología coadyuvante produce un resultado que no es útil por sí mismo; en cambio, una aplicación lleva a cabo una tarea que produce un resultado que tiene un valor directo para el usuario final, y en la obtención de este resultado puede intervenir una tecnología coadyuvante.

En este sentido, la DSP es una tecnología coadyuvante [Agirre y Edmonds 2006], como lo son otras tecnologías habituales de PLN, como la determinación de la categoría gramatical (*part-of-speech tagging*) o el análisis sintáctico (*parsing*). En cambio, la traducción automática (MT) o la transcripción automática de lenguaje hablado sí pueden considerarse como aplicaciones en su conjunto.

Una analogía de [Resnik 2006] compara un transformador de tensión entre 220 y 110 voltios con una tecnología coadyuvante, porque en sí mismo no tiene una relación directa con las necesidades del usuario. En cambio una maquina de afeitar eléctrica representa en su conjunto una aplicación que tiene un valor determinado para un usuario. Si este quiere utilizarla tanto en Europa como en EEUU, puede utilizar el transformador como tecnología coadyuvante.

La analogía con el transformador sirve para ilustrar otra característica de la DSP considerada aisladamente: mientras un transformador de tensión tiene muy bien definida su tarea, esto es, convertir corriente eléctrica de N voltios a M voltios dentro de una tolerancia determinada, y esta tarea es independiente de la aplicación final que lo utilice, sea la máquina de afeitar, una máquina de café o un televisor, no existe una definición generalmente aceptada de la "tarea" que debería hacer un sistema de DSP independientemente de la aplicación final que lo utilice. Por supuesto esta afirmación es perfectamente compatible con el hecho de que en los certámenes Senseval, por ejemplo, se formalice una tarea concreta a realizar por todos los sistemas participantes con el objetivo de comparar su eficacia lo más fielmente posible.

La dependencia de la definición de la tarea de DSP de la aplicación final concreta nos lleva a la cuestión central de esta sección: en qué aplicaciones de PLN la DSP explícita produce resultados tangibles positivos para la aplicación y por qué. Esta sería la pregunta cuya respuesta nos daría a su vez la respuesta a otra pregunta todavía más general: la razón de ser de la DSP. Esta última cuestión ha dado lugar a varios tipos de respuesta o argumentos a favor de su existencia: los llamados argumentos de fe, argumento por analogía y, como veremos, el más fuerte, que es el argumento de su utilidad o no en aplicaciones finales específicas.

2.8.1 Argumentos de fe

Los argumentos de fe son variantes de lo que [Dawkins 1986] llama "argumento de incredulidad personal" y que podríamos parafrasear como "esto debe ser cierto porque lo contrario sería increíble". En este caso se suele argumentar que, por ejemplo, si la

palabra *bank* tiene dos sentidos como "institución financiera" y como "orilla de un río", un sistema de IR que la utilice en una consulta, u otro tipo de sistema de PLN que la utilice con otros fines, deberá devolver documentos irrelevantes a menos que se especifique el sentido al que se refiere desambiguándolo de alguna forma.

Aunque este argumento se amolda muy bien a las intuiciones habituales, y de hecho se ha mantenido durante mucho tiempo, los hechos reales son, primero, que la DSP explícita juega un papel muy limitado en muchas aplicaciones PLN reales, y segundo, que esto no ha impedido que dichas aplicaciones reales sigan progresando considerablemente.

Estos hechos indican, como mínimo, que en vez de dar por supuesta la importancia de la DSP, se examine su papel en las aplicaciones reales. Como se argumenta en el capítulo 4, y en línea con la consideración de la granularidad de WordNet como demasiado fina, no está tampoco claro que las aplicaciones de PLN requieran un proceso que implique "la determinación de todos los sentidos diferentes de todas las palabras" o que necesite "asignar cada aparición de una palabra al sentido apropiado" [Ide y Véronis 1998] citado en [Resnik 2006].

2.8.2 Argumentos por analogía

El desarrollo de la tecnología lingüística durante las últimas dos décadas ha supuesto una revolución sobre la subdivisión tradicional del PLN en módulos relativamente independientes organizados en forma de cascada: análisis morfológico, análisis sintáctico, desambiguación de sentidos, representación semántica en *forma lógica* (*logical form*), análisis del discurso, etc.

El desarrollo de tecnologías coadyuvantes para mejorar esos subproblemas aisladamente ha jugado un papel muy escaso en los mayores avances logrados en tecnología de lenguaje natural para los usuarios finales, como por ejemplo la viabilidad comercial del reconocimiento automático del habla, la generalización del chequeo ortográfico de texto, o la incorporación de la recuperación de información textual de la web a la vida diaria. Estos logros se deben principalmente a técnicas escasamente lingüísticas tales como los n-gramas y el Modelado Oculto de Markov (Hidden Harkov Modeling), la lematización (stemming), la representación mediante vectores de coocurrencia de rasgos, etc. Todas estas son tecnologías lingüísticamente poco "profundas", en el sentido de poco alejadas de la forma visible en la superficie.

Sin embargo, la comunidad de la tecnología lingüística también ha descubierto que algunas formas de profundidad lingüística sí pueden marcar alguna diferencia, en concreto la estructura sintáctica. Esto ha llevado a que se haya podido argumentar por analogía que lo mismo podría ocurrir con la DSP.

En concreto, [Chelba y Jelinek 1998] han demostrado, tras décadas de experiencia de lo contrario, que el uso de la estructura sintáctica en modelos estocásticos del lenguaje pueden lograr reducciones en la perplejidad y en la tasa de errores de palabras en comparación con el modelado de trigramas estándar. Este resultado puede tener consecuencias en aplicaciones como reconocimiento de voz, MT estadística y OCR (Optical Character Recognition). Otro ejemplo es [Kumar y Byrne 2002], que demostraron que una medida sintáctica de la distancia léxica puede servir para mejorar la eficacia de modelos de alineamiento de palabras estocásticos en MT. Además, Microsoft Word ha incorporado chequeo gramatical basado en análisis sintáctico y [Chiang 2005] ha aplicado con éxito análisis sintáctico síncrono libre de contexto en MT.

Aunque estos resultados representan una cierta respuesta a la afirmación de que los métodos lingüísticos tienen poco que aportar a las aplicaciones en general, incluir a la DSP entre esos métodos es relativamente aventurado. Primero, porque habría que decir cuáles son las propiedades del lenguaje concretas que tienen valor para aplicaciones concretas, y entonces extender la analogía a la DSP explícitamente. Segundo, porque estos logros de los métodos lingüísticos no son suficientemente grandes como para cambiar la visión de los desarrolladores prácticos, que puede estar muy influida por hechos como que prácticamente todo el modelado de lenguaje en las aplicaciones reales todavía utiliza modelos de n-gramas.

2.8.3 Argumento de las aplicaciones concretas

Por tanto, el argumento más fuerte que podemos encontrar proviene de las aplicaciones concretas que usen la DSP, ya que los argumentos de fe y por analogía presentan deficiencias graves.

Si la pregunta básica es en qué aplicaciones la DSP ayuda y por qué, en realidad se debe realizar un trabajo descriptivo sobre todas las aplicaciones en las que intervenga bien la DSP explícita u otro tipo de procesos que se le asemejen suficientemente y analizar su papel en cada caso. Si bien no existe ninguna aplicación concreta en la que la DSP haya sido un éxito inequívoco, el caso contrario sí se ha dado: hay una aplicación muy concreta que ha sido un fracaso notorio: la recuperación de la información monolingüe (monolingual IR). Además, siendo esta aplicación concreta una de las que más fuerza han dado al argumento de fe, el que los hechos reales contradigan tan flagrantemente la intuición, ha convertido a esta aplicación en el buque insignia del pesimismo sobre el potencial práctico de la DSP.

De hecho, la relación entre la DSP y la IR, se ha utilizado frecuentemente como espejo de la utilidad práctica de cualquier tipo de PLN en IR. Por ejemplo, [Voorhees 1999] utiliza resultados negativos de DSP en IR para ilustrar los problemas de las técnicas profundas de PLN en IR en general. Se suele contrastar el “gran éxito” de la IR

monolingüe como aplicación generadora de "una fuerte industria" alrededor de la manipulación de lenguaje natural en forma de texto no estructurado, con el hecho de que este fenómeno de alcance "mundial" haya tenido lugar sin el uso de la DSP explícita.

Con todo, la IR monolingüe es sólo una aplicación más entre muchas, por lo que en el resto de esta sección se aplicará la pregunta básica del argumento de las aplicaciones concretas a cada una de éstas en que sea pertinente.

La DSP tradicional en las aplicaciones

La DSP tradicional parte del hecho de que muchas palabras tienen más de un significado. En ella, los significados o sentidos de las palabras se enumeran *a priori*, y la tarea consiste en determinar el sentido apropiado para cada aparición de la palabra o palabras objetivo. No es casualidad que esta sea precisamente la misión de la mayoría de las tareas de los ejercicios Senseval.

La DSP en la IR tradicional

La DSP en su concepción tradicional, que es la utilizada en los certámenes Senseval, se ha utilizado en numerosas ocasiones como medio para mejorar las prestaciones de los sistemas de IR, con resultados infructuosos.

El paradigma predominante de la IR es el que se basa en la representación de los documentos como "bolsa de palabras" ("*bag-of-words*"). En este tipo de representación, un documento o un trozo de texto se caracteriza como una colección no ordenada de términos (palabras) y la relevancia de un documento para una consulta dada depende básicamente de los términos o palabras que tengan en común. Naturalmente, las palabras sin contenido semántico (palabras vacías) se eliminan como términos de la representación. Además, las palabras conjugadas, declinadas o relacionadas por procesos morfológicos se reducen a una única representación común a través de la segmentación (*stemming*), de forma que por ejemplo una consulta sobre "*connecting my camera*" y un documento que contenga "*connection of a digital camera*" utilizarían el mismo término ("*connect*") para "*connecting*" y "*connection*" de forma que se facilitaría la obtención de ese documento para esa consulta (aparte del término común "*camera*" sin necesidad de segmentación).

Ha habido muchos intentos de generalizar la idea básica de la segmentación. En uno de ellos [Voorhees 1999] se intentó ir más allá de las palabras con la misma raíz léxica, como en la segmentación, para considerar identificadores de sentido tras la desambiguación. Así por ejemplo, si una consulta o un documento contiene la palabra *bank*, su representación contendría su identificador de sentido (de entre varios sentidos

predeterminados posibles) después de su desambiguación a partir de su contexto. De esta forma, en teoría, se evitaría la coincidencia del sentido *bank-1* con el sentido *bank-2* lo cual en principio sería beneficioso.

Esta generalización se ha intentado llevar más lejos, a los conceptos: un concepto determinado ('un contenedor de dinero') podría ser representado por una palabra de la consulta ('banco') y por un sinónimo (mismo concepto) en un documento ('cajero'). Así el sistema los reduciría al mismo identificador y, en teoría, mejoraría la eficacia de todo el sistema de IR.

Sin embargo en [Voorhees 1999] se concluyó que la DSP, lejos de mejorar la eficacia de un sistema IR tradicional, incluso la perjudicaba en pequeña medida. Esto no hizo sino confirmar resultados anteriores [Sanderson 1994][Krovetz y Croft 1992].

Las razones de estos resultados han sido ampliamente estudiadas y se pueden reducir básicamente a tres. La primera razón se debería a la longitud corta de las consultas. Esto haría que fuera difícil desambiguar sus palabras por contexto, al ser éste muy reducido. La segunda razón estaría relacionada con el hecho de que la mayoría de las palabras polisémicas tienden a tener un sentido, el más frecuente, que domina la distribución de frecuencia del corpus objetivo. Esto hace que la palabra en sí (sin desambiguar) sea casi tan buen representante del término como la palabra desambiguada. Pero la razón más poderosa es la tercera: la mayoría de los modelos de IR tienden a realizar una desambiguación implícita de las consultas (formadas por varias palabras), lo que hace que la IR sin desambiguación explícita lo haga bien, o en otras palabras, la DSP explícita no aporte ningún incremento de efectividad.

Como ejemplo [Resnik 2006] considérese la consulta "*interest bank Fed*" sobre un conjunto de documentos que tratan sobre finanzas una parte y sobre ríos otra sección. Es mucho más probable que un documento sobre finanzas contenga más de un término de esta consulta que un documento sobre orillas de ríos. Por lo tanto el documento de finanzas logrará una marca superior en un sistema de IR basado en bolsa de palabras que el documento sobre orillas de ríos, aunque no haya habido ninguna forma de desambiguación explícita. [Sanderson 2000] llama a este hecho el "efecto de colocación de palabras en las consultas".

Por tanto la cuestión real no es si la DSP tradicional explícita puede ayudar en IR, sino si puede aportar valor sobre el efecto de desambiguación implícita u otras técnicas simples.

Bastantes investigadores con amplia experiencia en PLN para IR han encontrado que la DSP explícita no es útil, por las razones recién expuestas [Voorhees 1999][Sparck Jones 1999]. Un aspecto interesante puesto de manifiesto en [Sanderson 1994] es el del nivel de precisión necesario en la DSP explícita. Este aspecto también está relacionado con el número de palabras objetivo a desambiguar necesario. En ese trabajo se pone de

manifiesto que basta un 20-30% de error en la DSP para que el sistema de IR no logre beneficiarse de ella. Además sugiere que el nivel de precisión necesario empieza siendo a partir de un 90% (menos de 10% de error), algo "inalcanzable en un futuro previsible, por lo menos en un nivel de granularidad de sentidos fino y una tarea de 'todas las palabras' [como la de Senseval]" [Resnik 2006]. Como se expone en el capítulo 4, estos niveles de más de un 95% de precisión son los que se logran en ambigüedades fuertes o de nivel homográfico, no necesariamente de sólo dos sentidos, y además las palabras objetivo que presentan ese tipo de ambigüedad son sólo una parte de todas las palabras del corpus, y éstas son las que realizaría una tarea de 'todas las palabras' al estilo de Senseval. [Mihalcea y Moldovan 2000] obtuvieron mejoras centrándose sólo en aquellas palabras que pueden desambiguarse con gran precisión, utilizando un subconjunto de una colección estándar de IR, utilizando una combinación de indexación basada en synsets de WordNet y basada en palabras. [Kim et al. 2004] centran la DSP donde ésta puede ser precisa y mitigan los efectos de la imprecisión aplicando etiquetas de granularidad gruesa a sustantivos logrando mejorar la eficacia por encima de líneas de base realistas en IR. [Liu et al. 2004] emplean desambiguación de alta precisión a términos de las consultas para realizar la expansión de éstas.

La DSP en aplicaciones relacionadas con la IR

Como se ha visto en la sección anterior, la IR tradicional ha sido un terreno difícil para la DSP tradicional. Las razones son, por un lado, el hecho de que la IR tradicional ya realiza desambiguación implícita por otros medios y, por otro, la formalización tradicional de la tarea de la IR en términos de relevancia a nivel de los documentos, lo que permite a los sistemas basados en 'bolsa de palabras' la identificación de documentos relevantes, sin una comprensión más profunda de la información solicitada por la consulta.

Sin embargo, puede que estas causas tengan menor impacto en dos aplicaciones emergentes relacionadas con la recuperación de la información: la Recuperación de la Información entre Idiomas (Cross Language Information Retrieval, CLIR) y la Respuesta a Preguntas (Question Answering, QA). En la CLIR, esto de debe a que la ambigüedad de la traducción de la consulta a otros idiomas complica la desambiguación implícita y en la QA, a que la propia naturaleza de la tarea tiende a no beneficiar los modelos de IR clásicos de 'bolsa de palabras'. Además, la clasificación de las consultas de los motores de búsqueda en la web, que necesitan su desambiguación, con el propósito de mejorar el direccionamiento de los anuncios publicitarios, está atrayendo la atención a otra aplicación relacionada, la Clasificación de Textos (Document Classification).

CLIR (Cross-language IR)

La CLIR es una aplicación de creciente interés, debido a la naturaleza cada vez más global de la búsqueda de información en la web, el comercio global, y la demanda de los servicios de inteligencia. Como se ha señalado en la sección anterior, la DSP explícita puede tener mayor aplicación en la CLIR que en la IR monolingüe debido a la interacción entre la ambigüedad de los sentidos y la ambigüedad de la traducción.

En una situación de CLIR el usuario presenta una consulta como las habituales, pero en este caso puede haber documentos relevantes escritos en un idioma diferente del utilizado en la propia consulta. Supongamos que el lenguaje en el que está escrito la consulta se llama L_Q, y el lenguaje en el que hay parte de los documentos relevantes y que ahora nos ocupa se llama L_D. Una estrategia normal, denominada *traducción de consultas*, consiste en traducir la consulta de L_Q a L_D, de forma que se reduce el problema a IR monolingüe en L_D. El problema ahora consiste en que además de la ambigüedad de sentidos dentro de L_D, ahora se añade la ambigüedad de sentidos introducida por la traducción de L_Q a L_D. Es decir, cada término de la consulta en L_Q se traduce a varias palabras en L_D, que representan sentidos diferentes del término en L_Q. En la mayoría de los sistemas de CLIR se incluyen todas las traducciones posibles, si bien con diferentes pesos. El ruido introducido por muchas de las palabras de la traducción interfiere con la desambiguación implícita propia de la IR monolingüe, dando lugar a una degradación de la eficacia de la consulta.

Para verlo con un ejemplo [Resnik 2006], considérese un caso en que la palabra x en una consulta en el idioma inglés se puede traducir a las palabras x_1, x_2 y x_3 en el idioma chino, y la palabra y en inglés se puede traducir a las palabras y_1, y_2, y_3 y y_4 en chino. Suponiendo que la traducción correcta de las dos palabras es la que produce como resultado las palabras x_1 y y_1, los documentos en chino que contengan ambas palabras conseguirán una puntuación más alta que los que tengan sólo una de las dos, como consecuencia del efecto de desambiguación implícita. Pero los documentos que tengan simultáneamente las palabras x_1 y y_2 tendrán exactamente las mismas posibilidades de puntuación aún cuando y_2 no es la traducción correcta de y en este contexto, y lo mismo ocurre con un total de once combinaciones erróneas. Naturalmente, estos documentos que alcanzan puntuaciones altas son irrelevantes desde el punto de vista de esta consulta concreta.

Basándose en este hecho [Oard y Dorr 1996] observaron que la polisemia es un "factor de limitación clave" con mayor intensidad en recuperación multilingüe que en recuperación monolingüe. En concreto afirman que "la polisemia puede reducirse

utilizando información sintáctica y semántica, de entre las que el tipo más sencillo lo constituye la 'formación de frases' [phrase formation]".

Como ejemplo [Resnik 2006] de esta última técnica, considérese la palabra *interest*, que es ambigua en inglés entre un sentido financiero y un sentido de tiempo libre. En chino, cada sentido se expresa con una palabra diferente: *li xi* y *xing qu*, respectivamente. Si una consulta en inglés contuviera las palabras *interest* y *rate*, y se tradujera al chino, si las dos palabras se tratan independientemente, la segunda traducción de *interest* podría perjudicar a la precisión de la consulta. Por ejemplo, se podría devolver como relevante un documento que hablara sobre la tasa (*rate*) elevada de crecimiento del interés (*interest*) de los chinos por el acceso a internet. En cambio, si las dos palabras no se tratan independientemente, porque un análisis previo de la consulta identifica *interest rate* como una frase del idioma inglés y se traduce toda la frase en su conjunto al chino, claramente esta traducción de toda la frase tendría menos términos que todas las combinaciones de las dos traducciones independientes, de forma que en la CLIR, "la DSP, que, como la 'formación de frases', ha demostrado poca utilidad en un contexto monolingüe podría ser un camino productivo para investigación subsiguiente" [Oard y Dorr 1996], citado en [Resnik 2006]. De hecho, varios experimentos descritos en la literatura sobre la CLIR confirman el valor de la selección de la traducción [Ballesteros y Croft 1997][Gao et al. 2001][Qu et al. 2002]. Aunque la mayoría de estos resultados no utilizan DSP explícita dos experimentos recientes sí lo hacen: [Clough y Stevenson 2004] y [Vossen et al. 2006]. Los dos utilizan EuroWordNet y el último utiliza DSP explícita especializada en dominio (ver sección 2.6).

Respuesta a preguntas (Question Answering, QA)

Aunque la QA es una aplicación de las más antiguas de PLN, en la actualidad su objetivo es encontrar respuestas a preguntas en texto de lenguaje natural, en general sin restricciones de dominio. A diferencia de la IR, que ante la consulta "*patente de la bombilla eléctrica de Edison*" devuelve una serie de documentos completos sobre la bombilla eléctrica y Edison, en la QA se le hacen al sistema preguntas concretas del tipo "*¿cuándo patentó Edison la bombilla eléctrica?*" y éste debe responder con una respuesta también concreta.

La investigación sobre QA ha aumentado considerablemente y en TREC-8 [Voorhees y Tice 2000][7] se llevaron a cabo las primeras comparaciones estándar entre sistemas. Según [Voorhees 1999] y [Sparck Jones 1999] la aplicación de análisis lingüístico, ya sea mediante técnicas de PLN o mediante otras técnicas más simples, podría ser interesante en esta tipo de tareas. Por ejemplo [Resnik 2006], un análisis sintáctico de la frase del párrafo anterior, que diera como resultado algo similar a lo siguiente:

[7] TREC (Text REtrieval Conference) es una competición anual donde se evalúan sistemas de Recuperación de la Información. Está organizada por el NIST (Nacional Institute of Standards and Technology) en Estados Unidos.

SUJETO(PATENTE, EDISON), PREDICADO(PATENTE, BOMBILLA), MODIFICADOR(BOMBILLA, ELÉCTRICA); podría no ser necesario en una aplicación de tipo IR, pero sí en una aplicación de tipo QA, donde por ejemplo hubiera que distinguir entre las respuestas *Edison patentó la bombilla eléctrica en 1879* y *La patente anterior de la bombilla eléctrica de Joseph Swan de 1878 supuso algunos problemas para Edison*.

Esta naturaleza de la tarea de la QA hace que el llamado "problema de la paráfrasis" [Oard y Dorr 1996] [Woods 1995] sea más acuciante que en la IR. Por ejemplo, la frase *La patente de Edison de 1879 de la bombilla eléctrica* tendría las mismas consecuencias en un sistema de 'bolsa de palabras' de IR, mientras que en QA la relación SUJETO(PATENTE, EDISON) de la pregunta inicial se perdería, con lo cual sería necesaria una etapa adicional de procesamiento más próximo a la semántica, es decir, más profundo o alejado de la superficie.

Es en este paso adicional hacia la semántica donde se puede creer que las distinciones de sentido explícitas son útiles para la QA. Un sistema muy conocido a este respecto es el LCC-SMU [Pasca y Harabagiu 2001a], que utiliza mayor cantidad de técnicas de PLN que la mayoría de sistemas y ha superado a muchos de ellos en evaluaciones estándar de QA. Este sistema utiliza representaciones sintácticas a nivel de dependencias para el análisis de las preguntas y las posibles respuestas, y además evalúa la calidad de éstas utilizando conocimiento léxico, morfológico y semántico a partir de WordNet y otras bases de datos léxico-semánticas. El sistema utiliza a veces distinciones de sentido implícitas, pero también usa DSP explícita [Pasca y Harabagiu 2001b] a partir de WordNet y la Web [Mihalcea y Moldovan 1999].

Otro desarrollo similar es el "Recognising Textual Entailment Challenge" [Dagan, Glickman y Magnini 2004], un ejercicio de evaluación sobre la paráfrasis. En este ejercicio se les suministra a los sistemas encabezamientos del tipo *Yahoo took over search company Overture Services Inc last year* junto con otros como *Yahoo bought Overture*, y éstos deben decidir si el primero implica el segundo, como ocurre en este caso. En la reciente Senseval-3 Dagan propuso una variante de este ejercicio destinada a la evaluación de sentidos de las palabras. Consiste en especificar en cada encabezamiento como los anteriores una palabra o frase léxica y determinar si los significados de éstas están relacionados por implicación. En el ejemplo anterior, *took over* en el contexto dado implica *bought*. Aunque en estas tareas puede que no sean necesarias representaciones explícitas de sentidos o el uso de DSP explícita, sí que podrían ser útiles, en cuyo caso podrían formar un puente muy útil entre la evaluación de la DSP explícita y la evaluación a nivel de aplicaciones como la QA, donde es muy importante la identificación de relaciones de implicación. Por analogía, esta tarea de implicación textual también podría servir de puente entre la DSP explícita y la evaluación de la MT, si consideramos la traducción correcta como una implicación mutua entre textos escritos en idiomas diferentes.

Clasificación de documentos

La mayoría de la investigación sobre la clasificación de textos en categorías predefinidas se basa en el mismo paradigma de 'bolsa de palabras' que la IR estándar. Además, algunos intentos de mejorar la representación de los textos con sentidos de palabras no han dado como resultado mejoras en la eficacia de los sistemas [Kehagias et al. 2003] [Moschitti y Basili 2004], por las mismas razones básicas que la IR monolingüe (véase sección 2.8.3).

Sin embargo, en varios casos sí se han obtenido beneficios. [Vossen et al. 2006] obtienen mejoras en clasificación de documentos utilizando una técnica de DSP basada en los dominios específicos (véase sección 2.6). [Bloehdorn y Hotho 2004] utilizan rasgos para representar los documentos obteniendo muy buenos resultados en las colecciones Reuters y Ohsumed (utilizadas en IR y extracción de información). Algunas mejoras se deben a la detección de expresiones de varias palabras y a la unión de sinónimos y otras a la generalización de conceptos. [Hotho et al. 2003] también registraron mejoras de resultados en clasificación de documentos.

Una aplicación particular de la clasificación de textos en los motores de búsqueda en la Web es la clasificación de las consultas de los usuarios con vistas a mejorar los anuncios publicitarios. Un ejemplo de Chris Brew, en comunicación personal a Philip Resnik [Resnik 2006], es el siguiente: si un usuario hace la consulta "*duck soup washigton d.c.*", qué tipo de anuncios le convendrán más, ¿a listados de restaurantes o a anuncios de un festival de cine de los hermanos Marx? De esta forma determinar las intenciones del usuario sería muy parecido a desambiguar el significado de sus consultas. [Li y Zheng 2005] registran una competición con este fin.

La DSP en la MT tradicional

Los sistemas de MT tradicionales se clasifican convencionalmente en sistemas interlingüísticos y sistemas de transferencia. En los primeros se utiliza una representación intermedia del "significado" del concepto expresado en el lenguaje origen, mientras que en los segundos no se utiliza este tipo de representación entre el lenguaje fuente y el lenguaje destino.

En el modelo de MT interlingüístico de [Dorr 1993] el léxico contiene varias entradas para el mismo verbo y con ayuda del contexto sintáctico se restringen el número de entradas viables. Por ejemplo [Resnik 2006], la oración fuente *I broke the news to Mary* sólo permite el sentido de comunicación del verbo *break*, en contraste con la frase *I broke the glass to Mary*. El resultado de este análisis es una representación semántica hecha a partir de componentes semánticos de las palabras de la oración. Los elementos constructores de esta representación suelen ser elementos semánticos interlingüísticos del tipo CAUSA, SER, etc. Finalmente hay una etapa de generación casi inversa en el cual

los elementos léxicos del lenguaje destino se eligen en función de su correspondencia con la semántica de la oración que se traduce.

Los sistemas de transferencia tienen la posibilidad de, al poder relacionar directamente palabras del lenguaje fuente con palabras del lenguaje destino, tratar las palabras del lenguaje destino como inventario de sentidos de las palabras del lenguaje fuente.

La evaluación de la eficacia de la DSP explícita en la MT tradicional es difícil debido a una serie de razones. Una de ellas es que la ambigüedad de sentidos es una entre muchos tipos de ambigüedad con los que tienen que tratar estos sistemas. Otra sería la relativa tardanza en la realización de evaluaciones estándar de estos sistemas. La DSP explícita no jugó ningún papel visible en ninguno de los sistemas participantes en las evaluaciones de la ARPA en los primeros 1990 [White y O'Conell 1994]. En certámenes más recientes la mayoría de los sistemas participantes eran sistemas de MT estadísticos (véase la próxima sección), o bien sistemas comerciales cuyos detalles permanecen como confidenciales.

Sin embargo, el sistema Systran, que es un sistema de transferencia clásico, es un ejemplo de cómo las técnicas de DSP explícita tradicionales se pueden utilizar con los términos del lenguaje destino como identificadores de sentidos. En comunicación personal a Philip Resnik [Resnik 2006] Laurie Gerber indica que, para algunos pares de lenguajes, Systran selecciona el significado en el lenguaje destino haciendo primero desambiguación monolingüe en el lenguaje origen, de forma parecida a los sistemas interlingüísticos. De todas formas también se usan "reglas léxicas específicas de las palabras" que usan el contexto para asignar el significado en el lenguaje destino, lo cual hace muy difícil aislar completamente el trabajo de la DSP en el proceso de traducción.

La ambigüedad de sentidos en la MT estadística

Aunque en la MT estadística la DSP explícita no juega ningún papel, la elección léxica en esos sistemas implica realizar muchas de las tareas de la desambiguación de sentidos, por lo que es muy interesante observar cómo las llevan a cabo.

La MT estadística tiene sus orígenes en el trabajo de [Brown et al. 1990] en IBM. En este tipo de MT estadística, la desambiguación de significados de las palabras se hace en dos pasos. En el primero, llamado la probabilidad "Modelo 1", se calcula la probabilidad condicional entre palabras $\Pr(f \mid e)$, donde f y e son palabras del lenguaje fuente y destino, respectivamente. El hecho de que las palabras estén intercambiadas en la fórmula, ya que estamos traduciendo del lenguaje fuente al lenguaje destino, se debe a la aplicación de la Regla de Bayes. Nótese que en el cálculo de esta probabilidad no se utiliza ninguna información del contexto de las palabras. El segundo paso utiliza un modelo basado en n-gramas, en el que se calcula la probabilidad de una palabra basada en las n-1 palabras precedentes, en el idioma destino, donde n suele ser 2 ó 3. Por

ejemplo, considérese la palabra francesa *essence*, que se puede traducir al inglés como *essence* (en español *esencia* como en esencial) o como *gas(oline)* (en español *gasolina*). La elección de la traducción depende, además de las probabilidades no contextuales del "Modelo 1" del contexto en el lenguaje destino. Si el modelo estadístico de este contexto considera que las dos palabras precedentes más probables son *out of*, lo más seguro es que el sentido asignado sea el segundo.

En los últimos años el tipo de MT estadístico dominante ha pasado a ser el modelo de MT basado en frases (*phrased-based*) [Och 2002][Koehn et al. 2003]. Este enfoque es parecido al llamado "example-based machine translation" (EBMT) [Nagao 1984] [Brown 1996], en el que la oración destino se construye recomponiendo la oración fuente con "trozos" ("*chunks*") almacenados en una 'memoria de traducción'. En la versión estadística se utiliza un modelo probabilístico para hacer corresponder "frases" en los idiomas fuente y destino. Estas probabilidades se basan en modelos aprendidos de datos de entrenamiento [Koehn et al. 2003][Koehn 2004]. En este contexto la palabra "frases" significa cualquier subconjunto contiguo de palabras en la oración fuente o destino, sin ninguna connotación lingüística.

El paso de las palabras, como en el modelo IBM, a las frases se debe en parte a la observación de que el contexto local en el lenguaje fuente puede aportar mucha información al proceso de selección léxica, algo que el sistema de IBM no hacía. Otra razón apunta a la traducción no literal de frases, como por ejemplo [Och 2002] la traducción de la frase *das wird schwierig* del alemán al inglés: lo correcto sería *that will not be easy*, pero un sistema como Systran lo hace literalmente a *that becomes difficult*, que es claramente incorrecto. Una última razón sería la traducción de palabras funcionales como preposiciones, artículos y partículas, en las que "la traducción correcta depende sobre todo del contexto en el idioma fuente". Este es un caso en el que la correspondencia entre frases en los dos idiomas se beneficia del contexto local, de forma consistente con la hipótesis 'un sentido por colocación' de [Yarowsky 1993] (véase sección 2.2.4).

La relación entre la DSP explícita y la MT estadística se ha investigado en algunos trabajos recientes. [Carpuat y Wu 2005a] intentaron integrar un módulo de DSP explícita con inventario de sentidos en un sistema de MT estadístico tipo IBM, sin éxito. En [Carpuat y Wu 2005b] comparan empíricamente la desambiguación estadística de un sistema MT con la desambiguación de la DSP tradicional en una tarea de desambiguación de Senseval y registran ventajas de eficacia para la DSP tradicional, lo que según ellos significa que la MT estadística debería beneficiarse de "las predicciones hechas por los modelos de DSP". [Jiménez et al. 2005] demuestran que las correspondencias de traducciones extraídas de un inventario de sentidos multilingüe son beneficiosas para la MT estadística. [Cabezas y Resnik 2005] y [Vickrey et al. 2005] han desarrollado independientemente sistemas de MT en los que intentan disociar completamente las técnicas de DSP de inventarios de sentidos explícitos. En su lugar los "sentidos" son palabras en el idioma destino y se entrenan clasificadores

supervisados con textos alineados paralelos que suministran parejas palabra/"sentidos" de entrenamiento. Esta idea será retomada más adelante en esta misma sección.

Otras aplicaciones emergentes

Además de la CLIR y la QA revisadas en esta misma sección, otras dos aplicaciones emergentes que necesitan identificar categorías semánticas son la extracción y minería de información a partir de texto y la adquisición de conocimiento semántico.

El objetivo de la extracción de información es coger un texto en lenguaje natural como entrada y rellenar un "cuestionario" ("*template*") para describir una serie de relaciones básicas que se cumplen en el texto para un dominio particular determinado. En la minería de datos textuales, el objetivo es descubrir modelos de relaciones que existen en grandes cantidades de texto.

[Resnik 2006] pone como ejemplo uno extraído de la bioinformática. Hay un volumen enorme de literatura sobre biología molecular que trata sobre genes, proteína y enzimas, entre muchas otras cosas, y se están creando bases de datos gigantescas que están estructurando toda esa información. El problema está en que hay un vacío entre el texto libre de los artículos científicos y la información estructurada de las bases de datos. En 2002 la competición Copa KDD [Yeh et al. 2002] propuso como tarea analizar artículos científicos y extraer información sobre el genoma de la *Drosophila*, que consistía en identificar todos los genes que aparecían en los artículos y decidir si el artículo registraba una relación entre el gen y un producto del gen, que podía ser una proteína y/o ARN. El problema se agrava debido a que en la literatura de biología molecular el mismo término puede referirse indistintamente y según el contexto a un gen, una proteína o una molécula de ARN. De ahí que la desambiguación de esos términos en contexto, es decir, lo que hace la DSP [Hatzivassiloglou et al. 2001] puede ser muy útil en esta tarea.

[Weeber et al. 2001] tratan sobre la desambiguación en PLN de lenguaje médico, y mencionan aplicaciones como apoyo a las decisiones médicas, indexación y descubrimiento a partir de la literatura.

Estos problemas van más allá del dominio de la medicina y afectan tanto a aplicaciones tradicionales como nuevas. Por ejemplo, PLN puede referirse a natural language processing o a neurolonguistic programming; MG puede referirse a miligramo o a magnesio; *New York* puede referirse a la ciudad de Nueva York o al estado; *George Bush* puede referirse a Goorge W. Bush o a su padre. Nótese que todos estos ejemplos pueden reducirse a ambigüedades a nivel homográfico (de dos sentidos) como las que resuelve el algoritmo original de Yarowsky de 1995 [Yarowsky 1995] (véase sección 5.7) en corpus homogéneos sin fluctuaciones de dominio y a las que se dirige el

algoritmo presentado en este trabajo para corpus generales con cualquier variación de dominio.

Otro tipo de aplicación emergente donde la DSP explícita puede tener gran valor lo constituyen aplicaciones cuyo objetivo es colocar términos o frases en una estructura de conocimiento explícita. Entre ellas figura el desarrollo de interfaces de usuario mejoradas. En [Hearst 2000] se aboga por interfaces de búsqueda más orientadas al objetivo que integren metadatos, como información de tópicos, a la vista del usuario. [Yee et al. 2003][Stoica y Hearst 2004] ilustran esta posibilidad mediante una interfaz que buscara una colección de imágenes de bellas artes, y que creara categorías para la colección automáticamente a partir de los encabezamientos de las imágenes utilizando WordNet; en este ejemplo, ante términos ambiguos se forzaba a ignorar el término o a elegir el primer sentido (el más frecuente) de WordNet. El proyecto MALACH [Gustman et al. 2002] también usa una interfaz hombre-máquina: pretende dar acceso a un archivo muy grande de historias orales sobre el Holocausto; el archivo contiene 116.000 horas de grabaciones orales en 32 idiomas obtenidas en 52.000 entrevistas. Aparte de la transcripción de todo este material, los autores deben hacer corresponder trozos de las grabaciones transcritas con categorías de un tesauro formado por tópicos, topónimos, etc. Sawyer et al. (en preparación) proponen que el etiquetado semántico simple y poco sofisticado puede ser útil para organizar y presentar a los usuarios los contenidos de los documentos sobre un nuevo dominio de problema en la fase de ingeniería de requisitos.

Otro ámbito de trabajo importante como la Web Semántica, donde se pretende dar a toda la información que hay en la Web "un significado bien definido, permitiendo a los ordenadores y a las personas mejorar el trabajo cooperativo" [Berners-Lee et al. 2001]. En este proyecto se está desarrollando un gran esfuerzo para la definición de estructuras de conocimiento ontológico y también para aplicar etiquetado semántico automático a la Web a muy gran escala [Dill et al. 2003]. En [Wilcock et al. 2004] y [Oltramari et al. 2004] se trata sobre las conexiones entre la tecnología lingüística y la Web Semántica.

La construcción de ontologías como ayuda a las aplicaciones informáticas tiene una larga tradición anterior a la Web Semántica. Algunas veces la DSP es una parte explícita de este proceso: [Dorr y Jones 2000] emplean DSP para mejorar la creación de léxicos semánticos a gran escala; [Rigau et al. 2002] describen un proceso de autoarranque (*bootstrapping*) que incluye DSP y adquisición de conocimiento en un entorno multilingüe; [Basili et al. 2004] tratan sobre la creación de fuentes de conocimiento multilingüe en un contexto de QA basada en ontologías.

Como ejemplo final tenemos la lexicografía: [Kilgarriff y Tugwell 2001] presentan una aplicación en la que la DSP explícita sirve de ayuda a los lexicógrafos, en sentido inverso al habitual, y [Löfberg et al. 2004] tratan de etiquetado semántico de granularidad de sentidos gruesa para búsquedas en un diccionario sensibles al contexto.

DSP no tradicional

En esta sección se revisan aplicaciones en las que la distinción de sentidos no es una lista de éstos enumerada explícitamente.

Representaciones amplias del lenguaje

En [Kilgarriff 1997] se hace notar que hay una cantidad considerable de literatura teórica, desde la lingüística cognitiva [Lakoff y Johnson 1980] hasta la teoría generativa del léxico [Pustejovsky 1995], que defiende representaciones del significado de las palabras más ricas que la simple enumeración de sentidos. Los argumentos que se esgrimen se basan fundamentalmente en dos observaciones. La primera es que los significados se pueden extender mucho más libremente de lo que expresan las listas de sentidos. Por ejemplo el verbo *to eat* en inglés se puede interpretar a la noción general de 'destruir por consumo' al entender la frase *The computer ate my document*. La segunda es que el léxico contiene una gran variedad de relaciones sistemáticas entre sentidos de palabras. Por ejemplo [Resnik 2006], para *to eat* tenemos sentidos de animal o comida, como *chicken, lamb, goose*, sentidos de continentes o cantidades contenidas, como *cup, bowl, bottle*, y muchas más.
En [Wilks 1997] [Wilks 1998] se recuerda que estas ideas tiene una larga tradición en la literatura de PLN y se han desarrollado en teorías como la semántica preferencial [Wilks y Fass 1992]. Sin embargo, más allá del evidente interés teórico que tienen, su aplicación práctica en tecnologías del lenguaje ha sido escasa, en parte debido a su intensidad cognitiva y a su dificultad de formalización e implementación a una escala manejable.

Modelos de uso

Una organización alternativa del léxico, pero más práctica, sí ha tenido bastante influencia en las aplicaciones prácticas de tecnología lingüística. Se trata de los modelos de uso (patterns of usage en inglés), que se basan en la organización de las palabras y/o los sentidos según sus patrones de distribución en los corpus concretos. Las ideas sobre ello se remontan a [Firth 1957] [Sparck Jones 1964] [Harris 1968]. El hecho de que estos patrones de distribución reflejen una semántica inherente es relativamente poco importante desde el punto de vista de las aplicaciones prácticas, donde lo realmente interesante es que se mejore la eficacia.

Un ejemplo de los modelos de uso es la llamada Latent Semantic Indexing (LSI) [Deerwester et al. 1990]. En esta técnica, que se usa en IR, el vocabulario de una colección de documentos se representa mediante una matriz V en la que un elemento v_{ij} es una función ponderada de la frecuencia con que una palabra w_i aparece en un

documento d_j. De esta forma, aquellas palabras que tienden a aparecer en muchos de los mismos documentos están representadas por filas de la matriz similares. La dimensión de la matriz se reduce mediante una descomposición de valor singular (singular value decomposition, SVD) y las palabras con modelos de uso similares se agrupan. Así, si las palabras *dentista* y *estomatólogo* reciben representaciones similares, un documento que incluya 'estomatólogo' puede recibir una puntuación alta como respuesta a una consulta que contenga 'dentista', aunque el documento y la consulta no tengan palabras en común y la eficacia en términos de precisión y exhaustividad del sistema de IR aumente.

El hecho de que dentista y estomatólogo sean palabras con significados parecidos no quiere decir que esta técnica sirva para descubrir relaciones de significado entre las palabras a partir de patrones de uso distribucionales. Siguiendo con este ejemplo, es probable que la palabras *diente* y *cavidad* también aparezcan relacionadas. La relación entre dentista y cavidad no tiene por qué ser semántica o siquiera conceptual, y no pasa de ser una relación de ocurrencia en documentos sobre temas similares, un tipo de relación que se podría definir de alguna manera si fuera necesario.

Sin embargo, a efectos de la IR, las relaciones conceptuales que hubiera no son tan importantes como el efecto de las representaciones en el incremento de la eficacia. En [Schütze y Pedersen 1995] [Schütze 1998] [Jing y Tzoukermann 1999] pueden verse los resultados de evaluaciones formales en este sentido. Además las representaciones distribucionales también se usan en temas muy relacionados como la recuperación interactiva basada en agrupamiento de documentos [Hearst y Pedersen 1996] y en organizaciones distribucionales de los sentidos [Véronis 2004]. En [Mihalcea, Tarau y Figa 2004] [Erkan y Radev 2004] se introducen técnicas inspiradas en el algoritmo PageRank de Google que utilizan una representación distribucional basada en grafos en lugar de vectores.

Aunque como se ha dicho un poco más arriba no hay una relación semántica o conceptual evidente entre las palabras relacionadas por esta representación distribucional, se podría afirmar que de hecho hay algún tipo de relación, aunque no sea obvia. [Kilgarriff 1997] propone un modelo de sentido de las palabras en el que las distinciones de sentidos sean "grupos de usos de palabras" que existen "en relación a conjuntos de intereses" determinados por una aplicación PLN o un corpus objetivo. Incluso va más allá cuando afirma que "un conjunto de sentidos independiente de cualquier tarea no es un concepto coherente". [Krovetz 2002] abunda en ello cuando dice que "las diferentes aplicaciones de lenguaje natural necesitan distinciones de sentidos diferentes". [Schütze 1998] implementa estas ideas en una aplicación de IR con sus sistemas de discriminación de grupos contextuales mientras que [Agirre y Edmonds 2006] señalan la DSP dependiente de la tarea como una de las cuestiones abiertas más importantes en DSP.

Relaciones entre idiomas

La disposición de diccionarios bilingües o de traducciones paralelas (corpus alineados) suministra la posibilidad de tener un significado "oculto" que se muestra en dos representaciones externas, una en cada idioma [Resnik 2004]. En cambio, en un corpus monolingüe sólo son observables las palabras, no los sentidos.

Un tipo de aplicación de este paralelismo lo constituye la LSI interlingüística (*cross-language Latent Semantic Indexing*, CL-LSI) [Littman et al. 1998]. Esta técnica es una aplicación inmediata de la LSI a corpus paralelos produciendo un espacio "semántico" interlingüístico, en el que la proximidad implica similitud, pero *independientemente* de si las dos palabras pertenecen al mismo lenguaje o no. De esta forma se tiene un espacio de regiones de similitud semántica *independiente de los idiomas*, algo similar a sentidos de las palabras interlingüísticos. La técnica se puede aplicar a CLIR de forma análoga a la LSI monolingüe.

Otra forma de explotar las relaciones interlingüísticas es la caracterización de estructuras de conocimiento discretas usando las correspondencias entre los idiomas. [Dyvik 2002] utiliza un diccionario bilingüe para obtener conjuntos de palabras del tipo de los synsets de WordNet y relaciones entre ellos. [Resnik y Yarowsky 2000] [Chugur, Gonzalo y Verdejo 2002] demuestran empíricamente que cuanto más diferentes sean dos sentidos monolingües mayor es la probabilidad de que se traduzcan de diferente forma a otros idiomas. Además este hecho da lugar a una medida empírica de distancia entre los sentidos a partir de la que se obtienen estructuras muy parecidas a las distinciones lexicográficas de los diccionarios monolingües. [Ide 2000] ha aplicado esta idea a gran escala con un corpus paralelo multilingüe y [Ide et al. 2002] [Tufis et al. 2004] lo han aplicado al etiquetado de sentidos. [Diab 2003] ha desarrollado un algoritmo no supervisado que usa corpus paralelos para conseguir DSP monolingüe en los dos idiomas, etiquetando con un inventario de sentidos en uno de los dos idiomas. [Bhattacharya et al. 2004] ha formalizado de forma probabilística y ha extendido esta idea. [Ng et al. 2003] utilizan corpus bilingües para obtener ejemplos de entrenamiento para un clasificador supervisado.

Una última línea de investigación, que data de los primeros 90, se basa en utilizar las palabras del segundo idioma como etiquetas de sentidos del primer idioma. [Brown et al. 1991] utilizaron textos alineados para entrenar el etiquetado de palabras del primer idioma con "sentidos" del segundo idioma. [Gale et al. 1992] propusieron utilizar palabras alineadas como etiquetas de sentido en DSP. [Dagan y Itai 1994] introdujeron técnicas que desarrollaban esta idea utilizando sólo corpus monolingües en los dos idiomas. [Li y Li 2004] han extendido el algoritmo de Yarowsky [Yarowsky 1995] a una situación bilingüe para resolver el problema de desambiguación de palabras traducidas utilizando palabras del idioma chino como sentidos del idioma inglés.

Esta línea de investigación (véase también la sección 2.8.3) se probó en Senseval-2 y formó la base de una tarea de Senseval-3. El hecho de etiquetar unas palabras con otras en otro idioma tiene dos ventajas claras. Relaja el problema del CBAC en los algoritmos supervisados (en el sentido estricto), ya que los corpus paralelos son mucho más fáciles de obtener que los corpus etiquetados manualmente monolingües. Como segunda ventaja, sirve de conexión entre la DSP y algunas aplicaciones finales concretas, como la MT y la CLIR, donde ha demostrado su utilidad.

Capítulo 3. El Cuello de Botella de la Adquisición de Conocimiento (CBAC)

Como se ha comentado en el capítulo 1 existen fundamentalmente tres clases generales de sistemas de DSP: los basados en conocimiento, los supervisados basados en corpus y los no supervisados basados en corpus. En ese capítulo también se señaló que de los tres tipos, los más eficaces en términos de precisión son los supervisados; pero también que estos métodos utilizan corpus etiquetados manualmente relativamente grandes como medio de entrenamiento, lo cual constituye su principal desventaja. Este problema se denomina habitualmente en la literatura anglosajona *knowledge acquisition bottleneck*, y nos referiremos a él como *cuello de botella de la adquisición de conocimiento* o abreviadamente CBAC.

El *cuello de botella de la adquisición de conocimiento, CBAC* (Knowledge Acquisition Bottleneck) es un problema que afecta de forma importante a los métodos de DSP supervisados y, en menor medida, a los métodos basados en conocimiento.

En general, ambos tipos de métodos necesitan recursos de información "externa" distintos al propio corpus objetivo sin etiquetar. En el caso de los métodos supervisados, esta información "externa" consiste en un corpus etiquetado de entrenamiento, y en el caso de los métodos basados en conocimiento se trata de diccionarios, tesauros, o bases de conocimiento léxico generales.

En el primer caso, el corpus etiquetado de entrenamiento tiene que guardar cierta relación con el corpus objetivo: por lo menos ambos corpus deben estar escritos en el mismo idioma; el dominio debe también ser el mismo, o al menos no debe divergir excesivamente; y la extensión también debe ser en cierta medida comparable o guardar ciertas proporciones.

En el segundo caso, las bases de conocimiento externo suelen ser independientes del corpus objetivo y en general son comunes a muchas tareas que sólo tienen en común el lenguaje o idioma objetivo.

Esto quiere decir que la principal barrera que supone el CBAC para los métodos basados en conocimiento es la del idioma o, en otras palabras, la solución sería la construcción de diccionarios, tesauros y bases léxicas en muchos idiomas. Esto no quiere decir que sea la única: una base de datos de este tipo puede estar especializada en un dominio exclusivamente.

El problema del CBAC es más grave en los métodos supervisados porque los corpus no sólo están especializados en un idioma, lo cual ya es importante, sino que forzosamente también lo están en un dominio, e incluso en un tamaño. Esto quiere decir que se debe dedicar gran esfuerzo físico y económico en generarlos y mantenerlos, dado que es una actividad en principio manual y además llevada a cabo preferiblemente por especialistas.

Dada esta diferencia en la naturaleza del problema en los dos tipos de métodos, y considerando además que en DSP los métodos supervisados son más eficaces en cuanto a precisión, dedicaremos el resto de este capítulo al tratamiento del CBAC en los sistemas supervisados.

3.1 El CBAC en los sistemas supervisados

El problema del CBAC es tan fuerte en los sistemas supervisados que en la mayoría de los corpus disponibles es incluso difícil encontrar el número suficiente de apariciones de los sentidos de una palabra (objetivo). Esto es así incluso en el idioma inglés, que es con diferencia el más estudiado.

Para ayudar a resolver el problema se están desarrollando una serie de líneas de investigación que se describen a continuación [Gonzalo and Verdejo 2006].

3.1.1 Adquisición automática de ejemplos de entrenamiento

La adquisición automática de ejemplos consiste en utilizar una base léxica externa (análoga a las utilizadas por los sistemas de DSP basados en conocimiento) como por ejemplo WordNet, o bien un corpus etiquetado, para obtener ejemplos nuevos de un corpus no etiquetado muy grande (por ejemplo Internet).

Adquisición mediante búsqueda directa en la web

[Leacock et al. 1998] hicieron un trabajo pionero utilizando una base léxica externa. Utilizaron WordNet para obtener palabras sinónimas monosémicas de la palabra

objetivo y con ellas encontrar ejemplos de entrenamiento a partir de un corpus muy grande. Por ejemplo, utilizando *business suit* como sinónimo monosémico de *suit* se pueden encontrar frases o ejemplos de entrenamiento del correspondiente sentido de *suit*.

[Mihalcea y Moldovan 1999] utilizaron sinónimos monosémicos y glosas de WordNet para construir consultas (*queries*) que lanzaron en el buscador Altavista. Utilizaron cuatro procedimientos ordenados decrecientemente por precisión para rastrear la web en busca de ejemplos:

1. Sinónimos monosémicos. Por ejemplo *recollect* como sinónimo monosémico de *remember-1*.
2. Frases definitorias o glosas. Por ejemplo la glosa de *produce-5* es "bring onto the market or release, as of an intellectual creation". La glosa se analiza automáticamente y se escoge "bring onto the market" como frase definitoria. *Release* sería descartada por ser ambigua.
3. En caso de no tener éxito, se construye una *query* booleana formada por los sinónimos conectados por operadores OR y a su vez en conjunción (operador AND) con palabras de las frases definitorias conectadas entre sí con el operador NEAR. Por ejemplo, para *produce-6*, que tiene los sinónimos *grow, raise, farm* y *produce*, y la frase definitoria "*cultivate by growing*" la consulta sería "*cultivate* NEAR *growing* AND (*grow* OR *raise* OR *farm* OR *produce*)".
4. En caso de volver a fallar se repetiría la consulta del punto 3. pero relajando el operador NEAR sustituyéndolo por el operador AND.

Una vez obtenidos los ejemplos se procesan para comprobar que la categoría sintáctica (*part of speech*) de la palabra objetivo de cada ejemplo es correcta. En caso contrario, se descarta el ejemplo.

Este método produjo una media de 670 frases ejemplo por sentido de la palabra objetivo. El resultado se evaluó manualmente de entre 1080 ejemplos de 120 sentidos y el 91% de los ejemplos resultaron correctos. A pesar de que la precisión es alta el sistema no se probó como entrenador de un sistema DSP real.

[Agirre y Martinez 2000] reprodujeron la misma estrategia para construir un corpus de entrenamiento etiquetado y lo utilizaron para entrenar un sistema DSP supervisado y chequeado frente a un subconjunto de Semcor[8]. Los resultados fueron desalentadores y sólo algunas palabras obtuvieron mejores resultados que los aleatorios[9]. Los autores

[8] Semcor, es el corpus etiquetado con sentidos más grande de acceso público. Está compuesto de documentos extraídos del BC (Brown Corpus) y está etiquetado tanto sintáctica como semánticamente. Las etiquetas sintácticas se anotaron utilizando el etiquetador de Brill [Brill 1995], y las etiquetas semánticas se anotaron manualmente utilizando los sentidos de WordNet 1.6, por el propio equipo que creó WordNet.

[9] Los resultados aleatorios se obtienen adjudicando los sentidos posibles predeterminados utilizando un generador de esos sentidos aleatorio.

atribuyeron los malos resultados a que, aunque los ejemplos extraídos fueran en gran proporción correctos, podrían estar produciendo rasgos erróneos sistemáticamente, y a que el número de ejemplos de cada sentido no era proporcional a la frecuencia de los sentidos (de hecho todos los sentidos recibieron el mismo número de ejemplos de entrenamiento).

Los mismos autores repitieron un experimento parecido en [Agirre y Martinez 2004]. En este caso se concentraron en la técnica de los sinónimos monosémicos y mostraron que puede servir para extraer ejemplos de todos los sustantivos de WordNet. Estudiaron el efecto de la frecuencia de los sentidos y compararon las siguientes posibilidades:

1. Mismo número de ejemplos por cada sentido.
2. Todos los ejemplos encontrados en la web.
3. Misma proporción de ejemplos por sentido que en Semcor.
4. Misma proporción de ejemplos por sentido que en el corpus de entrenamiento de Senseval-2.
5. Misma proporción que la obtenida por el método de [McCarthy 2004] (Sección 2.2.4).

La proporción de sentidos resultó tener gran influencia en el *recall* obtenido: 38% en el peor caso (mismo número de ejemplos) frente a 58% en el mejor (misma proporción que Senseval-2). Por tanto mantener la proporción de Senseval-2 resultó óptimo en los datos de Senseval-2. En general, los resultados obtenidos (en cuanto a *recall*) en Senseval-2 muestran que entrenar un sistema DSP supervisado con datos de entrenamiento procedentes de la web produce resultados mejores que cualquier sistema no supervisado de entre los participantes en esa competición. Por tanto, los datos procedentes de la web pueden ser muy útiles para DSP. Sin embargo, todavía no alcanzan los resultados obtenidos por los corpus anotados (etiquetados) manualmente, ni siquiera la línea de base del sentido más frecuente.

Según la opinión de [Gonzalo y Verdejo 2006] la razón de este bajo rendimiento de los datos de entrenamiento procedentes automáticamente de la web estaría en que éstos solamente capturan una fracción de todos los posibles ejemplos, justo aquellos que ocurren junto a los términos de la consulta, pero ninguno de todos los demás.

Una forma de resolver este problema sería buscar la web con el conjunto de consultas más amplio posible para cada sentido de la palabra objetivo. Esto es lo que intentan los métodos descritos en la siguiente sección.

Autoarranque a partir de ejemplos semilla

[Mihalcea 2002a] elabora el método descrito en la sección anterior [Mihalcea y Moldovan 1999] utilizando un enfoque de *bootstrapping* inspirado en el algoritmo de Yarowsky.

El método genera un conjunto de semillas extraídas de Semcor, WordNet y la web, utilizando el método de los sinónimos monosémicos. Estas semillas forman consultas que atacan la web. Las palabras que rodean a la palabra objetivo (sustantivos y construcciones verbo/sustantivo) en los documentos obtenidos de la web se desambiguan utilizando el algoritmo de [Mihalcea y Moldovan 2000]. A su vez estas expresiones sirven como semillas de una nueva búsqueda en la web.

El corpus etiquetado generado (llamado GenCor) logró uno de los mejores resultados en Senseval-2, y ello debido al corpus obtenido de la web: en la tarea de "todas las palabras" la heurística del primer sentido logra una precisión de 64%; utilizando sólo Semcor y Wordnet se logra 65% y con el corpus de la web se llega a 69%.

En [Mihalcea 2002b] compara el mismo sistema con los sistemas supervisados manuales para un conjunto pequeño de palabras objetivo obteniendo resultados similares.

Estos serían los mejores resultados obtenidos hasta ahora para DSP utilizando datos de la web, y confirman el potencial de esta para resolver el problema del CBAC. Sin embargo, este sistema sigue sin ser totalmente automático, ya que utiliza Semcor y otras fuentes de conocimiento léxico como semillas, con lo que estaría sujeto a los mismos problemas que los sistemas basados en conocimiento. Por ejemplo, esa fuente de semillas iniciales tan grande, de momento sólo está disponible para el idioma inglés.

Adquisición a través de directorios de la web

Los directorios web, como por ejemplo Yahoo! Directory o Open Directory Project (ODP)[10] son categorías temáticas jerárquicas que organizan la información de la web, de forma que un usuario pueda navegar (*browse*) por su contenido refinando sucesivamente los temas que le interesen, en lugar de realizar consultas (*query*) a través de un buscador convencional. Una característica importante es que los temas jerárquicos (estructura) de los directorios y las asociaciones de páginas web a ellos se desarrollan manualmente. Los directorios web podrían considerarse una estructura más equilibrada (*balanced*) que la web "desnuda", desde el punto de vista de la web como corpus.

[Santamaría et al. 2003] desarrollaron un sistema para asociar automáticamente sentidos de palabras de WordNet con directorios ODP, bajo la hipótesis de que la asignación de uno o más directorios web a un sentido de una palabra sería una fuente enorme de información temática sobre ese sentido de la palabra. Además esta información no estaría formada exclusivamente por las páginas web correspondientes, sino también por la propia estructura jerárquica de directorios y subdirectorios correspondientes.

[10] http://dmoz.org

Este sistema funciona lanzando una *query* formada por sinónimos de cada sentido de la palabra objetivo de WordNet (de forma parecida a la descrita en la sección 2.6.1; además se pueden utilizar sinónimos de los otros sentidos de la palabra objetivo como información negativa en la *query*) sobre el sistema de directorios ODP, que a su vez responde con un conjunto de directorios (en lugar de documentos). Esta respuesta se compara con el sentido original. Para ello se dispone tanto de la propia cadena de subdirectorios que forman cada directorio respuesta, como de la propia cadena de hiperónimos del sentido de WordNet. Se aplican una serie de filtros para descartar posibles asociaciones erróneas y finalmente se calcula una medida empírica de confianza en cada asociación.

Este sistema se puede aplicar de forma casi inmediata para obtener un corpus etiquetado con sentidos. Basta extraer las ocurrencias de la palabra objetivo, bien en el conjunto de páginas web del directorio (o directorios) correspondientes, o bien en el resumen manual que describe las páginas del directorio (del sistema ODP).

Según [Gonzalo y Verdejo 2006] este sistema tendría, comparado con los descritos hasta ahora, mostrando *a priori* las siguientes ventajas: 1) la web catalogada es una fuente de información más equilibrada que la web en sí. 2) como los resultados de la consulta son directorios en vez de páginas web en el sentido de los buscadores estándar, muchas de las ocurrencias de la palabra objetivo en esos directorios no tienen por qué ocurrir a la vez que las palabras sinónimas de la consulta original, con lo que se permitiría una variedad mayor de ejemplos de entrenamiento. 3) los directorios no están sujetos a copyright y además son más estables en el tiempo que las propias páginas web. Como desventaja estos autores señalan que algunos sentidos de palabras no tienen mucha especificidad en un determinado dominio, es decir, no se pueden relacionar fácilmente con un tópico concreto y por tanto no se les puede aplicar directamente el método.

[Santamaría et al. 2003] compararon la calidad de los ejemplos obtenidos de ODP de esta forma con los ejemplos etiquetados manualmente utilizados en la tarea muestra léxica del inglés (English lexical sample task) de Senseval-2. En lugar de utilizar los ejemplos obtenidos de las páginas web de los directorios sólo usaron los ejemplos que aparecían en las páginas que describían esos directorios. El sistema demostró que no tenía suficiente cobertura: de entre 29 sustantivos polisémicos de la tarea sólo 10 obtuvieron ejemplos de dos o más sentidos. Sin embargo, para ese subconjunto la exhaustividad (*recall*) obtenida fue similar a la obtenida con los ejemplos etiquetados manualmente, aunque sólo en la mitad de los casos: en el resto volvió a haber un problema de cobertura, y no hubo suficiente número de ejemplos para obtener el mismo nivel de exhaustividad. Por tanto la cobertura es el principal problema de este método.

Adquisición a través de corpus paralelos

Si se dispone de corpus paralelos (corpus traducidos a uno o más idiomas) se puede desambiguar el sentido de una palabra polisémica en el caso de que la traducción a otro idioma sea diferente para cada sentido [Gale et al. 1992]. Por ejemplo, si una ocurrencia de *bank* en un corpus en inglés se traduce en un corpus paralelo en francés a *rive* podemos adjudicar a *bank* el sentido 'orilla' frente al sentido 'financiero'. De esta forma, corpus paralelos alineados pueden ser una fuente de ejemplos etiquetados para un sistema de DSP supervisado. Por supuesto esto sólo funciona cuando la traducción no conserve la ambigüedad, cosa que suele ser más frecuente (la no conservación) cuanto más lejanas sean las familias lingüísticas de los dos idiomas.

Un método que intenta aliviar el CBAC de esta forma está descrito en [Diab 2003]. Nótese que los corpus paralelos, es decir, traducidos, son un recurso muchas veces elaborado manualmente, por lo que el cuello de botella seguiría existiendo de forma parecida. [Diab 2003] utiliza corpus paralelos traducidos por un sistema de MT automático. El algoritmo utilizado es básicamente el sugerido arriba, pero necesita usar una base de conocimiento externo (en este caso concreto WordNet) para etiquetar los diferentes sentidos desambiguados. Al utilizar corpus traducidos automáticamente, en realidad no se utiliza ningún corpus etiquetado manualmente, aunque sí una base de datos léxica.

Este sistema obtuvo un *recall* (exhaustividad) del 57% en la tarea de todas las palabras de Senseval-2, y *optimizado* de forma probabilística por [Bhattacharya et al. 2004] llegó al 65% sobre la misma tarea. [Diab 2004] comparó un sistema supervisado (utilizando ejemplos manuales) en la tarea muestra léxica del inglés en Senseval-2 con su sistema que utilizó ejemplos automáticos obtenidos con un sistema de MT y obtuvo mucho peores resultados, pero igualó a los mejores sistemas no supervisados en la misma tarea. Hay que notar que un corpus paralelo traducido automáticamente es un corpus 'ruidoso' en el sentido de que su traducción es de peor calidad que una traducción (manual) hecha por una persona.

Otros sistemas parecidos utilizan, en vez de un sistema automático de traducción MT, las equivalencias lingüísticas de las propias bases de datos multilingües de WordNet, como EuroWordNet [Vossen 1998], BalkaNet [Tufis et al. 2004a], o el proyecto MEANING [Vossen et al. 2006]. Por ejemplo [Tufis et al. 2004b] utilizan BalkaNet con la siguiente estrategia: dada una ocurrencia de dos palabras alineadas *w1* y *w2* en dos corpus paralelos, se buscan en el ILI (*Interlingual* Index) que relaciona las WordNets de sus idiomas correspondientes. Si ambas palabras comparten un registro común en este índice, ya se anota ése sentido como el de la ocurrencia de *w1* y *w2*. Si no, se utiliza una medida de similitud semántica entre los pares de registros del ILI. Si hay más de uno, se aplica la heurística del sentido más frecuente de la palabra objetivo. Además se utiliza un algoritmo de *clustering* de sentidos que hace que un sentido decidido se propague a otras ocurrencias de la palabra objetivo no decididas todavía. Este método fue probado

por sus autores sobre una versión anotada de la novela *1984* de George Orwell, logrando una precisión del 75%, que es comparable al ITA (*inter-annotator agreement*) para ese mismo corpus.

[Chan y Ng 2005] han aplicado directamente corpus paralelos a la tarea "*English all-words*" de Senseval-2, usando técnicas similares, y logrando unos resultados (77% de precisión), comparables a los obtenidos por el mismo sistema supervisado, pero entrenado con Semcor (76%), que es un corpus de entrenamiento manual. Su sistema asigna a los sentidos WordNet del corpus de evaluación de la tarea (*English all-words*) las entradas correspondientes de dos diccionarios Inglés-Chino, obteniendo la traducción correspondiente en chino a ese sentido, y extrayendo ejemplos de esas ocurrencias en corpus alineados inglés-chino. El problema de la conservación de la ambigüedad se ignora básicamente, lo cual no impide al sistema lograr un resultado sorprendente.

A pesar de los buenos resultados de estos sistemas, en realidad el problema que pretenden solucionar sigue estando ahí: los corpus paralelos (no obtenidos automáticamente con MT) son recursos manuales y por tanto lentos y caros, por no hablar del uso de bases léxicas externas, como las WordNet.

Una posible solución a este problema sería la creación de corpus paralelos automáticamente, utilizando la web. Existen sistemas que buscan URLs de la web con traducciones paralelas y los filtran y generan corpus alineados. Entre ellos están PTMINER [Chen y Nie 2000], BITS [Ma y Liberman 1999] y STRAND [Resnik 1999b]. Este último obtuvo una precisión de 97% y un *recall* de 83% sobre un conjunto de candidatos de los idiomas inglés y francés. [Resnik y Smith 2003] mejoraron STRAND y obtuvieron un corpus paralelo inglés-árabe de más de un millón de palabras por lenguaje con precisión 95% y *recall* 99%.

Según [Gonzalo y Verdejo 2006] en ese momento estos corpus paralelos automáticos todavía no se habían aplicado a DSP, pero esperaban su aplicación inminente dado el éxito de los corpus paralelos (manuales).

Etiquetado cooperativo usando la web

La web puede considerarse un corpus gigantesco pero también como un sistema cooperativo para una cantidad enorme de usuarios. Estos usuarios pueden cooperar voluntariamente en el etiquetado manual de ejemplos, contribuyendo a aliviar el problema del CBAC.

Esta idea ha sido puesta en práctica en el proyecto Open Mind Word Expert (OMWE)[11] por [Chklovski y Mihalcea 2002]. En este sistema, los ejemplos a etiquetar por los

[11] http://www.teach-computers.org/word-expert.html

usuarios se seleccionan automáticamente entre los más difíciles. Esto se hace usando dos clasificadores automáticos de diferentes tipos que tienen una tasa de acuerdo bastante baja, con lo que los casos de acuerdo tienen una gran precisión, y los de desacuerdo son los que se presentan a los usuarios.

Para mantener alta la calidad de las anotaciones de los voluntarios, se usa un sistema de doble anotación por dos usuarios para cada ejemplo. Aún así, en Senseval-2 la tasa de ITA (inter-annotator agreement) fue mucho más baja (62.8%) [Mihalcea y Chklovski 2003] que la del corpus de test de la competición (85.5%) [Kilgarriff 2001]. Incluso en Senseval-3 los mejores sistemas superaron ligeramente a los voluntarios de OMWE, que se utilizó como corpus de test de la competición. Aparte de esto, el OMWE era ya [Gonzalo y Verdejo 2006] en 2006 el mayor corpus etiquetado manual para DSP en inglés, y estaba trabajando en rumano y en equivalencias de traducción entre inglés-hindi e inglés-francés. Sin embargo, según estos mismos autores su relativamente bajo rendimiento lo hace más apto para colecciones de test que de entrenamiento.

3.1.2 Autoarranque (bootstrapping)

Ninguno de los métodos de la sección anterior logra resolver completamente el problema del CBAC en los sistemas supervisados. Los métodos conocidos como de *bootstrapping* [Abney 2002][Abney 2004] abordan el problema desde una perspectiva diferente: en lugar de pretender lograr automáticamente un conjunto grande de ejemplos de entrenamiento, intentan aplicar un algoritmo llamado de *bootstrapping* que permita etiquetar todo el corpus objetivo a partir de un conjunto muy pequeño de ejemplos (muy precisos) correctos. Estos ejemplos iniciales, llamados semillas, se "propagan" por todo el corpus incrementando poco a poco el número de ejemplos etiquetados hasta que se logra etiquetar todo el corpus objetivo. Dado que el número de ejemplos semilla puede ser muy pequeño, estos métodos logran el objetivo de DSP sin la necesidad de un corpus grande etiquetado manualmente, resolviendo en la práctica el problema del CBAC.

El algoritmo de Yarowsky se encuentra entre estos métodos. Como se verá en los capítulos 5, 6 y 8, el algoritmo de Yarowsky resuelve el problema del CBAC desde el punto de vista del idioma objetivo, desde el punto de vista del tamaño, pero no desde el punto de vista del dominio. Este último aspecto se intentará resolver en el capítulo 10.

Métodos de coentrenamiento

Estos métodos [Blum y Mitchell 1998] utilizan dos clasificadores (viendo el problema de DSP como un problema de clasificación) complementarios. Estos clasificadores se entrenan inicialmente con los ejemplos *semilla* y después se aplican a todo el corpus. Sólo los ejemplos que logran un grado de acuerdo en la predicción suficientemente alto desde el punto de vista de los dos clasificadores se etiquetan. Los dos clasificadores se vuelven a entrenar en el nuevo conjunto de ejemplos etiquetados (normalmente más

grande) y el proceso se repite en varias iteraciones hasta que se logra un número de ejemplos etiquetados estable y suficientemente grande.

Los dos clasificadores complementarios se construyen de acuerdo a dos visiones diferentes (y complementarias) de los datos, es decir, dos codificaciones de rasgos diferentes. Según [Blum y Mitchell 1998] se necesita que las dos visiones sean condicionalmente independientes dada la etiqueta de la categoría (sentido). [Abney 2002] demuestra que esta condición se puede relajar. [Clark et al. 2003] demuestran que el reentrenamiento simple, sin coentrenamiento basado en acuerdo entre dos clasificadores, puede dar en algunos casos resultados igual de buenos y con mucha menor carga computacional.

[Mihalcea 2004] introduce una combinación de coentrenamiento y votación por mayoría, que logra suavizar las curvas de aprendizaje y mejorar el rendimiento medio. Este método requiere una distribución constante entre elementos ya clasificados y elementos todavía sin clasificar, lo cual supone un conocimiento a priori de la distribución de los sentidos en el corpus objetivo, algo poco realista.

[Pham et al. 2005] probaron varias variantes de coentrenamiento en Senseval-2, incluido el método de [Mihalcea 2004]. Los resultados que obtuvieron mostraron que el algoritmo básico de coentrenamiento no es mejor que el de autontrenamiento (próxima sección), pero sus versiones sofisticadas sí obtuvieron mejoras de eficacia (utilizaron aprendizaje Naïve Bayes).

Métodos de autoentrenamiento

Los métodos de autoentrenamiento son básicamente similares a los de coentrenamiento, pero usan un único clasificador (no hay dos visiones de los datos). El clasificador se entrena inicialmente en los ejemplos semilla y luego se aplica al resto del corpus. Sólo los ejemplos que obtienen una medida de confianza superior a un umbral predeterminado se incluyen entre los nuevos ejemplos etiquetados. El proceso se repite en varias iteraciones hasta que el número de ejemplos etiquetados se estabiliza (normalmente en un número muy próximo al total de ejemplos del corpus objetivo).

El algoritmo de Yarowsky [Yarowsky 1995a] es el representante más importante de esta familia de algoritmos y se describe detalladamente en el capítulo 5. En los capítulos 5 y 8 se describen varios aspectos y el rendimiento de este algoritmo.

En el capítulo 10 se introduce una metodología de *bootstrapping* basada en el algoritmo de Yarowsky que intenta solucionar sus deficiencias más importantes, descritas en el capítulo 6. Esta metodología se basa en los mismos rasgos de codificación que el algoritmo original, pero utiliza un enfoque *digital* con votación ponderada, basado en uno de los rasgos de codificación, la propiedad de *un sentido en cada discurso* vista en las secciones 2.2.4 y 5.2.

Capítulo 4. La granularidad de los sentidos en PLN

La gran mayoría de sistemas de Desambiguación del Sentido de las Palabras (DSP) prácticos actualmente utiliza como dato de *entrada* una lista de los sentidos posibles para la palabra o palabras que se desea desambiguar. Esta lista es comúnmente denominada inventario de sentidos. Éste puede obtenerse de un diccionario, de un tesauro, de una ontología o de una base de datos léxica; y otras veces puede estar hecho a medida para cada aplicación concreta; y a veces incluso, como se verá más adelante en este mismo capítulo, puede ser inexistente a priori. La distinción del número de sentidos del inventario, llamado también granularidad, es un asunto que forma parte del diseño del sistema de DSP y que podría calificarse como de los más importantes del mismo.

Tanto los sistemas de DSP basados en conocimiento procedente de bases de datos léxicas externas como los sistemas supervisados basados en corpus etiquetados con sentidos, necesitan que se les suministre el correspondiente material externo de alguna forma. Una de las principales características de este material externo es la granularidad de los sentidos de las palabras a desambiguar. Naturalmente, este material de entrenamiento o de información tiene que ser elaborado por personas especializadas en esas tareas. Como se vio en el capítulo 3 esta elaboración representa un problema importante, sobre todo para los sistemas supervisados, denominado el problema del CBAC.[12]

La elaboración de esos corpus por parte de especialistas (lexicógrafos) es un asunto que no deja de ser controvertido, empezando por el hecho de que *no* hay un número exacto de sentidos para las palabras polisémicas. Por poner un ejemplo, los diccionarios ingleses de la editorial Oxford University Press aparecen en al menos cuatro tamaños diferentes (Main, Shorter, Concise, Pocket), de los cuales los más pequeños presentan

[12] KAB, Knowledge acquisition bottleneck (Cuello de botella de al adquisición de conocimiento, CBAC).

un número más bajo de sentidos para cada palabra. Sin embargo, esto no quiere decir que los sentidos en un diccionario más pequeño tengan que ser un subconjunto de los de la misma palabra en otro más grande. Si se juntan varios sentidos en uno más general que los incluya, ni siquiera hay una correspondencia biunívoca entre estos sentidos colapsados.

Esta naturalmente no es la única crítica respecto a la elaboración de estas bases de datos por parte de lexicógrafos. También se argumenta que el objetivo de estos especialistas es explicar el significado de las palabras a alguien que no lo conozca, cosa que no tiene por qué coincidir con los objetivos de las diferentes tareas de PLN. En general, se constata que los DFE (Diccionario en Formato Electrónico) no han proporcionado las grandes cantidades de semántica 'gratis' que los más optimistas esperaban al principio [Ide and Wilks 2006].

La principal aportación de los DFE a las tareas de DSP es la distinción del número (y definición por algún medio) de sentidos de una palabra polisémica. Desde los años 90 WordNet[13] ha sido la base de datos o el DFE utilizado casi exclusivamente en DSP. Sin embargo, muchos investigadores reconocen que el hecho de que WordNet sea el standard *de facto* desde entonces se debe fundamentalmente a que está disponible de forma gratuita.

La principal crítica que se objeta a WordNet es que, debido a la organización de los sentidos de las palabras alrededor de *synsets*[14], el resultado es que el número de distinciones de sentidos es demasiado alto, o bien, de una granularidad demasiado fina. Esto no sólo ocurre en WordNet, sino en muchos otros diccionarios como el LDOCE.[15] Ya desde 1993, en [Kilgarriff 1993] se señalaba que las personas especializadas en la tarea de etiquetar sentidos no eran capaces de distinguir muchas de las distinciones de sentidos en ese diccionario. Este hecho ha sido confirmado desde entonces por numerosos estudios, que sitúan el nivel máximo de *acuerdo entre anotadores* (*inter-annotator agreement*[16]) de aproximadamente el 80% [Edmonds y Kilgarriff 2002]. Este

[13] WordNet (WN) es una base de datos léxica a gran escala del idioma inglés desarrollada por el Cognitive Science Laboratory de la Universidad de Princeton. En la última versión (2.1) contiene 155.327 palabras correspondientes a 117.597 conceptos léxicos, agrupados en 4 categorías sintácticas: sustantivos, verbos, adjetivos y adverbios. Aunque mantiene ciertas similitudes con los diccionarios monolingües, como las glosas y ejemplos, WN está organizada según relaciones semánticas, siguiendo jerarquías y redes de relaciones entre palabras.

[14] Los synsets son los grupos de palabras relacionadas semánticamente de WordNet.

[15] LDOCE (Longman Dictionary of Contemporary English) es un diciconario de los más utilizados en investigación lingüística, debido a que sus definiciones están escritas con un vocabulario controlado, el Longman's Defining Vocabulary, compuesto por aproximadamente 2000 palabras (y sus derivadas). Además, las palabras están etiquetadas con "subject field labels" que se corresponden básicamente con etiquetas de dominio para cada sentido de cada palabra. También incluye "box codes", que son conceptos básicos como "abstracto", "animado", "humano", etc. Que se pueden utilizar para representar información de preferencias selectivas de sentidos de los verbos.

[16] ITA (inter-tagger agreement) es una medida de un límite superior de la eficacia de un sistema de desambiguación automático calculado comparando cómo dos o más personas especializadas y entrenadas con los mismos criterios etiquetan los mismos datos. El acuerdo (agreement) puede calcularse como coincidencia exacta o como solapamiento. Sin embargo, los sistemas automáticos pueden superar el ITA

problema se ha intentado solucionar en el caso de WordNet juntando varios subsentidos muy finos en un sentido más general que los abarque [Dolan 1994][Chen y Chang 1998][Palmer et al. 2006]. Sin embargo esto no ha resultado una tarea fácil como muestran estos trabajos. Por ejemplo, no se sabe todavía en qué nivel se debe parar de juntar sentidos, caso de que se pudiera determinar ese nivel. Como se apunta en [Ide y Wilks 2006] este es un asunto que hasta ahora no ha sido abordado seriamente por los investigadores en DSP. En general, según estos autores, a pesar de su uso verdaderamente extendido en PLN, se ve poca literatura encaminada a mejorar WordNet para resolver el problema de determinar automáticamente el sentido de las palabras.

Tanto desde un punto de vista teórico lingüístico [Wierzbicka 1989][Pustejovsky 1995] como desde uno más práctico en DSP [Schütze 1998] hay toda una corriente que rechaza la propia noción de inventario o lista predefinida de sentidos de las palabras. En el caso de DSP se intenta determinar las distinciones de sentidos basándose sólo en las ocurrencias de grupos de palabras en los contextos. Aunque esto parezca una solución relativamente tautológica, por lo menos podría tener la virtud de darnos una idea del nivel de granularidad necesario con relación al estado del arte en DSP, y no a partir de unas definiciones hechas a priori por una serie de especialistas dedicados en principio y desde mucho antes a asuntos no estrictamente relacionados con el problema "práctico" de la DSP.

Otra corriente dentro de la DSP trata de determinar los sentidos basándose exclusivamente en correspondencias entre traducciones de corpus paralelos en dos o más idiomas [Brown et al. 1990][Dagan y Itai 1994][Resnik y Yarowsky 1997a, 1997b][Ide 1998][Ide et al. 2001][Ide et al. 2002][Dyvik 1998][Dyvik 2004][Resnik y Yarowsky 2000][Diab y Resnik 2002][Ng et al. 2003][Tufis et al. 2004]. Como se ha visto en la sección 2.3 esta técnica se basa en que la traducción de una palabra en contexto a otro (u otros) idiomas suele producir diferentes traducciones para diferentes sentidos de la palabra traducida. Puede ser necesaria la traducción a más de un idioma, si la ambigüedad se conserva en la primera o subsiguientes traducciones.

Estos trabajos demuestran que esta técnica, al menos desde un punto de vista experimental, ofrece buenos resultados, comparando los *clusters* o grupos de sentidos producidos con los generados por personas especialistas. También tiene su contrapartida teórica entre importantes lingüistas del siglo XX. Sin embargo, desde un punto de vista práctico tiene graves problemas para su aplicación en PLN, al menos en un futuro próximo. El principal inconveniente es que la obtención de un inventario "completo" por este método requeriría cantidades ingentes de datos (relativamente difíciles de conseguir) e incluso un reprocesamiento continuo de nueva información debido a la evolución y renovación de los idiomas. Otro problema consistiría en cómo identificar o

---

manual, porque la adjudicación de los sentidos ideales ocurre depués de que el acuerdo se compruebe, de forma que los sistemas estarían comportándose como personas entrenadas lingüísticamente, que han aprendido del corpus.

referirse en esta hipotética "lista" a los diferentes sentidos sin tener que recurrir de algún modo a una "definición" más o menos parecida a la de un diccionario. Un inconveniente más técnico surgiría de posibles conflictos entre idiomas: por ejemplo, un idioma concreto podría hacer distinciones demasiado finas; en ese caso, no sabríamos si distinguir el sentido más fino de todos, o bien el más común, o el de todos los idiomas, etc. A pesar de todo esto, el uso de la comparación de traducciones en corpus paralelos parece ser la mejor opción para elaborar un inventario de sentidos.

4.1 La DSP en aplicaciones de PLN

Las aplicaciones clásicas y mejor establecidas del PLN son la Recuperación de la Información (Information Retrieval, IR) monolingüe, y la Traducción Automática (Machine Translation, MT). De las dos, la más representativa es la MT, porque sirve para evaluar casi cualquier teoría de PLN y porque a su vez es la única que tiene unos criterios indiscutibles de evaluación. Esto se debe, sobre todo, a que todo el mundo sabe lo que es una buena traducción a su idioma materno, sin necesidad de saber lingüística o PLN. Esto no ocurre con otras tareas, como la DSP.

La MT sirvió para justificar la introducción de la DSP, porque se pensó que ésta la ayudaría considerablemente. Sin embargo, esto no se produjo y los éxitos de la DSP a niveles de precisión de 95% no se han "transmitido" a la MT, hasta el punto de que ni en ésta (ni en IR) hay casi nunca una fase de DSP explícita.

En ambos casos, MT e IR, los sistemas necesitan realizar DSP, y de hecho lo hacen, pero de forma indirecta o como efecto lateral de otros procesos en lugar de hacerlo de forma explícita.

4.2 El nivel de granularidad necesario

Ya desde 1994 [Dagan y Itai 1994] han sostenido que la distinción de sentidos que realmente usan los sistemas de PLN es a nivel de homógrafo (por ejemplo, *planta* es un homógrafo que tiene un significado como 'vegetal' y otro como 'factoría'). Por ello consideran que esa es la distinción necesaria por definición para PLN.

Desde un punto de vista más especulativo, muchos autores también consideran la distinción a nivel de homógrafo como la distinción básica [Wierzbicka 1989][Ruhl 1989][Antal 1965]. Esta tradición tiene su paralelo en PLN, donde hay autores que, por ejemplo prefieren léxicos más bien compactos, o reglas por las que se extienden los léxicos [Wilks y Catizone 2002].

La acepción estricta de homógrafo, es decir, palabras totalmente independientes entre sí desde un punto de vista etimológico, que accidentalmente se escriben igual en un idioma determinado, puede relajarse y aceptar en esta distinción básica a palabras muy

polisémicas pero relacionadas etimológicamente, como por ejemplo, en español, *clase* como 'categoría' o como 'aula'.

Este hecho no se basa en la teoría, sino que se fundamenta en experimentos psicolingüísticos. [Klein y Murphy 2001] [Klein y Murphy 2002] [Rodd et al. 2002] [Rodd et al. 2004] realizaron experimentos con personas a las que primero se presentaba una palabra polisémica en un contexto, y luego la misma palabra en otro contexto con otro significado muy diferente. Comprobaron que los tiempos de reacción de las personas no eran más cortos en el caso de los homógrafos estrictos que en el de las palabras muy polisémicas, pero etimológicamente relacionadas. Esto significa que esas palabras, aunque estén relacionadas etimológicamente, son tan distintas para la mente del sujeto como lo son los homógrafos estrictos, es decir, son igual de relevantes como palabras polisémicas para el PLN.

La distinción de sentidos a nivel de homógrafos es lo suficientemente marcada como para que se pueda distinguir mediante criterios multilingüísticos, comparando las traducciones en corpus paralelos. De esta forma, si la misma palabra se traduce de forma diferente a otro idioma en un corpus paralelo, podemos hacer esa distinción de sentidos a nivel homográfico. Esto ocurre con tanta mayor intensidad cuanto menos emparentados estén los idiomas. Si éstos son muy próximos entre sí puede ocurrir un fenómeno de "encadenamiento" de sentidos [Cruse 1986] [Lakoff 1987] siguiendo el mismo proceso histórico, y la ambigüedad se conservaría con la traducción, por lo que no se podrían distinguir los sentidos homográficos mediante los corpus paralelos.

Un último criterio para la distinción de sentidos a este nivel es su uso en diferentes dominios, o dicho de forma recíproca, si los sentidos no se pueden distinguir por dominio, a efectos prácticos es como si fuera un único sentido.

Por tanto las aplicaciones prácticas de PLN, si necesitan DSP, lo hacen a nivel de sentidos homográficos, que son los sentidos que los psicolingüistas ven representados separadamente en el léxico mental, se pueden distinguir mediante traducciones alineadas o mediante cambios de dominio. Las distinciones de granularidad más fina se necesitan raramente, en esos casos se utiliza otro tipo de procesamiento y en general no se pueden determinar fiablemente a partir del contexto.

[Ide y Wilks 2006] terminan su artículo "recomendando centrar la atención del trabajo en mejorar los sistemas explícitos de DSP para lograr una precisión próxima al 100% para distinciones de sentidos a nivel de homógrafo. Como hemos dicho antes este tipo de distinciones son suficientes para IR, MT y otras aplicaciones de PLN…".

En capítulos posteriores desarrollaremos algoritmos para lograr sistemas de DSP que logren una precisión próxima al 100% para distinciones de sentidos a nivel de homógrafo y que no se vean afectados por el problema del CBAC. Como se verá, los algoritmos actuales que logran esto lo hacen sobre un tipo de corpus muy especial, los

corpus de texto periodístico, y en corpus reales de texto general su precisión es mucho más baja, cercana al 70%. Por tanto, los algoritmos desarrollados lograrán aumentar considerablemente ese nivel de precisión y se acercarán a los niveles del 100% de precisión, en corpus de texto general sin restricciones.

Capítulo 5. El algoritmo de Yarowsky

El algorimo que lleva su nombre fue presentado por David Yarowsky en un artículo de 1995 [Yarowsky 1995] después de varios artículos previos que pueden considerarse relacionados [Yarowsky 1992][Yarowsky 1993][Yarowsky 1994]. En este capítulo se desarrollan los conceptos relacionados que serán aplicados en capítulos posteriores del presente trabajo, prestando especial atención al trabajo original de 1995, ya que ese algoritmo no sólo es importante en sí mismo y como pionero de otros muchos que le han sucedido dentro del campo del aprendizaje semisupervisado, sino también porque constituye la base del algoritmo nuevo que se presenta en este trabajo en el capítulo 10 y que constituye fundamentalmente un sistema para poder aplicar el algoritmo de Yarowsky de 1995 a cualquier corpus de texto general.

5.1 Introducción

En sus trabajos Yarowsky propone un nuevo algoritmo no supervisado (sobre este calificativo, véase capítulo 2) que puede desambiguar palabras con gran precisión en un corpus grande y no etiquetado. En su argumentación indicó las dos propiedades del lenguaje humano que explota su algoritmo: *un sentido por colocación* y *un sentido por discurso*. El mismo autor asegura como, resumen del algoritmo, que es un procedimiento robusto y autocorrectivo, y que exhibe muchas de las ventajas de los métodos supervisados, como la sensibilidad a la información sobre el orden de las palabras, cosa que no hacían otros algoritmos no supervisados. Además dice que utiliza la palabra colocación (*collocation*) en su definición tradicional del diccionario: "que aparece en la misma localización; una yuxtaposición de palabras". Véase la sección 2.1 para la fuente de conocimiento y los rasgos que utiliza Yarowsky.

5.2 La propiedad "un sentido por discurso"

Esta propiedad fue introducida por Gale, Church y Yarowsky en [Gale et al. 1992] y establece que las palabras tienden a exhibir sólo un sentido por discurso o documento (véase capítulo 2 para una descripción de la propiedad). Según el autor, esta propiedad no había sido explotada en su totalidad para la desambiguación de sentidos hasta este artículo de 1995.

El algoritmo no hace un uso "duro" de esta propiedad, porque esta se usa conjuntamente con la otra propiedad (un sentido por colocación), que modela el contexto de la palabra objetivo, y cuando ésta última propiedad arroja una prueba probabilística muy fuerte, la primera se ignora.

En su artículo de 1995 Yarowsky muestra una tabla con los resultados de *precisión* (*accuracy*) y *aplicabilidad* (*applicability*) de la propiedad para el mismo corpus en el que más adelante obtiene los resultados de desambiguación (véase la sección 5.7 para una discusión sobre el corpus utilizado), para 37 232 ejemplos etiquetados manualmente, y para un conjunto de diez homógrafos importantes en la literatura. Esos resultados se muestran en la Tabla 5.1. Según esta tabla, la propiedad se cumple con gran fiabilidad, y por tanto se puede explotar como fuente de información para la desambiguación.

5.3 La propiedad "un sentido por colocación"

Esta propiedad fue observada y cuantificada por Yarowsky en [Yarowsky 1993], y establece que las palabras tienden a exhibir sólo un sentido para una determinada colocación. En otras palabras, dado un ejemplo de la palabra objetivo, con su contexto, si en ese contexto figura una colocación (hay una determinada palabra en ese contexto), normalmente en todos los casos (ejemplo/contexto) en que ocurra esa misma colocación, la palabra objetivo exhibirá el mismo sentido.

En este contexto, "colocación" significa que una palabra determinada está colocada en el contexto de la palabra objetivo en una determinada posición. Por ejemplo, inmediatamente a su derecha (+1), inmediatamente a su izquierda (-1), dos lugares a su derecha o izquierda (+2, -2), o, entre otras posibilidades, dentro de una ventana de *+/-k* palabras a su derecha o a su izquierda. Por ejemplo, si tenemos un contexto de la palabra *planta* que contiene *+/-k* palabras a su derecha e izquierda, y un fragmento de ese contexto fuera "*... en la planta petrolífera se localizó...*", una colocación de este contexto contendría la siguiente información: **palabra**: *petrolífera*; **posición**: *+1*; **sentido**: *factory*. Todos los contextos de *planta* que contuvieran la palabra *petrolífera* inmediatamente a su derecha contabilizarían como un caso más de esa colocación.

Palabra	Sentidos	Precisión (%)	Aplicabilidad (%)
Plant	living/factory	99.8	72.8
Tank	vehicle/container	99.6	50.5
Poach	steal/boil	100.0	44.4
Palm	tree/hand	99.8	38.5
Axes	grid/tools	100.0	35.5
Sake	benefit/drink	100.0	33.7
Bass	fish/music	100.0	58.8
Space	volume/outer	99.2	67.7
Motion	legal/physical	99.9	49.8
Crane	bird/machine	100.0	49.1
Promedio		**99.8**	**50.1**

Tabla 5.1. Muestra la precisión y la aplicabilidad de la propiedad "un sentido por discurso" para diez homógrafos importantes en la literatura. La precisión indica el porcentaje de acierto en la predicción utilizando la propiedad. La aplicabilidad se refiere al porcentaje de veces que se ha podido aplicar la propiedad; en otras palabras, el porcentaje de apariciones del homógrafo en cualquier documento del corpus en las que no era la única ocurrencia en el documento (discurso).

En su artículo de 1993, Yarowsky cuantificó la validez de esta propiedad, aplicando un algoritmo de DSP supervisado. Aplicando siete tipos diferentes de colocaciones: palabra no vacía inmediatamente a la izquierda/derecha, primera palabra no vacía a la izquierda/derecha, pareja sujeto/predicado, pareja verbo/objeto, adjetivo/sustantivo, Yarowsky evalúa la eficacia de este algoritmo en un 92% de precisión con un 98% de recall (exhaustividad). (Véase la sección 6.1 para una descripción más detalla de los experimentos llevados a cabo por Yarowsky).

5.4 El algoritmo de 1995

El hecho de que las colocaciones estén repartidas por todo el corpus y de que sirvan para indicar el sentido de la palabra objetivo, da pie a utilizar un mecanismo de *bootstrapping* a partir de un conjunto muy pequeño de ejemplos etiquetados que sirven como *semillas*.

El algoritmo de aprendizaje o clasificación, tal como lo describe Yarowsky, consta de 5 pasos:

PASO 1:
A partir de un corpus grande, identificar todas las ocurrencias de una palabra polisémica dada (palabra objetivo). Almacenar cada contexto (por ejemplo las *+/-50* palabras a la izquierda/derecha) de cada ocurrencia como líneas de un fichero o conjunto de entrenamiento inicialmente no etiquetado. Por ejemplo[17]:

[17] Los siguientes gráficos son los originales de [Yarowsky 1995].

Sense	Training Examples (Keyword in Context)
?	... company said the *plant* is still operating
?	Although thousands of *plant* and animal species
?	... zonal distribution of *plant* life . ...
?	... to strain microscopic *plant* life from the ...
?	vinyl chloride monomer *plant* , which is ...
?	and Golgi apparatus of *plant* and animal cells
?	... computer disk drive *plant* located in ...
?	... divide life into *plant* and animal kingdom
?	... close-up studies of *plant* life and natural
?	... Nissan car and truck *plant* in Japan is ...
?	... keep a manufacturing *plant* profitable without
?	... molecules found in *plant* and animal tissue
?	... union responses to *plant* closures . ...
?	... animal rather than *plant* tissues can be
?	... many dangers to *plant* and animal life
?	company manufacturing *plant* is in Orlando ...
?	... growth of aquatic *plant* life in water ...
?	automated manufacturing *plant* in Fremont ,
?	... Animal and *plant* life are delicately
?	discovered at a St. Louis *plant* manufacturing
?	computer manufacturing *plant* and adjacent ...
?	... the proliferation of *plant* and animal life
?	... ...

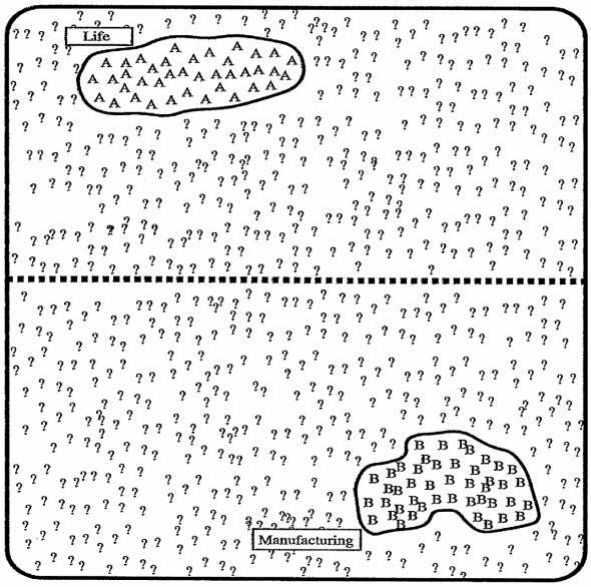
En este fichero cada línea sería un contexto de la palabra objetivo *plant*, inicialmente no etiquetado (código '?').

PASO 2:

Para cada sentido de la palabra objetivo (se suponen dos sentidos, pero el proceso se puede generalizar directamente a más de dos sentidos) identificar un pequeño número de contextos del fichero anterior representativos de cada uno de los dos sentidos. El resto de contextos (no etiquetados) constituyen un conjunto *residual*. Para llevar a cabo este paso se pueden utilizar varias estrategias que requieren una intervención humana (no automática) mínima o nula. Estas estrategias se discuten en la sección siguiente.

Sense	Training Examples (Keyword in Context)
A	used to strain microscopic *plant* **life** from the ...
A	... zonal distribution of *plant* **life** . ...
A	close-up studies of *plant* **life** and natural ...
A	too rapid growth of aquatic *plant* **life** in water ...
A	... the proliferation of *plant* and animal **life** ...
A	establishment phase of the *plant* virus **life** cycle ...
A	... that divide **life** into *plant* and animal kingdom
A	... many dangers to *plant* and animal **life** ...
A	mammals . Animal and *plant* **life** are delicately
A	beds too salty to support *plant* **life** . River ...
A	heavy seas, damage , and *plant* **life** growing on ...
A	... ...
?	... vinyl chloride monomer *plant* , which is ...
?	... molecules found in *plant* and animal tissue
?	... Nissan car and truck *plant* in Japan is ...
?	... and Golgi apparatus of *plant* and animal cells ...
?	... union responses to *plant* closures . ...
?	... ...
?	... ...
?	... cell types found in the *plant* kingdom are ...
?	... company said the *plant* is still operating ...
?	... Although thousands of *plant* and animal species
?	... animal rather than *plant* tissues can be ...
?	... computer disk drive *plant* located in ...
B	... ...
B	automated **manufacturing** *plant* in Fremont ...
B	... vast **manufacturing** *plant* and distribution ...
B	chemical **manufacturing** *plant* , producing viscose
B	... keep a **manufacturing** *plant* profitable without
B	computer **manufacturing** *plant* and adjacent ...
B	discovered at a St. Louis *plant* **manufacturing**
B	... copper **manufacturing** *plant* found that they
B	copper wire **manufacturing** *plant* , for example ...
B	's cement **manufacturing** *plant* in Alpena ...
B	polystyrene **manufacturing** *plant* at its Dow ...
B	company **manufacturing** *plant* is in Orlando ...

En este archivo se han identificado ciertas semillas con el sentido A y otras con el sentido B. El siguiente gráfico ilustra la situación al final del paso 2.



Se representa todos los contextos de la palabra objetivo, unos ya etiquetados, y otros todavía no (símbolo '?').

PASO 3a:

Entrenar un algoritmo de clasificación supervisado[18] con los contextos semilla etiquetados como SENTIDO-A/SENTIDO-B, por ejemplo el algoritmo de lista de decisión. Este entrenamiento consiste en contar el número de colocaciones diferentes entre todos los contextos de entrenamiento para cada uno de los dos sentidos; para cada colocación, calcular el cociente entre el número total de veces que se produjo esa colocación para cada sentido; calcular el logaritmo de este cociente; y añadir este número (log-likelihood) a una lista de colocaciones ordenada decrecientemente por ese número. El siguiente gráfico muestra un ejemplo parcial de la lista de decisión entrenada. La colocación más significativa sería 'life', del sentido A (vegetal), con una log-likelihood de 8.10.

Initial decision list for *plant* (abbreviated)

LogL	Collocation	Sense
8.10	***plant* life**	⇒ A
7.58	**manufacturing *plant***	⇒ B
7.39	**life** (within ±2-10 words)	⇒ A
7.20	**manufacturing** (in ±2-10 words)	⇒ B
6.27	animal (within ±2-10 words)	⇒ A
4.70	equipment (within ±2-10 words)	⇒ B
4.39	employee (within ±2-10 words)	⇒ B
4.30	assembly ***plant***	⇒ B
4.10	***plant*** closure	⇒ B
3.52	***plant*** species	⇒ A
3.48	automate (within ±2-10 words)	⇒ B
3.45	microscopic ***plant***	⇒ A
	…	

[18] Se puede utilizar cualquier algoritmo de clasificación supervisado que devuelva probabilidades para la clasificación. Por ejemplo clasificadores Bayesianos [Mosteller and Wallace 1964], algunos tipos de redes neuronales, pero no reglas de Brill [Brill 1993].

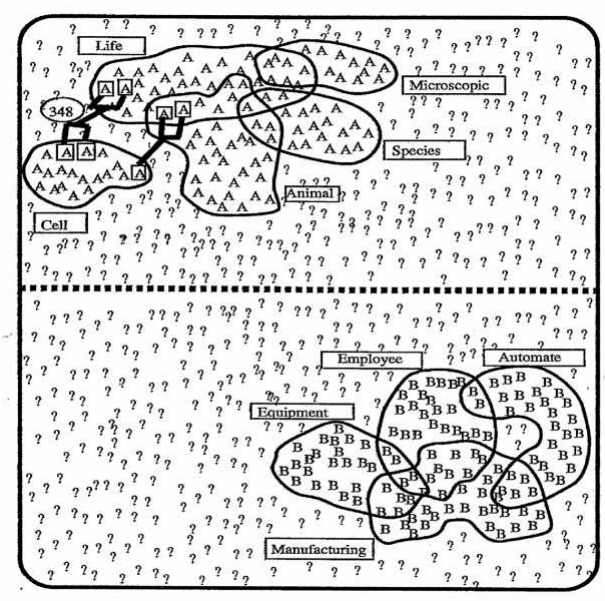
PASO 3b:
Aplicar el clasificador obtenido a *todo* el conjunto de muestra. Esto consiste en, para cada contexto, para cada colocación en él, recorrer la lista de decisión desde el principio (log-likelihood máxima) buscando esta colocación y obtener la log-likelihood correspondiente; de entre todas las colocaciones del contexto, seleccionar la primera en la lista, es decir, la de log-likelihood máxima; si esta log-likelihood supera un umbral predeterminado, añadir su correspondiente sentido (A ó B) como etiqueta del contexto. Estos contextos recién etiquetados se añaden al conjunto de semillas, que va creciendo. Estos contextos son una nueva fuente de colocaciones para futuros entrenamientos.

PASO 3c:
La propiedad *un sentido por discurso* se puede aplicar en este momento opcionalmente. Serviría para filtrar y aumentar la adición del paso anterior: si hay varios contextos de la palabra objetivo *pertenecientes al mismo discurso/documento* que han sido etiquetados con el mismo sentido, digamos A, entonces por esta propiedad sería posible extender esa etiqueta al resto de contextos del mismo documento/discurso. Naturalmente, esta extensión se haría sólo si las probabilidades de los sentidos etiquetados, así como el número de ellos y el número total de contextos del discurso/documento cumplen ciertas condiciones. El siguiente gráfico ilustra esta extensión:

Labeling previously untagged contexts
using the one-sense-per-discourse property

Change in tag	Disc. Numb.	Training Examples (from same discourse)
A → A	724	... the existence of ***plant*** and animal life ...
A → A	724	... classified as either ***plant*** or animal ...
? → A	724	Although bacterial and ***plant*** cells are enclosed
A → A	348	... the life of the ***plant*** , producing stem
A → A	348	... an aspect of ***plant*** life , for example
? → A	348	... tissues ; because ***plant*** egg cells have
? → A	348	photosynthesis, and so ***plant*** growth is attuned

Esta extensión de los contextos de entrenamiento puede lograr un *puente* entre colocaciones aisladas entre sí dentro de un mismo discurso/documento y que de otra forma no se añadirían al conjunto de contextos etiquetados. Esto se ilustra en el siguiente gráfico:



Análogamente, esta extensión puede corregir contextos mal etiquetados previamente. Esto se puede ver a continuación:

Error Correction using the one-sense-per-discourse property

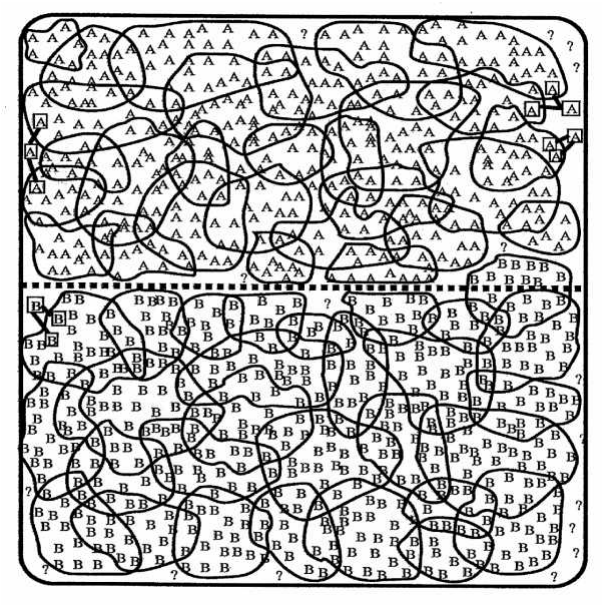
Change in tag	Disc. Numb.	Training Examples (from same discourse)
A → A	525	contains a varied ***plant*** and animal life
A → A	525	the most common ***plant*** life , the ...
A → A	525	slight within Arctic ***plant*** species ...
B → A	525	are protected by ***plant*** parts remaining from

PASO 3d:
Repetir el paso 3 iterativamente. El número de ejemplos etiquetados aumentará en cada iteración y el de no etiquetados disminuirá progresivamente.

PASO 4:
Parar cuando los parámetros del sistema y el número de contextos no clasificados se estabilicen.

Nótese que cada contexto contiene muchas colocaciones distintas, y que incluso puede haber conflictos entre ellas. Sin embargo, el algoritmo de la lista de decisión no hace ningún tipo de cálculo de combinaciones entre ellas, sino que sólo usa una, la más fiable. Esta estrategia elimina muchos de los inconvenientes que surgirían debido al uso de hecho de fuentes de conocimiento no independientes.

PASO 5:
El clasificador utilizado en la última iteración supervisada se puede aplicar a todo el corpus, anotando todos los ejemplos con etiquetas de sentido y probabilidades.



5.5 Opciones para elegir las semillas originales

Las semillas originales deben ser muy precisas y productivas, en el sentido de no contener errores (o tener los mínimos posibles) y describir los sentidos de forma que el número de nuevas semillas pueda aumentar. En su artículo de 1995 Yarowsky indica tres posibles estrategias para etiquetar contextos semilla iniciales, que van desde procedimientos totalmente automáticos a otros que requieren intervención manual. Son las siguientes:

- **Usar palabras de un diccionario**

  Este método consiste en buscar una entrada del sentido correspondiente en un diccionario, y extraer automáticamente palabras que aparecen en esa entrada con mucha mayor frecuencia en esa entrada que en todo el diccionario. Las palabras de esta entrada se pueden ponderar según su colocación en la definición, como se indica en [Yarowsky 1993]. Estas palabras pueden utilizarse como semillas en los ejemplos del corpus objetivo.

- **Usar una sola colocación para cada sentido**

  Este método es menos sofisticado que el anterior y consiste en identificar una única colocación (palabra, posición, sentido) significativa de cada sentido y etiquetar como ejemplos semilla iniciales todos aquellos contextos del corpus objetivo que contengan esa palabra. Yarowsky indica que WordNet puede ser una fuente automática de esas palabras.

- **Etiquetar colocaciones muy frecuentes del corpus objetivo**

Las palabras que ocurren con mucha frecuencia en el contexto de la palabra objetivo suelen ser muy indicativas de uno de los sentidos de ésta. Sin embargo, es imprescindible que una persona actúe como juez y decida a cuál de estos (dos) sentidos pertenece una palabra dada. Yarowsky indica que este proceso manual se puede realizar de forma rápida: menos de dos minutos para una lista de 30 a 60 de esas palabras. Además, un análisis de ocurrencias conjuntas puede evitar el solapamiento y optimizar la labor manual del juez.

En la tabla de resultados al final del capítulo (Tabla 5.2) se exponen los obtenidos con cada una de estas tres estrategias.

5.6 Uso de la propiedad de *un sentido por discurso*

Como se indicó en el paso 3c del algoritmo, se puede incrementar la eficacia de éste aplicando en ese paso, o bien sólo al final de todo el algoritmo, la propiedad conocida como *un sentido por discurso*. Según esta propiedad, todos los ejemplos de la palabra objetivo que ocurran en el mismo documento/discurso tenderán a referirse al mismo sentido de esa palabra.

Cuando la propiedad se usa sólo al final de todo el algoritmo, se hace para corregir errores. Los ejemplos de un determinado documento que se etiquetaron para un sentido con un nivel de confianza bajo, se pueden corregir con el sentido mayoritario en ese documento. Para tomar esta decisión se tiene en cuenta el número total de ejemplos del documento (n), el número de ejemplos etiquetados en cada sentido y las puntuaciones de fiabilidad logradas por cada etiquetado. Si la suma de las fiabilidades de un sentido excede la del otro por un umbral determinado, teniendo en cuenta el número de ejemplos total del documento, se sobrescriben los casos minoritarios. Para $n=2$ esto no suele ocurrir, pero para $n>=4$ casi todos los casos minoritarios suelen corregirse, excepto los muy fiables.

Cuando la propiedad se usa al final de cada iteración (paso 3c), sirve para evitar que decisiones equivocadas iniciales ganen terreno. La aplicación de la propiedad es básicamente igual, excepto que ahora, si no se decide claramente el sentido mayoritario del documento, en vez de dejar las etiquetas ya decididas como estaban, *todas* las etiquetas del documento se devuelven al conjunto inicial de ejemplos no decididos.

La principal desventaja de esta propiedad es la cobertura: suele haber bastantes ejemplos de la palabra objetivo que se encuentran solos en su documento.

5.7 Evaluación y resultados

Yarowsky presenta resultados de la aplicación de su algoritmo a doce palabras objetivo extraídas aleatoriamente de un conjunto de palabras fuertemente polisémicas estudiadas previamente en la literatura. El número de palabras es relativamente bajo debido al coste muy elevado en tiempo necesario para etiquetar a mano los ejemplos que se utilizan para evaluar los resultados.

El corpus utilizado fue "un corpus de 460 millones de palabras que contiene artículos de periódicos, abstracts científicos, transcripciones de lenguaje hablado, y novelas".

Palabra	**Sentidos**	**Contextos**	**Sentido más frecuente**	**Algoritmo supervi-sado**	**Dos palabras**	**Diccio-nario**	**Manual**	**Sólo al final**	**En cada iteración**
plant	living/factory	7538	53.1	97.7	97.1	97.3	97.6	98.3	98.6
space	volume/outer	5745	50.7	93.9	89.1	92.3	93.5	93.3	93.6
tank	vehicle/container	11420	58.2	97.1	94.2	94.6	95.8	96.1	96.5
motion	legal/physical	11968	57.5	98.0	93.5	97.4	97.4	97.8	97.9
bass	fish/music	1859	56.1	97.8	96.6	97.2	97.7	98.5	98.8
palm	tree/hand	1572	74.9	96.5	93.9	94.7	95.8	95.5	95.9
poca	Seal/boil	585	84.6	97.1	96.6	97.2	97.7	98.4	98.5
axes	Gris/tools	1344	71.8	95.5	94.0	94.3	94.7	96.8	97.0
duty	tax/obligation	1280	50.0	93.7	90.4	92.1	93.2	93.9	94.1
drug	medicine/narcotic	1380	50.0	93.0	90.4	91.4	92.6	93.3	93.9
sake	benefit/drink	407	82.8	96.3	59.6	95.8	96.1	96.1	97.5
crane	bird/machine	2145	78.0	96.6	92.3	93.6	94.2	95.4	95.5
Promedio		**3936**	**63.9**	**96.1**	**90.6**	**94.8**	**95.5**	**96.1**	**96.5**

Tabla 5.2. Resultados de precisión en porcentaje del algoritmo de Yarowsky para varias estrategias de selección de ejemplos semilla (*algoritmo supervisado*, *dos palabras* y *diccionario*) y de aplicación de la propiedad *un sentido por discurso* (una vez sólo al final y en cada iteración), comparados con un algoritmo supervisado y la línea de base de referencia del sentido más frecuente.

Los resultados obtenidos se presentan en la Tabla 5.2. La tercera columna muestra el número de ejemplos/contextos correspondiente a cada palabra objetivo en el corpus. La columna cuatro indica el porcentaje de frecuencia del sentido más frecuente. Este porcentaje se utiliza como línea de base para comparar la precisión de los algoritmos, en porcentaje y para un *recall* del 100%. Normalmente se supone que un algoritmo con unas prestaciones mínimas debe superar esta línea de base. En las columnas 6, 7 y 8 se muestran los resultados en porcentaje de precisión para un *recall* del 100% del algoritmo de Yarowsky sin aplicar la propiedad de *un sentido por discurso* y siguiendo tres estrategias diferentes para elegir los ejemplos semilla iniciales: en la columna 6, la estrategia de utilizar dos palabras que representen una colocación importante para cada sentido, en la columna 7 la utilización de la definición de un diccionario, y en la columna 8 el etiquetado manual de colocaciones muy frecuentes del corpus objetivo. En las columnas 9 y 10 se dan los resultados de la aplicación de la propiedad *un sentido por*

discurso sólo al final del algoritmo (columna 9) y al final de cada iteración (columna 10). En ambos casos la estrategia de elección de semillas iniciales es la de usar la definición de un diccionario (columna 7). En la columna 5 se ven los resultados de precisión de un algoritmo supervisado estándar usando una lista de decisión sobre el mismo corpus y sin usar información de discurso, es decir, la propiedad un *sentido por discurso*.

Capítulo 6. La dependencia de las colocaciones con el dominio

Como se ha visto en la sección 2.1, el algoritmo de Yarowsky utiliza como principal fuente de conocimiento las *colocaciones*, (KS 3) en términos de [Aguirre y Stevenson 2006] o bien la propiedad *un sentido por colocación*, en términos de [Yarowsky 1993]. En este capítulo se describe esta hipótesis tal como apareció en este artículo y a continuación se describen otros artículos que investigan la relación entre la hipótesis y las fluctuaciones de dominio en los corpus reales. Como se verá, las variaciones de dominio inter e intra corpus afectan considerablemente a las colocaciones en esos corpus y, como consecuencia, a la eficacia de los algoritmos que utilizan esa fuente de conocimiento para su funcionamiento, incluido el algoritmo de Yarowsky (véase el capítulo siguiente).

6.1 La hipótesis *un sentido por colocación*

En [Yarowsky 1993] Yarowsky define colocación como la ocurrencia conjunta de dos palabras con algún tipo de relación. En los experimentos llevados a cabo en ese artículo para probar la hipótesis *un sentido por colocación*, se consideraron entre estas relaciones las siguientes: palabra no vacía inmediatamente a la izquierda/derecha, primera palabra no vacía a la izquierda/derecha, y algunas relaciones sintácticas directas (sujeto/verbo, verbo/objeto y adjetivo/sustantivo).

La hipótesis *un sentido por colocación* afirma que las mismas colocaciones tienden a ser indicadores muy fuertes de los mismos sentidos de la palabra objetivo. En otras palabras, si en el corpus varios ejemplos (con su contexto) de la palabra objetivo exhiben las mismas colocaciones, es muy probable que se refieran al mismo sentido de dicha palabra.

6.1.1 Experimentos para probar la hipótesis

En el artículo mencionado se demostró la validez de esta hipótesis de dos formas relacionadas entre sí: se probó la relación directa entre la entropía de la distribución de probabilidades condicionales de los sentidos dadas las colocaciones y la precisión de un algoritmo que utilizase esas colocaciones como única fuente de conocimiento externo.

Corpus y palabras objetivo

Como corpus objetivo se utilizó "un corpus de 380 millones de palabras que consiste en noticias (AP newswire y Wall Street Journal), abstracts científicos (de NSF y el Department of Energy), debates parlamentarios de Canadian Hansards, una encilopedia médica, 100 libros Harper & Row, y varios corpus pequeños, como el Brown Corpus y frases ATIS y TIMIT".

Se utilizaron diversos tipos de palabras objetivo, clasificados según los siguientes criterios de ambigüedad:

- **Homógrafos etiquetados a mano:** bass, axes, chi, bow, colon, lead, sake, tear,…
- **Traducciones del inglés al francés:** sentence, duty, drug, language, position, paper, single,…
- **Homófonos:** aid/aide, cellar/seller, censor/sensor, cue/queue, pedal/petal,…
- **Ambigüedades del OCR:** terse/tense, gum/gym, deaf/dear, cookie/rookie, beverage/leverage,…
- **Pseudopalabras:** covered/waved, kissed/slapped, abused/escorted, cute/compatible,…

Una posible definición de la ambigüedad a nivel de homógrafo (polisemia fuerte) es la consideración de una palabra ambigua cuando su traducción a otro idioma en un corpus paralelo bilingüe es diferente (véase capítulo 4). En estos experimentos se obtuvieron datos de este tipo de ambigüedades provenientes de traducciones del inglés al francés, utilizando el corpus paralelo bilingüe inglés-francés Canadian Hansards, que registra debates parlamentarios canadienses. Sin embargo, como se advierte en el artículo, este corpus no es suficientemente grande en al menos dos aspectos: primero, la distribución de los dos sentidos polisémicos está muy desviada y, segundo es difícil encontrar suficientes ejemplos de palabras ambiguas; además, no hay otros corpus bilingües inglés-francés suficientemente grandes.

Los homógrafos etiquetados a mano utilizados son palabras del idioma inglés que tienen una polisemia muy fuerte, se escriben igual en los dos sentidos y pueden o no estar

relacionadas etimológicamente (véase sección 4.2). El inconveniente que presenta este grupo de palabras objetivo es, como se advierte en el artículo, que es difícil de obtener, debido a la intervención manual que, además, es subjetiva y fuente de errores. Esto es un ejemplo clásico del problema del CBAC (ver capítulo 3).

Las *pseudopalabras* son un método propuesto en [Gale, Church y Yarowsky 1992b] para atenuar este problema, aunque sólo tiene aplicación en los sistemas de evaluación de los métodos de DSP. Las *pseudopalabras* son ambigüedades de sentido creadas artificialmente cogiendo dos palabras de la misma categoría gramatical del idioma objetivo, por ejemplo en inglés *guerilla* y *reptile*, y juntándolas en una palabra nueva polisémica, en este caso *guerila/reptile*. Esta *pseudopalabra* se hace sustituir a todas las apariciones de las *dos* palabras iniciales en el corpus objetivo, de forma que tenemos una nueva palabra ambigua, que puede utilizarse para evaluar un sistema de DSP. Claramente, la ventaja de este método está en que hay una cantidad casi ilimitada de ejemplos de las palabras originales y además el etiquetado de un corpus de evaluación puede ser muy fiable, dada la divergencia de los sentidos de las dos palabras elegidas.

Los homófonos y las ambigüedades del OCR son un último tipo de ambigüedad para la que existen grandes cantidades de textos escritos y con los ejemplos "etiquetados" con su sentido, si bien, al igual que en el caso de las pseudopalabras su utilidad se limita a los sistemas de evaluación.

Medida de la entropía

La primera parte del experimento consiste en calcular la entropía de la distribución de probabilidades condicionadas de los dos sentidos dadas las colocaciones. Este cálculo se puede hacer para cada palabra objetivo. Dada una palabra objetivo (dos sentidos), se enumeran todas las colocaciones (de un tipo) para ambos sentidos que aparecen en el corpus, junto con el número de veces que ocurren. En [Yarowsky 1993] aparece el ejemplo de la pareja de homófonos Aid/Aide para la colocación *palabra no vacía a la izquierda*, con los datos que aparecen en la Tabla 6.1.

Como se puede ver en la tabla, la mayoría de las colocaciones tienen una frecuencia 0 en alguno de los dos sentidos. Si aplicamos a esa colocación la fórmula de la entropía con uno de los argumentos 0, nos dará como resultado entropía 0, lo cual no resulta aceptable: el hecho de que en esta muestra haya colocaciones con frecuencia 0 en uno de los sentidos no significa que la frecuencia de ese sentido en esa colocación sea 0. En su trabajo, Yarowsky usó validación cruzada de muestras independientes para calcular la frecuencia del sentido minoritario en colocaciones con distribución 1/0, 10/0, etc. Y llegó al resultado de 6% para la primera, 2% para la segunda, etc. Por tanto, en las colocaciones con esa distribución, la frecuencia no es 1/0 sino 0.94/0.06, ni 10/0, sino 0.98/0.02, etc. Calculando así todas las colocaciones y aplicando la media ponderada se obtiene una entropía H=0.09 bits para el ejemplo de la Tabla 6.1.

Colocación	**Aid**	**Aide**
Foreign	718	1
Federal	297	0
Western	146	0
Provide	88	0
Covert	26	0
Oppose	13	0
Future	9	0
Similar	6	0
presidencial	0	63
Chief	0	40
longtime	0	26
aids-infected	0	2
Sleepy	0	1
disaffected	0	1
indispensable	2	1
practical	2	0
squander	1	0

Tabla 6.1. Distribución de las colocaciones de la ambigüedad *aid/aide*. Cada columna indica el número de veces que la colocación ha aparecido en el contexto de la palabra, con cada uno de los dos sentidos.

Algoritmo que usa las colocaciones

La segunda parte del experimento consiste en calcular la precisión o eficacia de un algoritmo que use las colocaciones como única fuente de información para decidir uno de los dos sentidos de las palabras ambiguas, y además comprobar la relación entre esta precisión para cada uno o para todos los tipos de colocación y la entropía para los mismos datos.

El algoritmo puede aplicarse con una sola colocación. En este caso, cuando se procesa un ejemplo nuevo, se obtienen de su contexto todas las colocaciones del tipo dado, y se busca en una tabla de frecuencias para esa palabra objetivo y para esa colocación, como la Tabla 6.1, todas las palabras que correspondan. Se escoge la que tenga mayor frecuencia, y se asigna el sentido que corresponda a esa frecuencia. Si no hay ninguna coincidencia, simplemente se etiqueta el sentido más frecuente.

El algoritmo se puede generalizar a varias colocaciones, con lo cual se usa más de una fuente de prueba. Se usa una *lista de decisión* [Rivest 1987] [Sproat, Hirschberg y Yarowsky 1992]. Primero se calcula la misma información (frecuencia de cada palabra en cada sentido, como en la Tabla 6.1) para cada colocación. Para cada caso se calcula el logaritmo de su cociente de probabilidades:

$$Abs(Log(\frac{\Pr(Sentido_1 \mid colocación_i)}{\Pr(Sentido_2 \mid colocación_i)}))$$

Este valor representa la probabilidad de que si hay esta colocación, el sentido mayoritario sea el correcto de este nuevo ejemplo. Todos estos valores, uno para cada

colocación (*palabra, lugar, sentido*), donde *lugar* representa el tipo de colocación, se incluyen en una lista ordenada decrecientemente por esos mismos valores. En este caso, cuando se procesa un ejemplo nuevo, se extraen de su contexto todas las colocaciones (por ejemplo (*petrolífera, palabra a la derecha, factoría*)) y la primera que esté en la lista, por tanto la más probable, es la que determina el sentido elegido.

Este algoritmo tiene la virtud de no intentar combinar las fuentes de prueba diferentes, ya que éstas (las colocaciones) en este caso no son independientes entre sí. Por tanto, funciona mejor que los métodos Bayesianos, que suponen esa independencia, pero aunque no la supongan, no intenta combinarlas, como por ejemplo los árboles de decisión, sino que simplemente se utiliza la fuente de prueba más fuerte.

Este algoritmo, propuesto en 1993, es claramente un predecesor del algoritmo definitivo de 1995 (ver capítulo 5). La única diferencia importante es que este es un algoritmo supervisado, mientras que el de 1995 es lo que se puede llamar un algoritmo semisupervisado, ya que utiliza una técnica de *bootstrapping* como se ha visto en ese capítulo. Evidentemente, las ventajas del definitivo son evidentes desde un punto de vista del problema de CBAC. El algoritmo de 1993 es un algoritmo supervisado eficaz como tantos otros, incluso puede que más, por las ventajas que ofrece la lista de decisión que se han apuntado en el párrafo anterior, pero no es necesariamente el más eficaz. Por tanto una comparación directa entre las eficacias o precisiones de los dos algoritmos no tiene mucho sentido. Simplemente el algoritmo de 1993 sirve para demostrar la certeza de la hipótesis *un sentido por colocación*, como se verá a continuación.

6.1.2 Resultados

En la Tabla 6.2 se pueden ver los resultados obtenidos en este experimento tal como se exponen en [Yarowsky 1993], tanto de la entropía de las distribuciones de probabilidad condicionada como de la precisión del algoritmo desambiguador supervisado. Se consideran siete tipos distintos de colocaciones y se distingue, para cada una de ellas, entre las diferentes categorías gramaticales de la palabra objetivo. Se incluyen resultados de exhaustividad (*recall*, porcentaje de ejemplos con algún resultado), y en el caso de no obtener resultado se distingue la razón de ello entre que no haya colocación de ese tipo en el ejemplo (columna *sin colocación*) y que no la haya en ningún ejemplo (columna *sin datos*).

Además, los resultados obtenidos por el algoritmo supervisado reflejan la distribución de probabilidades condicionales del sentido dada la colocación, como se puede ver comparando las columnas *precisión* y *entropía*. Esto quiere decir que, por ejemplo, para la colocación *palabra no vacía inmediatamente a la derecha*, si encontramos esa colocación, una media del 97% de los ejemplos van a exhibir el mismo sentido, y además, esta es la precisión esperada de un algoritmo que utilice esa colocación como

fuente de prueba para desambiguar ejemplos con esa misma colocación. Evidentemente esta precisión es mucho mayor que la que habría al elegir un sentido al azar, que sería del 69%, ya que esta es la línea de base del sentido más frecuente en este experimento.

De estos resultados se desprende que la hipótesis *un sentido por colocación* se satisface con muy alta probabilidad para ambigüedades binarias (de dos sentidos). Los resultados de la columna *precisión* van desde 90% hasta 99%, para diferentes tipos de colocación y categoría gramatical y la media es del 95%.

Colocación	Categoría Gramatical	Entropía	Precisión	Recall	Sin Colocación	Sin Datos
Palabra	Todas	0.18	0.97	0.29	0.57	0.14
no vacía	Sustantivo		0.98	0.25	0.66	0.09
inmediatamente	Verbo		0.95	0.14	0.71	0.15
a la derecha(A)	Adjetivo		0.97	0.51	0.27	0.22
Palabra	Todas	0.24	0.96	0.26	0.58	0.16
no vacía	Sustantivo		0.99	0.33	0.56	0.11
inmediatamente	Verbo		0.91	0.23	0.47	0.30
a la izquierda(B)	Adjetivo		0.96	0.15	0.75	0.10
Primera	Todas	0.33	0.94	0.51	0.09	0.40
palabra	Sustantivo		0.94	0.49	0.13	0.38
no vacía	Verbo		0.91	0.44	0.05	0.51
derecha	Adjetivo		0.96	0.58	0.04	0.38
Primera	Todas	0.40	0.92	0.50	0.06	0.44
palabra	Sustantivo		0.96	0.58	0.06	0.36
no vacía	Verbo		0.87	0.37	0.05	0.58
izquierda	Adjetivo		0.90	0.45	0.06	0.49
Sujeto/	Sustantivo	0.33	0.94	0.13	0.87	0.06
Verbo	Verbo	0.43	0.91	0.28	0.33	0.38
Verbo/	Sustantivo	0.46	0.90	0.07	0.81	0.07
Objeto	Verbo	0.29	0.95	0.36	0.32	0.32
Adjetivo/sustantivo	Adjetivo	0.14	0.98	0.54	0.20	0.26
Sólo A y B	Todas	-	0.97	0.47	0.31	0.21
Todas	Todas	-	0.92	0.98	0.00	0.02

Tabla 6.2. Entropía de la distribución de probabilidades condicionales de los sentidos dadas las colocaciones y precisión de un algoritmo supervisado que utilice sólo como fuente de conocimiento diferentes tipos de colocaciones. Las categorías gramaticales se refieren a la palabra objetivo, *no* a las colocaciones. Se puede ver que los resultados del algoritmo supervisado reflejan la distribución de las probabilidades condicionales del sentido dada la colocación, comparando las columnas *precisión* y *entropía*. Téngase en cuenta que ambas medidas están relacionadas inversamente.

Otra conclusión importante de este experimento, sobre todo referida a la eficacia de un algoritmo práctico que utilice la hipótesis como fuente fundamental de prueba para la desambiguación, sería la relación entre el tipo de colocación (más próxima o más lejana de la palabra objetivo) y la exhaustividad o *recall* y la *precisión*. En concreto, se puede observar que para todos los tipos de colocación de la Tabla 6.2 consideradas individualmente, el *recall* nunca supera el 58%. En cambio, considerando todas juntas el *recall* llega hasta el 98%. Clasificando las colocaciones entre locales (palabra no vacía inmediatamente a la izquierda/derecha y adjetivo/sustantivo) y no locales (primera palabra no vacía a la izquierda/derecha, sujeto/verbo y verbo/objeto) la *precisión* cae desde niveles 96-97% hasta niveles de 92-94%.

Esta última observación está relacionada también con la *precisión* para distribuciones muy poco frecuentes (Figura 6.1). Para una distribución 1/0 se esperaría que la predicción del sentido basada exclusivamente en ella estaría casi al nivel de la línea de base (69%). Como se puede ver en la Figura 6.1, para la colocación *palabra no vacía inmediatamente a la derecha*, el nivel de precisión para una distribución 1/0 es del 92%, y crece rápidamente hasta el 99% para frecuencias 15/0. En cambio, para una colocación a una distancia de 30 palabras de la palabra objetivo, una distribución 1/0 tiene un poder de predicción de 72% (3% sobre el 69% de la línea de base del sentido más frecuente). Esta propiedad tendrá repercusiones en el diseño de un algoritmo práctico como el algoritmo de Yarowsky de 1995 (véase capítulo 5).

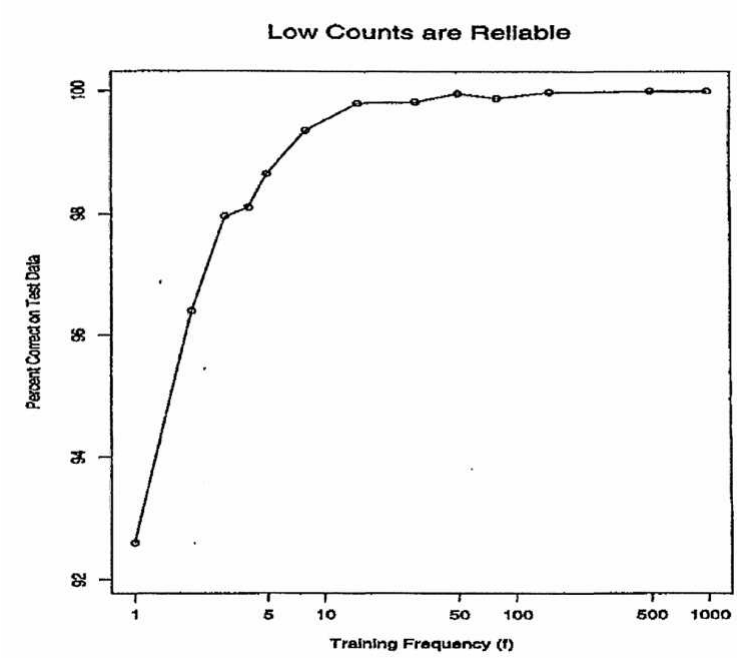


Figura 6.1. Precisión de la desambiguación basada en la colocación *palabra no vacía inmediatamente a la derecha* en función de la frecuencia de la colocación *f* en el corpus de entrenamiento, para casos sin ningún contraejemplo (distribuciones *f/0*).

6.2 Relación entre la hipótesis *un sentido por colocación* y las variaciones de dominio

En [Martínez y Agirre 2000] los autores se preguntan si la hipótesis de *un sentido por colocación*, que fue probada para un determinado corpus y para ambigüedades binarias en [Yarowsky 1993], se satisface también cuando hay cambios de dominio en el corpus o cuando las ambigüedades son de más de dos sentidos.

En su trabajo, Martínez y Agirre usaron como corpus objetivo la colección DSO [Ng y Lee 1996], que está etiquetada con sentidos de WordNet [Miller 1990]. El número de sentidos o granularidad de éstos usada en WordNet es mucho mayor que 2 (el número

de sentidos medio es mayor que 5), un hecho que ha sido criticado abundantemente y también se hace en este trabajo. La colección DSO usa dos corpus diferentes: el corpus equilibrado Brown y el corpus del Wall Street Journal, cosa que es aprovechada en [Martinez y Agirre 2000] para estudiar cómo afectan las variaciones de dominio a la certeza de la hipótesis *un sentido por colocación*.

Un trabajo que se puede considerar pionero de [Martínez y Agirre 2000] es [Krovetz 1998]. En este trabajo se intenta probar la certeza de la hipótesis *un sentido por discurso*, es decir, la segunda propiedad del lenguaje textual usada por el algoritmo de Yarowsky, pero para una granularidad más fina que la de sólo dos sentidos. [Krovetz 1998] utiliza SemCor [Miller et al. 1993] y el propio corpus DSO y llega a la conclusión de que esta hipótesis no se cumple para el nivel de granularidad de WordNet, que es mucho más fino. Más específicamente, aporta resultados en los que aproximadamente el 33% de palabras ambiguas en el corpus tenían más de un sentido dentro de un mismo discurso. Este resultado contrasta con el aproximadamente 4% de palabras en la misma situación descrito en [Gale et al. 1992] para ambigüedad de dos sentidos. Este resultado indica como mínimo que el nivel de granularidad es un aspecto importante, tratado con detalle en el capítulo 4. También se tratan aspectos importantes de la investigación de Krovetz relacionados con el autoarranque del algoritmo de Yarowsky en corpus de texto general en el capítulo 10.

6.2.1 El corpus DSO

La colección DSO [Ng y Lee 1996] se centra en 191 palabras polisémicas frecuentes, todas ellas sustantivos o verbos. Contiene unas 1000 oraciones por palabra objetivo. En total, suman 112.800 oraciones con 192.874 ejemplares de las palabras objetivo, todas ellas etiquetadas con sentidos WordNet [Miller et al. 1990].

Los ejemplos utilizados se obtuvieron del corpus Wall Street Journal (WSJ) y del corpus Brown (BC). Los ejemplos del WSJ aportan 114.794 ocurrencias, y los del BC aportan 78.080.

El Brown Corpus [Francis y Kucera 1964] es en teoría un corpus equilibrado. Sus documentos están clasificados según las categorías de la Tabla 6.3. Sin embargo, los autores del corpus no indican los criterios que siguieron para seleccionar estas categorías, y sólo indican que "los ejemplos representan un amplio rango de estilos y variedades de prosa". Si nos fijamos en la lista de categorías, llegamos a la conclusión de que se confunden las variaciones de dominio (o tópico) con las variaciones de género.

Por ejemplo, la categoría *Religion* representa un tópico más que un género, mientras que las tres categorías *Press* representan géneros. Además, hay tópicos que estarían representados en más de una categoría. Por este motivo Martínez y Agirre advierten en su artículo de que se estudiarán los efectos de las variaciones de "género y tópico", teniendo

en cuenta que "un análisis más detallado sería necesario para medir el efecto de cada uno de ellos".

A.	Prosa informativa	Press: reportage
B.		Press: editorial
C.		Press: Reviews (theatre, books, music, dance)
D.		Religión
E.		Skills and obvies
F.		Popular lore
G.		Belles lettres, biography, memoirs, etc.
H.		Miscellaneous
J.		Learned
K.	Prosa imaginativa	General fiction
L.		Mistery and detective fiction
M.		Science fiction
N.		Adventure and western fiction
P.		Romance and love store
R.		Humor

Tabla 6.3. Lista de las categorías en que están clasificados los textos del BC. Las de arriba se engloban en prosa informativa, y las de abajo en prosa imaginativa. Se puede observar que no hay un criterio riguroso de elección de esas categorías.

6.2.2 Definición de los experimentos

En [Yarowsky 1993] se demuestra la validez de la hipótesis *un sentido por colocación* de las dos formas vistas en la sección anterior: midiendo la entropía de la distribución de probabilidades condicionales, y midiendo la eficacia de un algoritmo supervisado que utiliza listas de decisión entrenadas con las colocaciones como única fuente de conocimiento externo. En ese artículo se demuestra que ambas medidas están muy relacionadas. Por ello, Martínez y Agirre sólo utilizan la medida de la eficacia de un algoritmo supervisado como ese. Además dicen que al medir la eficacia de ese algoritmo en diferentes corpus (o subcorpus) siempre utilizarán el mismo número de ejemplos por palabra objetivo en cada corpus, para eliminar el posible impacto de la dependencia con la cantidad de datos.

Como palabras objetivo utilizaron 21 sustantivos y verbos seleccionados de trabajos previos [Agirre y Martinez 2000][Escudero et al. 2000]. En la Tabla 6.4 se puede ver la lista de estas palabras objetivo, junto con su número de sentidos y número de ejemplos del BC y del WSJ.

6.2.3 Colocaciones

En los experimentos se utilizaron una serie de colocaciones que pudieran extraerse fácilmente de un corpus etiquetado con categorías gramaticales (*part of speech tagged corpus*) y que se pueden dividir en tres grupos: colocaciones locales de palabras no

vacías, colocaciones locales de palabras funcionales y de categorías gramaticales, y colocaciones globales de palabras no vacías.

Las colocaciones locales de palabras no vacías son cinco: palabra no vacía a la izquierda/derecha, y al menos una palabra no vacía, bien considerando dos palabras a la izquierda/derecha o bien considerando una palabra a la izquierda y una palabra a la derecha.

Las colocaciones locales de palabras funcionales y de categorías gramaticales son diez: para las categorías gramaticales se utilizan los mismos cinco contextos que en el grupo anterior, pero se apunta la categoría gramatical (sustantivo, verbo, adjetivo, etc.) de la palabra; para las palabras funcionales se exige en los contextos palabra a la izquierda/derecha que ésta sea una palabra funcional y en los otros tres contextos que ambas palabras sean palabras funcionales, es decir, en caso de que alguna palabra sea no vacía, se considera una colocación del grupo anterior.

Palabra	Categoría	Sentidos	Ejemplos BC	Ejemplos WSJ
Age	S	5	243	248
Art	S	4	200	194
Body	S	9	296	110
Car	S	5	357	1093
Chile	S	6	577	484
Cost	S	3	317	1143
Head	S	28	432	434
Interest	S	8	364	1115
Line	S	28	453	880
Point	S	20	442	249
State	S	6	757	706
Thing	S	11	621	805
Work	S	6	596	825
Become	V	4	763	736
Fall	V	17	221	1227
Grow	V	8	243	731
Lose	V	10	245	935
Set	V	20	925	355
Speak	V	5	210	307
Strike	V	17	159	95
Tell	V	8	740	744

Tabla 6.4. Palabras objetivo utilizadas en los experimentos de [Martínez y Agirre 2000], su categoría gramatical (sustantivo o verbo), el número de sentidos utilizados y el número de ejemplos (apariciones) en el BC y en el corpus WSJ de cada una de ellas.

Las colocaciones globales de palabras no vacías son dos, cada una con los siguientes contextos: ventana de cuatro palabras alrededor de la palabra objetivo y toda la oración donde aparece la palabra objetivo. En ambas se exige que la palabra de la colocación sea no vacía.

Comparando este conjunto de colocaciones con las que se usan en [Yarowsky 1993] se ve que sólo tienen en común las colocaciones locales de palabra no vacía a la izquierda/derecha.

6.2.4 Adaptación de las listas de decisión a más de dos sentidos

El algoritmo de aprendizaje supervisado de la lista de decisión revisado en la sección anterior fue utilizado en [Yarowsky 1993] para resolver ambigüedades binarias. Este algoritmo se puede adaptar a ambigüedades de más de dos sentidos sin más que generalizar la fórmula de las probabilidades condicionales de los sentidos dadas las colocaciones a *n* sentidos, en vez de sólo dos, de esta forma:

$$\log\left(\frac{\Pr(sentido_i \mid colocación_k)}{\sum_{j\neq i}\Pr(sentido_j \mid colocación_k)}\right)$$

Para el cálculo de las probabilidades se utilizó la cuenta de frecuencias, como en [Yarowsky 1993]. Sin embargo, en el caso de un denominador nulo, no es sencillo generalizar la técnica empleada para este caso en ese trabajo. Por ello, se utilizó una técnica aproximada más sencilla pero válida: cada vez que un denominador era 0, se sustituyó por 0.1. Esta técnica también se utiliza en [Yarowsky 1994] y en el capítulo 8 de esta tesis.

6.2.5 Experimentos dentro de un mismo corpus

Estos experimentos se realizaron en cada una de las dos partes del corpus DSO: en la parte BC y en la parte WSJ. En cada una de ellas se entrenó el algoritmo supervisado y se etiquetó la misma parte. Los resultados se pueden ver en las Tablas 6.5 y 6.6.

De los resultados de estos dos experimentos se pueden obtener las siguientes conclusiones:

- Las colocaciones que producen los mejores resultados en cuanto a precisión son las colocaciones locales de palabras no vacías, y dentro de estas, las que consideran dos palabras. Si embargo, el recall es bastante bajo.
- Los resultados son mejores para el WSJ que para el BC. Esto se debe a que el BC es un corpus equilibrado y por tanto incluye variaciones de género y tópico, cosa que no ocurre en el WSJ. Este resultado es el primero que indica que la propiedad *un sentido por colocación* depende de las variaciones de dominio.
- En cuanto al hecho de estar utilizando más de dos sentidos (granularidad más fina) es claro que estos resultados, con una precisión máxima que no llega al

80%, son mucho más bajos que los obtenidos en [Yarowsky 1993] para ambigüedades binarias, donde se llega a una precisión del 99%.

	Sustantivo		Verbo		Total	
Colocaciones	**Pr.**	**Rec.**	**Pr.**	**Rec.**	**Pr.**	**Rec.**
Derecha	0.768	0.254	0.529	0.264	0.680	0.258
Izquierda	0.724	0.185	0.867	0.182	0.775	0.184
Dos palabras derecha	0.784	0.191	0.623	0.113	0.744	0.163
Dos palabras izquierda	0.811	0.160	0.862	0.179	0.830	0.166
Izquierda y derecha	0.820	0.169	0.728	0.129	0.793	0.155
Total local no vacía	0.764	0.502	0.737	0.497	0.755	0.500
Derecha (func.)	0.600	0.457	0.527	0.370	0.577	0.426
Izquierda	0.545	0.609	0.629	0.472	0.570	0.560
Dos palabras derecha	0.638	0.133	0.687	0.084	0.650	0.116
Dos palabras izquierda	0.600	0.140	0.657	0.108	0.617	0.128
Izquierda y derecha	0.721	0.220	0.694	0.138	0.714	0.191
Derecha (categoría gram.)	0.490	0.993	0.488	0.993	0.489	0.993
Izquierda	0.465	0.991	0.584	0.994	0.508	0.992
Dos palabras derecha	0.526	0.918	0.534	0.879	0.529	0.904
Dos palabras izquierda	0.518	0.822	0.614	0.912	0.555	0.854
Izquierda y derecha	0.555	0.918	0.634	0.891	0.583	0.908
Total local func. y categ.	0.622	1.000	0.640	1.000	0.629	1.000
Oración	0.611	1.000	0.572	1.000	0.597	1.000
Ventana 4	0.627	0.979	0.611	0.975	0.622	0.977
Total global	0.617	1.000	0.580	1.000	0.604	1.000
TOTAL	**0.661**	**1.000**	**0.635**	**1.000**	**0.652**	**1.000**

Tabla 6.5. Resultados de precisión y recall del experimento en el WSJ: entrenamiento en el WSJ y etiquetado en el mismo corpus. Se distingue un número considerable de colocaciones diferentes y dos categorías gramaticales: sustantivo y verbo.

	Sustantivo		Verbo		Total	
Colocaciones	**Pr.**	**Rec.**	**Pr.**	**Rec.**	**Pr.**	**Rec.**
Derecha	0.644	0.203	0.432	0.230	0.562	0.212
Izquierda	0.626	0.124	0.770	0.139	0.681	0.129
Dos palabras derecha	0.657	0.146	0.500	0.103	0.613	0.131
Dos palabras izquierda	0.714	0.092	0.819	0.122	0.774	0.103
Izquierda y derecha	0.647	0.088	0.686	0.114	0.663	0.098
Total local no vacía	0.675	0.405	0.635	0.404	0.661	0.405
Derecha (func.)	0.480	0.503	0.452	0.406	0.471	0.468
Izquierda	0.414	0.639	0.572	0.527	0.464	0.599
Dos palabras derecha	0.520	0.183	0.624	0.113	0.547	0.158
Dos palabras izquierda	0.420	0.131	0.648	0.173	0.516	0.146
Izquierda y derecha	0.549	0.238	0.654	0.160	0.577	0.210
Derecha (categoría gram.)	0.340	0.992	0.356	0.992	0.346	0.992
Izquierda	0.350	0.994	0.483	0.992	0.398	0.993
Dos palabras derecha	0.406	0.923	0.422	0.876	0.412	0.906
Dos palabras izquierda	0.396	0.792	0.539	0.897	0.452	0.829
Izquierda y derecha	0.416	0.921	0.545	0.885	0.461	0.908
Total local func. y categ.	0.486	1.000	0.560	1.000	0.512	1.000
Oración	0.545	1.000	0.492	1.000	0.526	1.000
Ventana 4	0.550	0.972	0.525	0.951	0.541	0.964
Total global	0.549	1.000	0.503	1.000	0.533	1.000
TOTAL	**0.577**	**1.000**	**0.564**	**1.000**	**0.572**	**1.000**

Tabla 6.6. Resultados de precisión y recall del mismo experimento que en la Tabla 6.5, pero sobre el corpus BC, es decir, con entrenamiento en el BC y etiquetado en el mismo corpus.

Como ejemplo de las colocaciones aparecidas y los sentidos manejados, en la Tabla 6.7 se pueden ver las colocaciones locales de palabras no vacías más fuertes de la palabra objetivo *state* en el corpus WSJ, con sus sentidos. Estos seis sentidos son los que aparecen en el corpus, de entre ocho sentidos distinguidos por WordNet. Estos seis sentidos, tal como aparecen en WordNet se pueden ver en la Figura 6.2.

6.2.6. Experimentos entre corpus diferentes

En estos experimentos, se entrena en el BC y se etiqueta en el WSJ, y viceversa. Los resultados de ambos tipos de experimentos se muestran en las Tablas 6.8 y 6.9, respectivamente.

Colocaciones	Log	Sentido 1	Sentido 2	Sentido 3	Sentido 4	Sentido 5	Sentido 6
state government	3.68					4	
six status	3.68					4	
State´s largest	3.68					4	
State of emergency	3.68		4				
Federal, state	3.68					4	
State, including	3.68					4	
Current state of	3.40		3				
State aid	3.40				3		
State where Farmers	3.40	3					
State of mind	3.40		3				
Current state	3.40		3				
State thrift	3.40				3		
Distributable state aid	3.40				3		
State judges	3.40					3	
a state court	3.40			3			
said the state	3.40					3	
Several states	3.40					3	
State monopolies	3.40				3		
State laws	3.50			3			
State aid bonds	3.40				3		
Distributable state	3.40				3		
State and local	2.01			1	1	15	
Federal and state	1.60				1	5	
State court	1.38			12		3	
Other state	1.38	4				1	
State governments	1.09	1				3	

Tabla 6.7. Ejemplo de colocaciones locales de palabras no vacías aparecidas en el corpus WSJ para la palabra objetivo *state*, junto con el número de veces que aparece refiriéndose a cada sentido de WordNet.

1. state -- (*the group of people comprising the government of a sovereign*)
2. state, province
   -- (*the territory occupied by one of the constituent administrative districts of a nation*)
3. state, nation, country, land, commonwealth, res publica, body politic
   -- (*a politically organizad body of people under a single government*)
4. state -- (*the way something is with respect to its main attributes*)
5. Department of State, State Department, State
   -- (*the federal department that sets and maintains foreign policies*)
6. country, state, land, nation – (*the territory occupied by a nation*)

Figura 6.2. Seis sentidos de la palabra *state* tal como aparecen en WordNet 1.6. En total hay 8 sentidos para esa palabra.

	Sustantivo		**Verbo**		**Total**	
Colocaciones	**Pr.**	**Rec.**	**Pr.**	**Rec.**	**Pr.**	**Rec.**
locales no vacías	0.597	0.338	0.591	0.356	0.595	0.344
locales func. y categorías	0.478	0.999	0.491	0.997	0.483	0.998
Globales	0.442	1.000	0.455	0.999	0.447	1.000
TOTAL	**0.485**	**1.000**	**0.497**	**1.000**	**0.489**	**1.000**

Tabla 6.8. Resultados de precisión y recall del experimento de las Tablas 6.5 y 6.6, pero esta vez realizando el entrenamiento en el BC y el etiquetado en el corpus WSJ.

	Sustantivo		**Verbo**		**Total**	
Colocaciones	**Pr.**	**Rec.**	**Pr.**	**Rec.**	**Pr.**	**Rec.**
locales no vacías	0.512	0.273	0.556	0.336	0.530	0.295
locales func. y categorías	0.421	1.000	0.486	1.000	0.444	1.000
Globales	0.392	1.000	0.423	1.000	0.403	1.000
TOTAL	**0.429**	**1.000**	**0.483**	**1.000**	**0.448**	**1.000**

Tabla 6.9. Precisión y recall obtenidos en el experimento dela Tabla 6.8, pero efectuando el entrenamiento en el corpus WSJ y el etiquetado en el BC.

Estos resultados muestran una bajada considerable tanto de la *precisión* como del *recall* frente a los resultados de los experimentos en el mismo corpus: la Tabla 6.8 indica una caída de la precisión de un 16% frente a la Tabla 6.5.

Para analizar la razón de estos resultados, se compararon las colocaciones locales de palabras no vacías extraídas de los dos corpus. La Tabla 6.10 muestra el número de colocaciones extraídas de cada corpus, el porcentaje de ellas compartidas entre los dos corpus y, de estas, el porcentaje de las que se hallan en contradicción, es decir, se refieren a sentidos diferentes.

El número relativamente bajo de colocaciones compartidas por ambos corpus sería la explicación de los malos resultados obtenidos. Otra explicación incluso más fuerte sería la presencia de colocaciones contradictorias, que en algunos casos es bastante alto (por ejemplo en '*point*'). Por ello se inspeccionó manualmente los casos de colocaciones

contradictorias y se llegó a la conclusión de que casi todas se debieron a errores o, al menos, a falta de consenso entre los anotadores manuales.

Palabra	Categoría	Coloc. BC	Coloc. WSJ	% compartidas	%contradictorias
Age	S	45	60	27	0
Art	S	24	35	34	20
Body	S	12	20	12	0
Car	S	92	99	17	0
Chile	S	77	111	40	5
Cost	S	88	88	32	0
Head	S	77	95	7	33
Interest	S	80	141	32	33
Line	S	110	145	20	38
Point	S	44	44	32	86
State	S	196	214	28	48
Thing	S	197	183	66	52
Work	S	112	149	46	63
Become	V	182	225	51	15
Fall	V	36	68	19	60
Grow	V	61	71	36	33
Lose	V	63	56	47	43
Set	V	94	113	54	43
Speak	V	34	38	28	0
Strike	V	12	17	14	0
Tell	V	137	190	45	57

Tabla 6.10. Colocaciones en común y en contradicción entre el BC y el corpus WSJ. La columna de la izquierda se refiere a la palabra objetivo. Las columnas tercera y cuarta se refieren al número de colocaciones diferentes de cada palabra objetivo (palabras diferentes aparecidas en su contexto) en el BC y en el corpus WSJ respectivamente.

Este último hecho hace llegar a la conclusión de que la hipótesis *un sentido por colocación* se cumple entre corpus diferentes, debido a que las contradicciones sólo se deben a errores. Es decir, la baja precisión de los experimentos entre corpus se debe exclusivamente al número bajo de colocaciones en común.

Este es uno de los hechos que ha motivado el estudio de las causas de ese número bajo de colocaciones en común, ya que sería el motivo último de los resultados de baja *precisión* entre corpus.

6.2.7 Utilización de ejemplos de entrenamiento y test obtenidos de los mismos documentos

Una de las razones de la mayor precisión obtenida en los experimentos dentro de un mismo corpus podría deberse al hecho de que, al obtener ejemplos de entrenamiento y ejemplos de test del mismo corpus, podría estar dándose el caso de que ambos tipos de ejemplos se estuvieran obteniendo en realidad de los mismos documentos. Por este motivo, se repitieron los experimentos dentro del mismo corpus, pero garantizando que

ejemplos de entrenamiento y de test salieran de diferentes documentos. Los resultados de ambos experimentos se muestran en las Tablas 6.11 y 6.12.

	Total precisión	**recall**	**Δ precisión**	**Local precisión**	**recall**	**Δ precisión**
Sustantivo	0.650	1.000	-0.011	0.762	0.486	-0.002
Verbo	0.634	1.000	-0.001	0.697	0.494	-0.040
Total	0.644	1.000	-0.011	0.738	0.489	-0.017

Tabla 6.11. Mismo experimento que el de la Tabla 6.5: entrenamiento en el corpus WSJ y etiquetado en el mismo corpus; pero esta vez garantizando que los ejemplos de entrenamiento y de test pertenecen a documentos diferentes del corpus.

	Total precisión	**recall**	**Δ precisión**	**Local precisión**	**recall**	**Δ precisión**
Sustantivo	0.499	1.000	-0.078	0.573	0.307	-0.102
Verbo	0.543	1.000	-0.021	0.608	0.379	-0.027
Total	0.514	1.000	-0.058	0.587	0.333	-0.074

Tabla 6.12. Mismo experimento que el de la Tabla 6.6: entrenamiento en el BC y etiquetado en el mismo corpus; pero esta vez garantizando que los ejemplos de entrenamiento y de test pertenecen a documentos diferentes del corpus.

Los resultados para el corpus WSJ indican que el hecho de elegir ejemplos de entrenamiento y test de los mismos o diferentes documentos no afecta en gran medida a la precisión, mientras que en el caso del BC sí hay un descenso significativo de ésta cuando los ejemplos se extraen de diferentes documentos. Este resultado indica que la mayor variación de género y tópico del BC está afectando a la precisión.

De todas formas, como se ve en los resultados comparados de la Tabla 6.13, este hecho (elegir ejemplos de los mismos o diferentes documentos) no explica por sí solo la gran degradación de la precisión en los experimentos entre corpus diferentes. En la próxima sección se examina la razón de este resultado.

Precisión total	**Mismo corpus (mismos documentos)**	**Mismo corpus (distintos documentos)**	**Distintos corpus**
WSJ	0.652	0.644	0.489
BC	0.572	0.514	0.448

Tabla 6.13. Resultados de precisión totales para experimentos en el mismo corpus, con ejemplos extraídos (posiblemente) de los mismos documentos y con ejemplos extraídos (seguramente) de distintos documentos, y para experimentos entre corpus diferentes. Los corpus son el BC y el corpus WSJ.

6.2.8 La variación de género y tópico

El hecho de que de los dos experimentos dentro del mismo corpus, los del WSJ obtengan mejor precisión que los del BC, ya indica que la mayor variación de género/tópico del BC está influyendo en su menor precisión. La diferencia de precisión

entre los experimentos que exige que los ejemplos de entrenamiento y test provengan de distintos documentos, y el hecho de que esto no suceda así para el BC apuntan a las mismas conclusiones. Sin embargo, para probar definitivamente esta hipótesis, se realizó otro experimento, aprovechando que una de las categorías del BC es 'Press: Reportage', es decir, un género/tópico similar al del WSJ.

En este experimento, se entrenó por un lado en el WSJ y por otro en todas las categorías del BC excepto precisamente la categoría 'Press: Reportage'. Con estas listas de decisión se etiquetó todo el BC. La Tabla 6.14 muestra los resultados.

Categoría	**WSJ** **Locales no vacías**		**Resto BC** **Locales no vacías**	
	Precisión	**Recall**	**Precisión**	**Recall**
Press: Reportage	0.625	0.330	0.541	0.285
Press: Editorial	0.504	0.283	0.593	0.334
Press: Reviews	0.438	0.268	0.488	0.404
Religión	0.409	0.306	0.537	0.326
Skills and Hobbies	0.569	0.296	0.571	0.302
Popular Lore	0.488	0.304	0.563	0.353
Belles Lettres,…	0.516	0.272	0.524	0.314
Miscellaneous	0.534	0.321	0.534	0.304
Learned	0.518	0.257	0.563	0.280
General Fiction	0.525	0.239	0.605	0.321
Mystery and …	0.523	0.243	0.618	0.369
Science Fiction	0.459	0.211	0.586	0.307
Adventure and…	0.551	0.223	0.702	0.312
Romance and…	0.561	0.271	0.595	0.340
Humor	0.516	0.321	0.524	0.337

Tabla 6.14. Resultados de precisión y recall del experimento de etiquetado de las categorías del BC entrenadas con el WSJ y con todas las categorías del BC excepto 'Press: Reportage'. Se han utilizado colocaciones locales (en el contexto de la palabra objetivo) y no vacías (colocaciones de palabras no vacías).

Como se puede ver en esa tabla, los mejores resultados del entrenamiento en el WSJ se obtienen en la categoría 'Press: Reportage' del BC, y además la precisión obtenida en esa categoría con ese entrenamiento también es mejor que la obtenida en esa categoría con el entrenamiento en el resto de categorías del BC.

Esto significa que de entre todas las categorías, las colocaciones de 'Press: Reportage' son las más similares a las del WSJ, o dicho de otra forma, los documentos con género/tópico similares tienen colocaciones en común.

Los resultados del último experimento demuestran que la hipótesis *un sentido por colocación* se cumple en todos los dominios, pero fluctúa, en el sentido de que las colocaciones cambian con las variaciones de dominio. Naturalmente estas variaciones

de las colocaciones influyen en los algoritmos desambiguadores que utilicen esa hipótesis como fuente de conocimiento fundamental.

Capítulo 7. Sistema experimental

El objetivo principal de este trabajo es el desarrollo de un algoritmo de autoarranque de DSP semisupervisado, y por lo tanto no sujeto al problema del cuello de botella de la adquisición de conocimiento (CBAC), que iguale en eficacia a los algoritmos de DSP plenamente supervisados en los corpus de texto general, y no sólo en los corpus de texto periodístico, e incluso pueda potencialmente superarlos en este tipo de corpus, que son los habituales en las aplicaciones reales, aprovechando las características especiales de los algoritmos de autoarranque.

Para la evaluación de sus características ha sido necesario disponer de una serie de corpus ya conocidos en la literatura especializada que permiten la comparación de los resultados obtenidos por distintos grupos de investigación y con un alto número de menciones en publicaciones de referencia.

Además, durante el desarrollo de la investigación ha sido necesario el diseño y desarrollo de distintos programas informaticos destinados al procesamiento de la información, manejo de los corpus y a la implementación del propio algoritmo de Yarowsky. A continuación pasamos a realizar una breve descripción.

7.1. Sistema utilizado para el estudio de la distribución estadística de las fuentes de dominio en los corpus de texto

7.1.1 Corpus utilizados

Se han utilizado dos corpus de uso común en la literatura como representantes de sus respectivas categorías: el corpus WSJ es un corpus de texto escrito de noticias periodísticas aparecidas en el diario norteamericano The Wall Street Journal entre los años 1987 y 1989; y el BNC (British National Corpus), un corpus de texto general escrito, equilibrado, compuesto por muestras de texto de inglés británico aparecidos

fundamentalmente en libros y publicaciones periódicas en el Reino Unido durante los años 1990. El corpus WSJ ha sido obtenido a través de licencia otorgada por el Linguistic Data Consortium (LDC) de la Universidad de Pennsylvania y el BNC también bajo licencia de la Universidad de Oxford.

Muestras de los corpus

Para el estudio de la distribución estadística de las fuentes de dominio en los dos tipos de corpus se han obtenido varias muestras o subconjuntos de cada uno de ellos. La forma de obtención de muestras de los corpus ha estado muy influida por la propia naturaleza de los corpus; mientras el BNC trata de representar todos los tipos de dominio que pueden aparecer en texto general habitual, el corpus WSJ está compuesto única y exclusivamente de artículos periodísticos, lo que le convierte en un tipo muy especial de corpus, como se demuestra en este mismo trabajo. Esto hace que los documentos del BNC, mucho más grandes en general que los del corpus WSJ, se presenten clasificados según varios criterios, siendo el dominio y el medio (libro, publicación periódica, etc.) los dos más importantes. En cambio, los documentos del WSJ, que son artículos de prensa diaria, no se encuentran clasificados explícitamente bajo ningún criterio especial, lo cual no quiere decir que no traten sobre temas muy variados, es decir, que no tengan también variaciones de dominio, aunque no sean tratadas explícitamente.

Esta clasificación de los documentos del BNC según diferentes dominios ha sido utilizada para lograr muestras pertenecientes a dichos dominios tan distintos. Hay que tener en cuenta también que las muestras del corpus no consisten sólo en documentos no procesados de los dominios en cuestión, sino que la naturaleza de la tarea de DSP impone la necesidad de procesar esos documentos para obtener los contextos de las ocurrencias de la palabra objetivo (palabra bajo desambiguación) en dichos documentos. Dicho en otros términos, la muestra del corpus para un dominio dado depende completamente de la palabra objetivo, ya que el contexto de una palabra objetivo en un documento suele ser completamente diferente del contexto de otra palabra objetivo cualquiera, en ese mismo documento.

Es por este motivo por el que en este trabajo se ha utilizado siempre un par corpus/palabra para hacer referencia a un determinado conjunto de muestras, de diferentes dominios, de dicho corpus. Esta nomenclatura también se ha utilizado con el WSJ, aunque en este corpus las muestras no se hayan tomado según dominios; se han tomado por orden cronológico desde el mes de enero de 1987 hasta lograr el tamaño deseado.

7.1.2 Medida para comparar fuentes de dominio

La medida utilizada para estudiar las variaciones de las fuentes de dominio en los corpus ha sido una medida de la ambigüedad. Como se ha señalado en el capítulo 9, ante la ausencia de una medida convencional de la ambigüedad de una muestra de texto dada, se ha recurrido a calcular el valor medio de la precisión de cuatro experimentos de DSP supervisados sobre cuatro algoritmos de aprendizaje automático (*machine learning)* correspondientes y de naturaleza muy diferente. Para llevar a cabo esta tarea se ha contado con la inestimable ayuda del conjunto de algoritmos de aprendizaje automático WEKA, de la Universidad de Waikato (Nueva Zelanda), de distribución libre.

A su vez, las muestras de pares corpus/palabra correspondientes a dominios (BNC) o a periodos de tiempo (WSJ) que se utilizan con el paquete WEKA se deben presentar a este sistema en un formato adecuado. El paquete SenseTools, desarrollado por Ted Pedersen, de la Universidad de Minnesota y por Satanjeev Banerjee, de la Universidad Carnegie Mellon, y disponible públicamente, es un conjunto de programas que transforman corpus entrenamiento y corpus de test en vectores de rasgos adecuados para ser tratados por el paquete WEKA. Los corpus de entrada al paquete SenseTools deben seguir el formato utilizado en el certamen Senseval-2.

El formato utilizado en Senseval-2 para representar un corpus de entrenamiento o de test utiliza el lenguaje de marcado XML. La figura 7.1 muestra una parte del principio de uno de esos ficheros para el par BNC/space. Como se puede observar, el fichero representa fundamentalmente los contextos de la palabra objetivo, en este caso utilizando una ventana de 21 palabras a la izquierda y 21 palabras a la derecha de ella. En el campo *senseid* de la etiqueta *answer* se puede incluir un número de sentido a continuación del valor (string) *space* y del carácter tilde (‘~’), si ese contexto se utilizara como entrenamiento en un fichero de ese tipo. En los ficheros de test se puede hacer lo mismo para informar al sistema del sentido correcto de ese contexto, de forma que éste pueda comparar el sentido que ha determinado con el correcto para calcular la precisión total.

El programa *nameconflate.pl* del paquete SenseTools, escrito en Perl, construye un fichero de formato Senseval-2 a partir de un fichero plano sin formato que represente a la muestra o subcorpus que se quiere utilizar como entrenamiento o como test. Una vez construido el fichero XML según el formato Senseval-2, se deben etiquetar manualmente los sentidos de los contextos que se quieran utilizar como entrenamiento y los que se quieran utilizar como test, dependiendo del tipo de fichero que se esté construyendo. Normalmente en los ficheros de test se etiquetan todos los contextos por exigencias del sistema, y además para tener una muestra aleatoria de la proporción real de ambos sentidos. En cambio en los ficheros de entrenamiento no es necesario etiquetar todos los contextos. Esto se debe a que es muy importante etiquetar el mismo número de contextos de cada uno de los dos sentidos; si no, se estaría perjudicando de

forma grave al sentido minoritario, que lograría unos resultados mucho más bajos de precisión, al contar con menos ejemplos de entrenamiento.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<corpus lang='english'>

<lexelt item="space">

<instance id="1">
<answer instance="1" senseid="space"/>
<context>
a thick party wall evidence of the existence of dry rot may be found in a musty smell pervading a suspect <head>space</head>
cotton wool-like fungal growth on timber  cracking and bulging of joinery mouldings such skirtings and door
</context>

</instance>
<instance id="2">
<answer instance="2" senseid="space"/>
<context>
it may be possible to deduce whether it applies by noting the pattern of nails which have penetrated into the roof <head>space
such an inspection should also help to identify any dampness or rot which has affected the boarding  where
</context>

</instance>
<instance id="3">
<answer instance="3" senseid="space"/>
<context>
hay so this fodder could be pitched readily into the racks in the stalls below because the loft exploited the roof <head>space</h
full height upper storey  in contrast  stables attached to large town mansions in central london
</context>

</instance>
```

Figura 7.1. Ejemplo del principio de un fichero XML que sigue el formato utilizado para la tarea *English Lexical Sample* en el certamen Senseval-2. Como se ve, cada *instancia* representa una aparición de la palabra objetivo *space* en el corpus. Cada ocurrencia de la palabra objetivo tiene un identificador único, y se almacena el contexto de esta palabra, representado por una ventana de 21 palabras a la iquierda y a la derecha de la aparición.

Una vez construido el fichero en el formato Senseval-2, éste se preprocesa con el programa *preprocess.pl*, también escrito en Perl. Este programa produce dos ficheros de salida. El primero es un archivo XML muy parecido al de entrada, en el que se tokeniza el texto entre las etiquetas <context> de acuerdo con las *expresiones regulares*[19] que se suministren como entrada al programa. En nuestro caso estas expresiones regulares son muy sencillas, porque los tokens son simplemente palabras del contexto. El segundo fichero de salida es un fichero plano de texto formado exclusivamente por todas las palabras de todos los contextos, sin formato especial, con el único objetivo de servir de entrada al programa *count.pl* del paquete NSP (Ngram Statistics Package), que contará el número de palabras diferentes y no es capaz de ignorar las etiquetas XML.

Los ficheros de muestra originales se transformarán al final en una representación de vectores de rasgos, donde cada ocurrencia de la palabra objetivo, que corresponde a un contexto, se representará por un vector binario de estos rasgos. En nuestro caso, estos rasgos serán las frecuencias de ocurrencia de cada colocación (palabra distinta) entre

[19] En nuestro caso estas expresiones regulares serán extremadamente sencillas, ya que sólo representarán a palabras únicas tal y como aparecen en el contexto.

todos los contextos. El paquete SenseTools realiza esto mediante dos programas: *nsp2regex.pl*, toma como entrada las frecuencias producidas por *count.pl* y los ficheros XML producidos por *preprocess.pl* y produce expresiones regulares para los rasgos necesarios, que serán la entrada del otro programa, *xml2arff.pl* que construye los vectores de rasgos finales. En estos vectores tendremos codificadas al final las colocaciones (palabras) que hay en cada contexto de la palabra objetivo y cuya frecuencia supere un determinado umbral predeterminado.

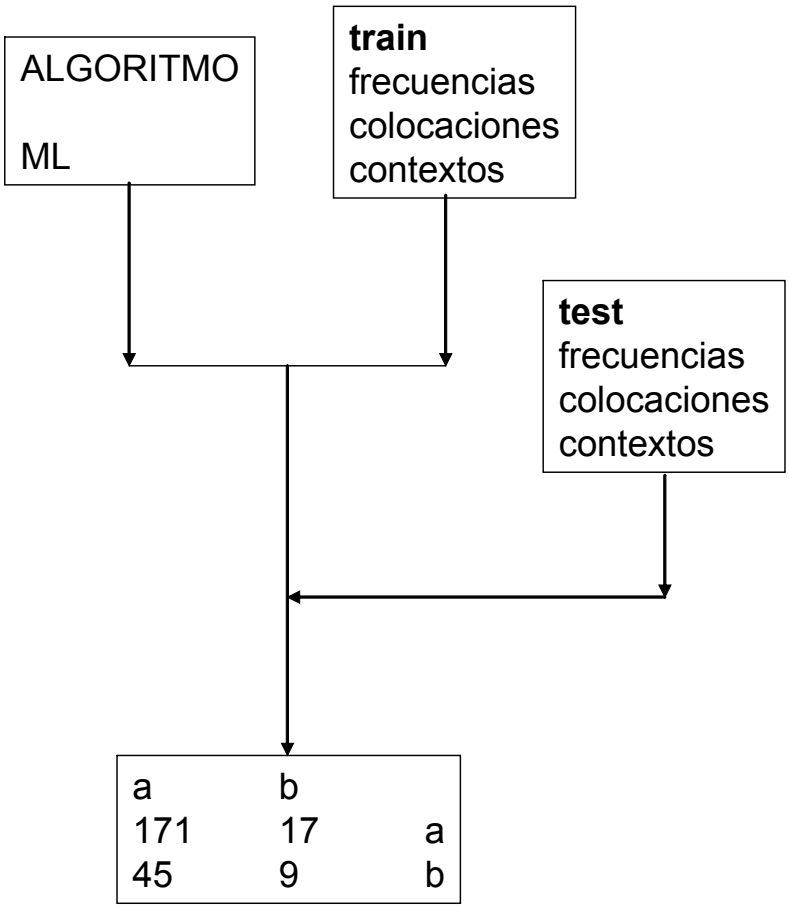


Figura 7.2 Muestra el esquema de funcionamiento de cada experimento de DSP supervisado utilizado para medir la ambigüedad de una muestra de un corpus. El fichero de entrenamiento representa la frecuencia de cada colocación en los contextos de las apariciones de la palabra objetivo usadas como entrenamiento del sistema de DSP supervisado. Además, a este sistema se le empotra un algoritmo de aprendizaje automático (ML) determinado. Una vez entrenado el sistema, se le suministra un fichero de prueba (test) que también está representado por las frecuencias de las colocaciones de los contextos de las ocurrencias de la palabra objetivo utilizadas como prueba. El seultado es una matriz cuadrada de cuatro enteros que indica el número de apariciones de cada sentido de la palabra objetivo en el fichero de prueba (sentidos *a* y *b*) que han sido clasificadas como sentidos *a* y *b*, es decir, correcta o incorrectamente. A partir de esta matriz el cálculo de la precisión del experimento es inmediato.

Finalmente, los ficheros de salida en formato *arff*, salida de *xml2arff.pl*, sirven de entrada al programa *WekaClassify*, que clasifica las ocurrencias (contextos) de la palabra objetivo de acuerdo con la representación de un 'modelo' aprendido por el sistema Weka de los ficheros de entrenamiento, también en formato *arff*.

El clasificador entrenado para construir ese modelo es una entrada de *WekaClassify*, y en nuestro caso ha sido cada vez un algoritmo de Aprendizaje Automático (AA) diferente: una red neuronal RBF, una red Bayesiana, una Tabla de Decisión y un Árbol de Decisión J48.

Por lo tanto, las entradas de *WekaClassify* son dos ficheros *arff*, uno de entrenamiento y otro de test, y un algoritmo de aprendizaje automático; y la salida es una matriz cuadrada de cuatro elementos, en la que aparece el número de ocurrencias de cada sentido clasificadas como ese sentido (clasificación correcta) o como el otro (clasificación errónea). A partir de esa matriz el sistema puede calcular e imprimir el recall del sentido mayoritario, el recall del sentido minoritario y la precisión total para un recall del 100%.

A partir de esos resultados se ha calculado la precisión media de los cuatro algoritmos, como medida de la ambigüedad de una muestra determinada de los corpus, algunas veces también denominada ambigüedad simple o autoambigüedad.

Para calcular las ambigüedades cruzadas se utiliza el fichero de entrenamiento de una muestra 'origen' y el fichero de test de otra muestra 'destino' diferente. Nótese que esta medida no es conmutativa. Finalmente la ambigüedad compuesta se calcula yuxtaponiendo los ficheros de entrenamiento de dos o más muestras y los ficheros de test de las muestras correspondientes. Nótese que en todas las medidas siempre se respeta el hecho de que debe haber el mismo número de ejemplos o contextos de entrenamiento para el sentido minoritario y para el sentido mayoritario.

7.2. Sistema utilizado para la evaluación del algoritmo de Yarowsky

El sistema que se ha utilizado para evaluar el algoritmo de Yarowsky es una implementación del algoritmo fiel a la descripción que se hace de él en el capítulo 5. La única diferencia consiste en que no se ha implementado la aplicación de la restricción OSPD (*one-sense-per-discourse*, véanse las secciones 5.2 y 5.4) en ninguna parte del programa, es decir, ni al final de cada iteración ni al final de todo el algoritmo.

Se ha aplicado una etapa previa de preprocesamiento, en la que se ha obtenido el contexto de cada ocurrencia de la palabra objetivo en el corpus. Este contexto está formado por 21 palabras a la izquierda y otras tantas a la derecha de cada ocurrencia. Para ello se ha utilizado un sencillo programa que utiliza dos colas auxiliares, una para cada lado, y recorre el corpus de izquierda a derecha, imprimiendo sobre un fichero de salida el contenido de ambas colas cada vez que encuentra la palabra objetivo.

Una vez obtenido este fichero preliminar con los contextos de la palabra objetivo, la primera fase del algoritmo consiste en determinar y etiquetar las semillas iniciales.

Como se ha descrito en el capítulo 5, existen varias formas de llevar a cabo esta tarea, desde procedimientos totalmente manuales hasta otros automáticos. El método seguido en este trabajo es el denominado de 'dos palabras'; que consiste en determinar dos palabras que suelen ser dos colocaciones muy significativas de cada uno de los dos sentidos de la palabra objetivo y anotar todos los contextos del fichero que contengan una de esas dos palabras, con su etiqueta correspondiente. Aunque este procedimiento no sea el más eficaz, el objetivo no es la máxima precisión, sino la comparación, y además se dispone de resultados de precisión obtenidos con este método en el artículo original de Yarowsky.

Figura 7.3 Muestra el etiquetado de las semillas de un fichero que representa un documento o una muestra de corpus por el método llamado de *dos palabras* [Yarowsky, 1995]. Las dos palabras son dos colocaciones muy representativas de cada sentido de la palabra objetivo y cada línea del fichero representa el contexto de una aparición u ocurrencia de la palabra objetivo en el corpus. El sistema etiqueta cada línea con el código de un sentido si la palabra correspondiente a ese sentido se encuentra en la línea o con un código especial (*-1*) si la línea no contiene ninguna de las dos palabras.

Un programa sencillo puede llevar a cabo este método; se le suministra como entrada las dos palabras, y recorre todas las líneas del fichero, que representan un contexto cada una; lee cada palabra de la línea y la compara con las dos palabras; si alguna palabra de la línea coincide con alguna de las dos palabras, imprime toda la línea en un fichero de salida, junto con la etiqueta correspondiente.

Es evidente que el éxito de este método de determinación de las semillas depende en gran medida de una elección adecuada de las dos palabras. Como ya se ha indicado, este método no es el más eficaz, ni tampoco el objetivo de este trabajo es una aplicación

práctica, por tanto lo que se ha pretendido es el autoarranque exitoso del algoritmo con una precisión lo más alta posible para este método, pero no una situación real. Esto quiere decir que se han elegido varias parejas de palabras hasta llegar a una de ellas que ha dado una precisión máxima comparable a la documentada por Yarowsky para este mismo método de elección de semillas.

El autoarranque del algoritmo una vez elegidos los contextos semilla es llevado a cabo por un programa que toma como entrada el fichero de contextos, de los cuales una parte (las semillas) están etiquetados con uno de los códigos de los dos sentidos, y ejecuta durante varias iteraciones los dos pasos básicos del algoritmo.

El primer paso lo realiza una función con encabezamiento *void paso1(FILE *f, lista *l);* que entrena una lista de decisión nueva con todos los ejemplos etiquetados hasta el momento (sólo las semillas en la primera iteración). Para ello recorre el fichero, e incluye en la lista de decisión las palabras que están en contextos etiquetados, incluyendo la información de la colocación. Esta información incluye el sentido de la colocación, su frecuencia y el tipo de colocación. Se distinguen tres tipos de colocación: palabra inmediatamente a la izquierda de la palabra objetivo, palabra inmediatamente a la derecha de la palabra objetivo y palabra en cualquier otra posición del contexto.

La lista *l* es dinámica y se procesa de izquierda a derecha al final de la función *paso1* por la función *void calculo_log(*l)* que calcula la *similitud logarítmica* (en inglés *log likelihood*) de cada colocación respecto a los dos sentidos. Para el caso en que la frecuencia de la colocación para uno de los dos sentidos sea 0, se incluye un factor de corrección llamado *alfa*, al que se le ha dado un valor empírico de 0.01.

Una vez creada y entrenada la lista de decisión, se aplica el clasificador que representa a todo el fichero (corpus) mediante la función *void paso2(FILE *f, FILE *g, lista *l, lista *l2, double umbral);*. Para ello se recorre todo el fichero de entrada **f*, se lee cada contexto y se introducen sus colocaciones, distinguiendo sus tres tipos, en la lista **l2*. A continuación se recorre esta lista y se calcula su colocación con similitud logarítmica máxima en la lista de decisión **l*. El sentido que tenga la colocación con similitud máxima será el asignado a ese contexto, y el número dado por esa similitud será la puntuación o marca lograda por ese sentido para este contexto. Además, esa puntuacón debe lograr un valor mínimo dado por *umbral*. Si ese umbral no se sobrepasa, no se decide por ninguno de los dos sentidos y el contexto se marca con un código especial que indique que su sentido todavía no se ha decidido. Véase el algoritmo de Yarowsky descrito en el capítulo 5 para comprobar que esta función sirve para aplicar el clasificador que representa la lista de decisión **l*.

El número típico de iteraciones que se han necesitado para autoarrancar el algoritmo de forma óptima ha sido habitualmente de 3. Los umbrales han sido normalmente uno bajo para la primera iteración, otro bastante más alto para la segunda y un tercer umbral igual al primero. Unos valores de ejemplo típicos de los umbrales habrían sido 3.0, 7.5 y 3.0.

El tiempo de ejecución total del autoarranque no ha sido nunca muy alto; de entre dos y cuatro minutos para los ejemplos del BNC y del corpus WSJ dados en las tablas, utilizando un PC a 1.6 GHz y 1014 MB de memoria principal bajo Windows.

7.3. Sistema utilizado para la implementación y evaluación del nuevo algoritmo de autoarranque

El sistema para la realización del nuevo algoritmo de autoarranque utiliza gran parte del programa de la sección anterior como método de autoarranque de cada documento individual del BNC. Las funciones *paso1* y *paso2* de ese programa se invocan dentro de

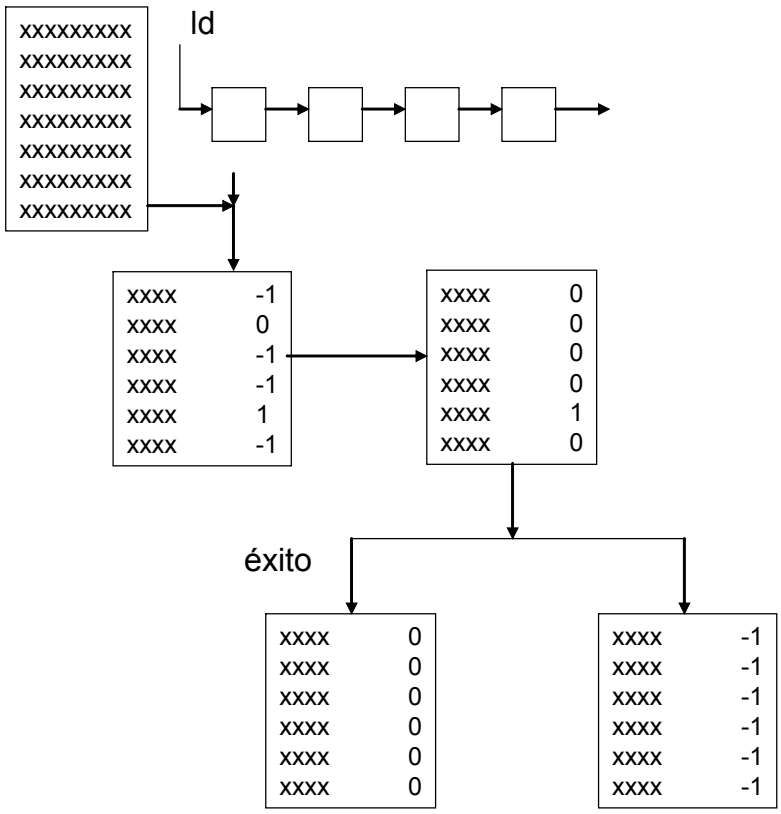


Figura 7.4 .Muestra primero el proceso de determinación automático de las dos semillas de cada sentido en un documento a partir de la lista de decisión *ld* generada en un autoarranque previo en algún otro documento; después, el autoarranque de ese documento utilizando el algoritmo de Yarowsky; y finalmente, la decisión binaria sobre el éxito del autoarranque y posterior etiquetado de todos los contextos del archivo con el código del tema homográfico de éste en caso de éxito o con un código especial (*-1*), en caso contrario.

la funcion *void funcion(char *ficheroff, char * ficherogg, int count)*. Esta función recibe como fichero de entrada un documento del BNC (es decir, un fichero con los contextos de la palabra objetivo de un documento dado) en el que dos contextos están etiquetados cada uno con uno de los dos sentidos diferentes de la palabra objetivo; y el argumento *count* representa el tamaño de ese fichero en número de contextos. Este argumento sirve

para ecualizar el umbral de la función *paso2* según el tamaño del fichero. Los umbrales antes de la ecualización son los mismos que se han indicado en la sección anterior y el factor de corrección es 0.80 si el número de contextos es mayor que 50, 0.75 si está entre 20 y 49 y 0.60 si está entre 2 y 19. El fichero de salida producido es el documento autoarrancado, que deberá pasar el criterio de decisión binario sobre el éxito o fracaso del proceso.

La forma de generar el fichero de entrada, que contiene dos contextos semilla etiquetados opuestamente, depende de si ya se dispone o no de una lista de decisión previamente creada; en otras palabras, si se trata del primer autoarranque de todo el proceso o no. Si se trata del primer autoarranque, entonces no se dispone de lista de decisión previa; por lo tanto el etiquetado de los dos contextos debe ser manual, como se ha indicado en la sección 10.2; además se crea un fichero 'artificial' formado por dos ficheros de temas homográficos opuestos; y se chequea el éxito del autoarranque contando el número de contextos de cada tema en cada fichero. Si este chequeo es positivo, ya se dispone de una lista de decisión, que se puede utilizar posteriormente sobre otros documentos. En este caso, la generación del fichero de entrada de *funcion* se puede hacer automáticamente: lo hace la función *void bootstrap(char *ficheroff, char * ficherogg, lista *listaord)*. Esta función recibe como entrada un documento sin autoarrancar y, utilizando la lista de decisión *listaord*, etiqueta *sólo* dos contextos de ese documento con sentidos opuestos. Por supuesto intentará utilizar primero las colocaciones de la lista con mayor puntuación de similitud logarítmica con cada sentido.

Una vez que *funcion* ha intentado autoarrancar un documento, *void procesar_fichero (char *fichero, int *sentido, double *u3);* aplica la decisión sobre el éxito o no del autoarranque. Esta función simplemente aplica la fórmula de la sección 10.2.1 y devuelve un sentido y un valor real como parámetros de salida. Este valor real se compara con un umbral predeterminado empíricamente y si se supera se considera que el autoarranque ha tenido éxito. El umbral utilizado ha sido inicialmente de 5.5, pero se ha ido incrementando con el número de iteraciones o vueltas alrededor de todos los documentos del corpus: ha tenido un valor de 6.0 entre la segunda y la novena vuelta y de 6.8 para las iteraciones posteriores a la décima inclusive.

Una vez que se ha decido con éxito el tema homográfico de un documento, se marcan todos sus contextos con el código de ese tema y se le incluye en una lista de documentos ya decididos. Además, la lista de decisión que se ha generado como consecuencia de su autoarranque exitoso se trunca a sus tres primeros elementos, es decir, sus tres colocaciones más significativas, y se encola en la cola correspondiente al tema holográfico del documento. Esto quiere decir que la lista de decisión que ya se ha aplicado a este documento se seguirá aplicando a todos los documentos aún no decididos, y la lista de decisión generada en este documento se desencolará eventualmente de esa cola, se combinará con todas las listas de tres elementos de la cola del otro sentido y se intentará aplicar a todos los documentos que en ese momento sigan sin decidirse, siguiendo el algoritmo de la Figura 10.3.

El número de iteraciones a lo largo de todos los documentos aún no decididos que se aplican típicamente durante un proceso de autoarranque de este algoritmo oscila entre unas 5, en pocos casos, y hasta unas 20 en la mayoría de los casos. Normalmente se logra un *recall* sobre todo el corpus de alrededor de un 10%, con una precisión muy alta, en torno al 95%, en cada autoarranque. En los tres homógrafos que se han utilizado se ha necesitado un total de cuatro autoarranques, exceptuando quizás el homógrafo *plant*, que obtiene los mismos resultados prácticamente desde el primer autoarranque. El tiempo de ejecución de cada autoarranque ha sido relativamente bajo, normalmente de unos 5 a 15 minutos, sobre la misma máquina y sistema operativo que el sistema de prueba del algoritmo de Yarowsky.

Capítulo 8. La dependencia con el dominio del algoritmo de Yarowsky

Si en el capítulo 5 se ha señalado a las colocaciones como la principal fuente de conocimiento lingüístico que emplea el algoritmo de Yarowsky, y en el capítulo 6 se ha mostrado que las colocaciones varían junto con las variaciones de dominio de los corpus, la cuestión que surge lógicamente es si el propio algoritmo de Yarowsky también sufre las consecuencias de los cambios de dominio de los corpus. Esta pregunta es la que se plantea en este capítulo y como se verá la respuesta es afirmativa: el algoritmo de Yarowsky se ve afectado por las fluctuaciones de dominio y además *en principio* estas fluctuaciones parece que perjudican a su eficacia medida como la precisión con la que desambigua las palabras.

En la sección 2.1 y en el capítulo 5 se ha revisado el tipo de Conocimiento Lingüístico (CL) que utiliza el algoritmo de Yarowsky: las colocaciones (*KS 3*). Las colocaciones se definen como "cualquier ocurrencia conjunta estadísticamente significativa de palabras" [Sag et al. 2002]. Se podría incluso emplear la palabra *coocurrencia* de dos o más palabras: cuando dos o más palabras *coocurren*, es decir, ocurren conjuntamente de una forma estadísticamente significativa, están almacenando cierta cantidad de conocimiento lingüístico que puede ser explotado por diferentes aplicaciones de PLN.

Las colocaciones tal como las acabamos de definir de una forma teórica, se tienen que poder codificar de alguna forma concreta por parte de los algoritmos prácticos. Esto es lo que hacen los llamados *rasgos* (sección 2.1.2). El tipo de rasgo que codifica las colocaciones se denomina *patrones locales* (sección 2.1.2). Los patrones locales están dotados de un alcance y un contenido. En el caso de la codificación del algoritmo de Yarowsky el alcance consiste en *n* palabras a la izquierda y *n* palabras a la derecha de la palabra objetivo; el contenido es la palabra textual completa (frente a otras posibilidades como lemas de palabras, categorías gramaticales de palabras, etc.). Es decir, el algoritmo de Yarowsky codifica el conocimiento lingüístico existente en las colocaciones utilizando una *ventana* de *+/-n* palabras completas (a veces llamado

contexto o, en la literatura anglosajona, *concordance*) alrededor de la palabra objetivo. Es evidente el uso de un corpus de texto en el que se encuentran diversas ocurrencias de la palabra objetivo que se intenta desambiguar.

La posibilidad de aplicar los patrones locales para extraer el conocimiento lingüístico almacenado en las colocaciones y así desambiguar el sentido de la palabra objetivo fue divisada y cuantificada por el propio David Yarowsky en 1993 [Yarowsky 1993]. Esa posibilidad fue tratada en ese artículo como la propiedad del lenguaje escrito denominada *un sentido por colocación* (*one-sense-per-collocation*). Esta propiedad consiste en que las palabras tienden a exhibir un único sentido para una determinada colocación. O bien, si en el contexto definido por los patrones locales de la palabra objetivo figura una colocación, es decir, se produce una ocurrencia de determinada palabra, el sentido que la palabra objetivo signifique en ese contexto tenderá a ser el mismo que en todos los demás contextos donde se produzca esa misma colocación, es decir, donde haya una ocurrencia de dicha palabra.

En su artículo de 1993 Yarowsy cuantificó la validez de la propiedad *un sentido por colocación*. Lo hizo probando que la entropía de la distribución de probabilidades condicionales de las colocaciones y la eficacia de un algoritmo supervisado que utiliza un algoritmo de Aprendizaje Automático (AA) entrenado con esas colocaciones son dos medidas que están muy relacionadas, y obteniendo una *precisión* de un 92% y un *recall* de un 98%. Con este resultado confirmó la hipótesis propuesta por el enunciado de la propiedad (véase sección 6.1 para una descripción pormenorizada de los experimentos llevados a cabo por Yarowsky).

En 2000 Martinez y Agirre [Martinez y Agirre 2000] llevaron a cabo una serie de experimentos encaminados a comprobar si la propiedad *un sentido por colocación* probada por Yarowsky seguía o no cumpliéndose en un corpus sometido a cambios o fluctuaciones de dominio.
En sus experimentos usaron el corpus DSO [Ng y Lee 1996] que está compuesto por parte del Brown Corpus [Francis y Kucera 1964] y parte del corpus Wall Street Journal (WSJ) y está etiquetado manualmente con sentidos para una serie de palabras objetivo comunes del idioma inglés, en total 191 palabras polisémicas frecuentes. Este etiquetado de sentidos usa una granularidad relativamente fina, ya que utiliza los sentidos que aparecen en WordNet [Miller et al. 1993] para dichas palabras objetivo. Recordemos que el número de sentidos medio en WordNet es mayor que 5, lo cual representa una granularidad bastante fina, y desde luego mucho más fina que la granularidad representada por la polisemia de dos sentidos utilizada por Yarowsky en casi todos sus experimentos. Recordemos también que este hecho ha sido muy criticado, llegando incluso a llamarlo el "problema de WordNet" y que muchos investigadores actualmente opinan que el número de sentidos de las palabras polisémicas que necesitan las aplicaciones reales de PLN es de sólo dos (véase el capítulo 4).

Martinez y Agirre utilizaron el mismo algoritmo supervisado que usó Yarowsky en 1993, pero naturalmente tuvieron que adaptar el algoritmo de AA (Lista de Decisión) al caso general de ambigüedad de más de dos sentidos.

En la primera parte de sus experimentos aplicaron el algoritmo supervisado a las dos partes del DSO, el BC y el WSJ, y observaron que los resultados fueron mejores para el segundo que para el primero. Este hecho lo interpretaron como indicio de que las colocaciones sí dependen del dominio, ya que el BC es un corpus equilibrado, con variaciones de género y tópico, cosa que no ocurre con el WSJ.

Este experimento se repitió garantizando que los ejemplos de entrenamiento y *test* en cada corpus nunca perteneciesen a los mismos documentos del corpus, algo que podría afectar a la precisión. El resultado fue que la precisión bajó en el BC y siguió igual en el WSJ, lo que se interpretó también como debido a las variaciones de dominio entre documentos del BC.

Finalmente probaron que entrenando en el WSJ y probando en el BC los mejores resultados se obtenían en la categoría de dominio del BC más similar al WSJ (categoría 'Press: Reportage').

Por tanto, si las colocaciones dependen de los cambios de dominio en los corpus, y si las colocaciones son la principal fuente de conocimiento lingüístico que utiliza el algoritmo de Yarowsky, la cuestión que surge lógicamente es si el algoritmo de Yarowsky también depende o no de los cambios de dominio de los corpus.

Dominio	*Textos*	*Palabras*	*%*
Applied science	370	7 104 635	8.14
Arts	261	6 520 634	7.47
Belief and thought	146	3 007 244	3.44
Commerce, finance	295	7 257 542	8.31
Imaginative	477	16 377 726	18.76
Leisure	438	12 187 946	13.96
Natural science	146	3 784 273	4.33
Social science	527	13 906 182	15.93
World affairs	484	17 132 023	19.62

Tabla 8.1. Categorías de dominio utilizadas en el corpus equilibrado British National Corpus (BNC), junto con el número de documentos, palabras y porcentaje de éstas en cada una.

Medio	*Textos*	*Palabras*	*%*
Book	1414	49 891 770	57.16
Periodical	1208	28 356 005	32.48
Published misc.	238	4 197 450	4.80
Unpublished misc.	249	3 508 500	4.01
To-be-spoken	35	1 324 480	1.51

Tabla 8.2. Categorías de medio utilizadas en el BNC. Nótese que los medios *libro* y *publicación periódica* representan juntos casi el 90% de las palabras de todo el corpus.

En [Sánchez de Madariaga y Fernández del Castillo 2008] se llevan a cabo una serie de experimentos encaminados a contestar esa pregunta. Para ello aplica el algoritmo de Yarowsky sobre dos corpus de naturaleza muy distinta como son el British National Corpus (BNC) y el corpus Wall Street Journal (WSJ). El primero figura como un ejemplo de corpus de texto general y equilibrado, en el que hay variaciones de dominio y género, y en el que esas variaciones están tratadas explícitamente, es decir, sus documentos están clasificados según esos dominios y géneros. En las Tablas 8.1 y 8.2 se muestran las categorías de dominios y medios utilizados en la clasificación de los documentos de ese corpus. El segundo figura como un corpus periodístico en el que no hay variaciones de dominio o al menos sus documentos no están clasificados según ese criterio.
Dado que el BNC es un corpus formado por textos escritos en inglés británico y el WSJ está compuesto por artículos escritos en inglés americano, se eligieron como palabras objetivo tres de los doce homógrafos utilizados originalmente por Yarowsky en su artículo de 1995 que funcionan normalmente como homógrafos en ambas formas del idioma inglés. El hecho de que sean parte de los doce homógrafos de Yarowsky permite comparar directamente los resultados de [Sánchez de Madariaga y Fernández del Castillo 2008] y [Yarowsky 1995].

Los resultados de la Tabla 8.3 al aplicar el algoritmo a un corpus sin variaciones de dominio como el WSJ coinciden casi milimétricamente con los resultados aportados por Yarowsky en 1995 (Tabla 5.2). Hay que tener en cuenta que en estos resultados se aplicó uno de los métodos de autoarranque más simples, el de 'dos palabras' (véase el capítulo 5), y que tampoco se ejecutó la fase del algoritmo en la que se aplica la propiedad *un sentido por discurso* (*one-sense-per-discourse*), lo cual hubiera aumentado la precisión alrededor de un 4% en todos los casos. De todas formas en la Tabla 5.2 se incluyen los resultados para ambas posibilidades.

Sin embargo, los resultados de los experimentos equivalentes sobre un corpus con variaciones de dominio como el BNC que se pueden ver en la Tabla 8.4 muestran que en este caso la precisión media es mucho menor, pasando del 91.2% al 74.7% de media. Teniendo en cuenta el 4% adicional por utilizar la propiedad *un sentido por discurso*, pasaríamos de un 95.2% a un 78.7%, es decir, pasaríamos del alrededor del 95% aceptable en aplicaciones reales de PLN a unos niveles claramente insuficientes.

1	2	3	4	5
palabra	**sentidos**	**tamaño**	**base**	**dos palabras**
drug	medicina/narcótico	2498	55.4	91.6
plant	vegetal/factoría	1511	83.6	91.8
space	volumen/exterior	623	59.0	90.2
MEDIA			66.0	91.2

Tabla 8.3. Porcentaje del sentido más frecuente y resultados de precisión al aplicar el algoritmo de Yarowsky a un corpus no equilibrado y de texto periodístico como es el WSJ. Se utilizó como método de autoarranque el de *dos palabras*, que es uno de los más simples (véase el capítulo 5). Se usaron tres homógrafos diferentes como palabras a desambiguar. Las tres palabras funcionan como homógrafos significativos tanto en inglés norteamericano (corpus WSJ) como en inglés británico (BNC).

1 **palabra**	2 **sentidos**	3 **tamaño**	4 **base**	5 **dos palabras**
drug	medicina/narcótico	1013	62.5	66.7
plant	vegetal/factoría	2676	66.3	87.2
space	volumen/exterior	4807	50.7	70.3
MEDIA			59.8	74.7

Tabla 8.4. Resultados de precisión del mismo experimento que el de la Tabla 8.3, pero aplicado al corpus equilibrado y de texto general BNC.

Por lo tanto, estos resultados demuestran que las variaciones de dominio presentes en corpus de texto general afectan negativamente y de forma considerable a la eficiencia del algoritmo de Yarowsky, frente a las posibles variaciones de dominio de corpus de texto periodístico, pasando de niveles óptimos a niveles no aceptables desde un punto de vista de las aplicaciones reales de PLN. En el capítulo 9 se estudia la distribución estadística de las fuentes de dominio en corpus de texto general y en corpus de texto periodístico y se sugiere una posible explicación a la precisión más alta lograda en los corpus periodísticos por los algoritmos de DSP. En el capítulo 10 se presenta una nueva metodología de autoarranque del algoritmo de Yarowsky que sirve para incrementar considerablemente su precisión en corpus con variaciones de dominio como las de los corpus de texto general.

Capítulo 9.
La distribución estadística del dominio en los corpus de texto

En el capítulo 8 se ha mostrado que, como consecuencia de la dependencia de las colocaciones del dominio descritas en el capítulo 6, y como consecuencia de la utilización de las colocaciones como principal fuente de conocimiento lingüístico por parte del algoritmo de Yarowsky descrita en los capítulos 2 y 5, a su vez este algoritmo depende de alguna forma del dominio. En dicho capítulo se presentaron resultados sobre la precisión del algoritmo de Yarowsky aplicado sobre dos tipos de corpus de texto de diferente naturaleza, a saber, un corpus periodístico habitual como es el corpus Wall Street Journal (WSJ) y un corpus de texto general equilibrado como es el corpus British National Corpus (BNC) [Sánchez de Madariaga y Fernández del Castillo 2008].

Los resultados de ese capítulo muestran cómo la precisión para el corpus WSJ coincide casi exactamente con los resultados aportados por Yarowsky en su trabajo original de 1995, mientras que la precisión para el BNC se degrada considerablemente, llegando a niveles inaceptables para las aplicaciones reales de PLN que utilizan DSP.

En este capítulo se presentan resultados del estudio de la distribución estadística de las fuentes de dominio en los corpus BNC y WSJ [Sánchez de Madariaga, Paice, Rayson y Fernández del Castillo, 2008], que justifican las causas de los resultados excepcionalmente altos de precisión de los algoritmos de DSP sobre corpus periodísticos como el WSJ.

Asímismo, en este capítulo se investiga la naturaleza de la distribución estadística de las *fuentes de dominio* en esos dos tipos de corpus de naturaleza a priori distinta, como medio para tratar de explicar esos resultados diferentes obtenidos tras la aplicación del algoritmo de Yarowsky.

Como medio para explorar la distribución del dominio en los corpus se ha utilizado una *medida de la ambigüedad* de muestras extraídas de los corpus. Esta medida de la ambigüedad se ha obtenido básicamente a través de la aplicación de experimentos de DSP supervisados sobre esas muestras. Se han aplicado medidas de *ambigüedad simple*, o *autoambigüedad*, *ambigüedad cruzada* entre muestras y *ambigüedad compuesta* sobre composición de las muestras.

9.1 Una medida de la ambigüedad de muestras de texto

Como no se dispone de una definición exacta de *ambigüedad del sentido de las palabras* o de la *ambigüedad de una muestra de texto* ni de un procedimiento preciso para medirlas, se ha recurrido a llevar a cabo una serie de experimentos de DSP supervisados de naturaleza muy variada sobre una muestra de texto dada como *medida de la ambigüedad* de dicha muestra.

La expresión 'de naturaleza muy variada' se refiere aquí a la naturaleza del algoritmo de Aprendizaje Automático, AA (en inglés, Machine Learning, ML) utilizado en cada experimento de DSP supervisado. El algoritmo de AA que se utiliza en un método de DSP supervisado es independiente del método de DSP supervisado en sí. Lo mismo ocurre en un método de DSP semisupervisado como es el algoritmo de Yarowsky: en su versión original, este algoritmo de AA era una Lista de Decisión, pero se podría haber utilizado cualquier otro. Esto quiere decir que, en un método de DSP supervisado, el algoritmo de AA utilizado influye en el resultado, medido en precisión, del experimento. En este capítulo nuestro objetivo del estudio no es una alta marca de precisión en los experimentos de DSP, sino medir de alguna forma el nivel de ambigüedad del subconjunto o muestra del corpus sobre el que se aplican los experimentos. Por lo tanto, no vamos a buscar el algoritmo de AA que optimice estos experimentos, sino a aplicar una gama lo más variada posible de estos tipos de algoritmos de AA, de forma que calculando la media aritmética de las precisiones obtenidas con cada uno de ellos logremos una *descripción* lo más completa posible de su ambigüedad.

Los algoritmos de AA son muy variados y pertenecen a diversas familias de naturaleza muy diversa. Se han elegido cuatro algoritmos representantes de cuatro de estas familias, intentando cubrir un espectro lo más amplio posible, aunque dentro de la simplicidad, que son: Red RBF (RBF Network) dentro de la familia de las redes neuronales, Red de Bayes (BayesNet) dentro de la familia de los algoritmos bayesianos, Tablas de Decisión (Decision Table) dentro de la familia de algoritmos basados en reglas, y Árbol J48 (J48 Tree) dentro de los algoritmos basados en árboles.

Por lo tanto, nuestra medida básica de la ambigüedad de una muestra de texto se va a obtener aplicando cuatro experimentos de DSP, cada uno con uno de estos algoritmos de AA, y calculando la media aritmética de la precisión de cada experimento.

Se debe hacer hincapié en que esta medida es completamente relativa, ya que depende de factores tales como el tamaño del corpus, el tamaño del conjunto de entrenamiento del método supervisado, y del algoritmo de AA que se utilice. Por lo tanto, no pretende ser una medida absoluta, sino que se utiliza de una forma relativa como medio para explorar la distribución de las *fuentes de dominio* de los corpus bajo estudio.

9.2 Muestras de los corpus bajo estudio

Como se ha indicado al principio de este capítulo, se ha estudiado la distribución estadística de las fuentes de dominio en corpus periodísticos y en corpus de texto general. Como representantes de ambos tipos de corpus se han utilizado el corpus WSJ y el BNC, respectivamente. En esta sección se describen las muestras extraídas de ambos corpus que se han utilizado para desarrollar el trabajo de campo que se ha desarrollado durante la investigación.

Al igual que se ha descrito en la sección anterior, la medida básica utilizada ha sido la *ambigüedad* de muestras de texto obtenida como resultado de experimentos de DSP supervisados sobre esas muestras. Este tipo de experimentos utiliza como *entrada* una serie de ocurrencias de la *palabra objetivo* u *homógrafo* en dicha muestra o subconjunto del corpus junto con su *contexto*. El contexto suele consistir en una *ventana* de +/- *k* palabras alrededor de la ocurrencia concreta de la palabra objetivo[20]. Naturalmente el contexto sirve como fuente de información para decidir a cuál de los dos posibles sentidos del homógrafo corresponde la ocurrencia concreta en el corpus de éste. En un método de DSP supervisado normalmente hay una fase de entrenamiento en la que el sistema recibe información sobre contextos etiquetados manualmente a priori (conjunto de entrenamiento) y con esa información y el contexto de una ocurrencia *nueva* debe decidir sobre el sentido a que se refiere esta ocurrencia.

Todo esto significa que dado un corpus completo cualquiera, la muestra de ese corpus que se vaya a estudiar, que normalmente será un subconjunto de él, depende además de la *palabra objetivo* u homógrafo considerado. Evidentemente los contextos de una palabra objetivo dada no van a ser iguales que los de otra cualquiera. A partir de ahora nos referiremos a una pareja corpus-homógrafo de la siguiente forma: estudiaremos básicamente cuatro casos, a saber, BNC/space, WSJ/space, BNC/drug y WSJ/drug. Es decir, nos centraremos en el estudio de los homógrafos *space* y *drug* en los corpus BNC y WSJ. Los sentidos de ambos homógrafos son: *sitio* o *volumen* frente a *espacio exterior* para el primero y *medicina* frente a *narcótico* para el segundo. La elección de estas dos palabras objetivo estuvo motivada por el hecho de que ambas funcionan como homógrafos tanto en inglés británico como en inglés americano. Hay que tener en

[20] A este contexto se le llama frecuentemente *concordance* en la literatura anglosajona.

cuenta que el BNC es un corpus formado por textos escritos en inglés británico, mientras que el WSJ está escrito en inglés americano.

Antes de describir las muestras del BNC y del WSJ realmente utilizadas en los experimentos es conveniente hacer un breve repaso de las principales características de ambos corpus.

9.2.1 Los corpus BNC y WSJ

El corpus WSJ está compuesto por aproximadamente 30 millones de palabras en 98 732 artículos aparecidos en el diario norteamericano The Wall Streer Journal durante los años 1987, 1988 y 1989. El corpus ha sido desarrollado por el Brown Laboratory for Linguistic Information Processing (BLLIP) de la Universidad de Brown (EE UU). El corpus no realiza ningún tipo de clasificación sobre el contenido u otro aspecto de los artículos, que aparecen en orden cronológico desde el mes de enero de 1987 en adelante.

El BNC es un corpus de propósito general equilibrado formado por texto en inglés británico a partir de 4054 muestras o 'documentos' de hasta 45000 palabras cada una con un total de más de 100 millones de palabras en unos 6 millones de oraciones. Ha sido desarrollado por un consorcio académico-industrial del Reino Unido que incluye el Longman Group, la British Library y las Universidades de Oxford y Lancaster, entre otros. El BNC es un corpus equilibrado. Esto quiere decir que los documentos se escogieron siguiendo criterios de dominio (Ciencias Naturales, Ciencias Sociales, Ciencias Aplicadas, Arte, etc.), medio (libro, revista, etc.) y época (entre determinadas fechas), y que se asignaron porcentajes de cuota para cada clase. Las Tablas 8.1 y 8.2 muestran las categorías de dominio y medio que se utilizaron para la clasificación de todos los documentos del corpus. Dentro de la categoría *libro* del criterio *medio* la mitad de los documentos o muestras se obtuvieron de forma aleatoria del catálogo *Books in Print 1992* de la editorial Whitaker. La clasificación en dominios y medios sirve para que, por ejemplo, podamos seleccionar todos los documentos del dominio *ciencias sociales* y del medio *libro*, por ejemplo, o bien, del dominio *ciencias sociales* y del medio *revista*; evidentemente, estos dos subconjuntos del corpus estarían compuestos por diferentes documentos del BNC, es decir, no contendrían ningún documento en común.

9.2.2 La muestras utilizadas en los experimentos

Para el análisis de la distribución de las fuentes de dominio en el BNC se siguieron criterios muy diferentes de los seguidos para el análisis del WSJ. Esta diferencia se basa fundamentalmente en la muy distinta clasificación de los documentos en ambos corpus. Como se ha indicado en la sección anterior, los documentos del BNC están clasificados siguiendo varios criterios en paralelo, mientras que los documentos del WSJ, mucho

menos extensos, no están clasificados de ninguna manera. Esto hace que podamos identificar la muestra o subconjunto del BNC *Ciencias Sociales/libro*, por ejemplo, cosa que evidentemente no se puede lograr en el WSJ.

De esta forma, para analizar la distribución de los dominios en el BNC, se han obtenido principalmente muestras o subconjuntos del corpus atendiendo a la clasificación de sus documentos en dominios. Así por ejemplo, para el par *BNC/space*, se obtuvieron los subconjuntos o muestras de los dominios *natural Science*, *Applied Science* Y *World Affairs* y dentro del medio *book*. Los subconjuntos *Applied Science/Book* Y *World Affairs/Book* contienen, para el homógrafo *space*, unos 500 contextos, es decir, ocurrencias de la palabra objetivo, cada uno, mientras que el subconjunto *Natural Science/book* contiene aproximadamente el doble de contextos. Para lograr que todas las muestras tengan un tamaño similar, este último subconjunto ha sido dividido en dos partes iguales, de unos 500 contextos cada una. Por tanto, para el par *BNC/space* se obtuvieron cuatro muestras, de los tres dominios arriba citados. Nótese que el hecho de que haya dos muestras del tipo *Natural Science/book* no significa que haya ninguna relación especial entre los documentos de ambas muestras, excepto que han sido clasificados por los autores del BNC dentro del dominio *Natural Science* y dentro del medio *book*.

Las muestras para el par *WSJ/space* contienen aproximadamente el mismo número de contextos, unos 500 cada una, pero por supuesto no están clasificadas según dominio/medio: se obtuvieron cuatro muestras de 500 contextos agrupando artículos del WSJ consecutivamente desde el mes de enero de 1987 en adelante, hasta cubrir un total de 2 000 ocurrencias/contextos del homógrafo *space*.

Para el par *BNC/drug* se obtuvieron cinco muestras de 500 contextos y cuatro pares dominio/medio, a saber, *Applied Science/periodicals*, *Natural Science/periodicals*, *Social Science/books* y *world affairs/periodicals*. Se obtuvieron dos muestras del par *world affairs/periodicals*, de forma análoga a las muestras del homógrafo *space*. Es interesante notar que los medios *books* y *periodicals* suman alrededor del 90% de todo el contenido del BNC.

9.3 Diseño de los experimentos

Cada experimento básico de DSP se diseñó de la siguiente forma: a partir de una muestra básica del par corpus/homógrafo como las descritas en la sección anterior, compuesta por unos 500 contextos, y después de etiquetar cada contexto manualmente con uno de los dos posibles sentidos holográficos, a los que se denominan *sentido 0* y *sentido 1*, se eligieron aleatoriamente 72 contextos de cada sentido (144 en total), que se utilizaron como conjunto de entrenamiento (*training set*) del experimento supervisado. El resto de contextos etiquetados de la muestra se utilizaron como conjunto de prueba (test set) del experimento.

El hecho de haber elegido en el conjunto de entrenamiento el mismo número de contextos de cada sentido responde a la dependencia de la eficacia de los experimentos supervisados del número de ejemplos de entrenamiento de cada sentido o, dicho de otra forma, si se hubiera respetado la proporción *a priori* de contextos de cada sentido en toda la muestra en el conjunto de entrenamiento, se hubiera favorecido el nivel de precisión del sentido mayoritario, al gozar éste de más ejemplos de entrenamiento.

Como se ha adelantado antes, para cada experimento definido por una pareja de conjuntos de entrenamiento y de prueba, se realizaron en realidad cuatro experimentos diferentes, variando en cada caso el algoritmo de AA dentro del método de DSP supervisado. La naturaleza de los algoritmos de AA se eligió muy diferente entre sí, para tener una medida lo más general posible de la ambigüedad de la muestra bajo estudio.

Los cuatro algoritmos de AA que se utilizaron son los siguientes: Red Neuronal RBF, Red Bayesiana, Tablas de Decisión y Árbol J48. Para la realización concreta de estos experimentos contamos con la ayuda inapreciable de la colección *Weka* de algoritmos de AA para Minería de Datos (Data Mining) y Tareas de Clasificación disponibles en el Weka Machine Learning Project [Witten y Frank 2005] desarrollado por la Universidad de Waikato (Nueva Zelanda) como software libre totalmente escrito en Java y de disponibilidad pública.

El experimento básico de medida de la ambigüedad de una muestra de texto utiliza un conjunto de entrenamiento como el descrito en esta sección, obtenido de la propia muestra, y un conjunto de prueba obtenido asimismo de ella. Sin embargo, también se han realizado experimentos de *ambigüedad cruzada* y de *ambigüedad compuesta*.

Se han de considerar las siguientes definiciones:

- La ambigüedad cruzada se define como la (media de la) precisión del experimento de DSP supervisado utilizando como conjunto de entrenamiento el de una primera muestra y como conjunto de prueba el de otra segunda muestra.

- La ambigüedad compuesta se define como la (media de la) precisión del experimento de DSP supervisado utilizando como conjunto de entrenamiento la yuxtaposición o composición de los conjuntos de entrenamiento de dos o más muestras básicas, y como conjunto de prueba la yuxtaposición de los conjuntos de prueba de las mismas muestras.

- La precisión de los experimentos de DSP se define como el número de contextos etiquetados correctamente por el algoritmo de entre el número total de contextos etiquetados (manualmente) que forman el conjunto de prueba. Se hace además una distinción entre la precisión para el sentido 0 y la precisión para el sentido 1.

> La precisión para uno de los dos sentidos se define como el número de contextos etiquetados con ese sentido de entre el número total de contextos etiquetados manualmente a priori con ese sentido.

Por lo tanto un experimento para el cálculo de la *ambigüedad simple* o *autoambigüedad* de una muestra determinada consiste en efectuar cuatro experimentos de DSP supervisados, cada uno con uno de los cuatro algoritmos de AA, y calcular tres valores medios sobre cuatro valores resultado: un valor medio corresponde a la precisión total, otro a la precisión del sentido 0 y otro a la precisión del sentido 1.

9.4 La naturaleza diferente del BNC y del corpus WSJ

9.4.1 Ambigüedad cruzada dentro del BNC y dentro del WSJ

Para investigar la distribución estadística de las fuentes de dominio en el BNC y en el WSJ se han utilizado como muestras o subconjuntos básicos de ambos corpus los que se han indicado en la sección previa. Como se observó en esa sección, las variaciones de dominio están en cierto modo reflejadas en la propia estructura 'administrativa' de los dos corpus: mientras en el BNC los documentos están clasificados explícitamente según una serie de dominios (véase la Tabla 8.1), nada de esto ocurre en el caso del WSJ. Por supuesto, esto no quiere decir que en este corpus no haya variaciones de dominio, pero como mínimo estas variaciones no han sido hechas explícitas por parte de los compiladores del corpus.

En las Tablas 9.1, 9.2 y 9.3 se pueden ver las ambigüedades simples y cruzadas de las muestras del par *BNC/space*. La Tabla 9.1 hace referencia a las ambigüedades totales, mientras que las Tablas 9.2 y 9.3 se refieren a los resultados de precisión para cada uno de los dos sentidos homográficos de *space*. En las tres tablas los nombres de las muestras en la columna de la izquierda se refieren a las los ficheros de los conjuntos de entrenamiento, mientras que los nombres en la fila de arriba indican los ficheros de los conjuntos de prueba (test). La diagonal principal presenta los resultados de ambigüedad simple de una muestra, a veces llamada también autoambigüedad.

De forma análoga las Tablas 9.4, 9.5 y 9.6 presentan los resultados para el par *WSJ/space*, y las Tablas 9.7, 9.8 y 9.9 y las Tablas 9.10, 9.11 y 9.12 hacen referencia a los pares *BNC/drug* y *WSJ/drug*.

A partir de los resultados de la ambigüedad cruzada de la Tabla 9.1 para el par *BNC/space* se pueden construir dos grupos de muestras relacionadas según el dominio. En el primer grupo estarían incluidas las dos muestras del dominio *Natural Science* y en el otro grupo las muestras de los dominios *Applied Science* y *world affairs*. Para hacer estos grupos se ha realizado el cálculo de la ambigüedad cruzada en las dos direcciones

de entrenamiento y test para cada par de muestras y se ha utilizado un umbral de 68.0, de forma que pares con precisión media más alta pertenecen al mismo grupo y pares con precisión media menor que ese umbral pertenecen a grupos diferentes. La ambigüedad media intragrupo es de 72.2 %, mientras que la ambigüedad media intergrupo es de 62.3%. Por lo tanto hay una caída de ambigüedad cruzada media de casi un 10%, lo cual indica un cambio de dominio fuerte.

	applied	**natural_1**	**natural_2**	**world**
applied	90.6	66.7	69.3	72.9
natural_1	68.3	79.6	68.0	64.0
natural_2	61.7	68.0	82.4	58.0
world	79.9	62.3	47.5	83.2

Tabla 9.1 Ambigüedad simple y cruzada de cuatro muestras del BNC con el homógrafo *space*. Las cuatro muestras pertenecen a tres dominios diferentes y al medio *books*. La columna de la izquierda representa los dominios de las muestras de entrenamiento y la fila de arriba representan los dominios de las muestras de test. Los resultados de la diagonal principal son los de la ambigüedad simple o autoambigüedad.

	applied	**natural_1**	**natural_2**	**world**
applied	92.2	90.6	82.3	84.6
natural_1	70.4	86.3	70.3	69.9
natural_2	58.0	65.6	79.8	60.5
world	87.6	94.1	91.0	89.2

Tabla 9.2 Muestra el mismo experimento que en la Tabla 9.1, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 0* (volumen) del homógrafo *space*. Los resultados de la Tabla 9.1 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

	applied	**natural_1**	**natural_2**	**World**
applied	83.4	39.7	63.0	27.4
natural_1	58.4	72.2	67.0	41.5
natural_2	78.7	70.7	83.7	48.2
world	44.5	26.2	17.0	59.8

Tablas 9.3 Muestra el mismo experimento que en la Tabla 9.1, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 1* (espacio exterior) del homógrafo *space*. Los resultados de la Tabla 9.1 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

MUESTRA	**1**	**2**	**3**	**4**
1	84.5	80.3	79.3	83.5
2	80.4	81.2	78.3	80.0
3	79.8	83.3	85.6	79.5
4	81.6	79.0	80.0	84.9

Tabla 9.4 Ambigüedad simple y cruzada de cuatro muestras del WSJ con el homógrafo *space*. Las cuatro muestras están formadas por documentos contiguos del corpus, sin distinguir explícitamente entre dominios. La columna de la izquierda representa las muestras de entrenamiento y la fila de arriba indica las muestras de test. Los resultados de la diagonal principal son los de la ambigüedad simple o autoambigüedad.

MUESTRA	1	2	3	4
1	94.3	92.3	89.9	94.9
2	91.5	88.7	86.6	93.0
3	85.8	85.6	84.4	85.4
4	94.7	91.3	91.7	96.1

Tablas 9.5 Muestra el mismo experimento que en la Tabla 9.4, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 0* (volumen) del homógrafo *space*. Los resultados de la Tabla 9.4 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

MUESTRA	1	2	3	4
1	77.3	75.6	75.3	74.2
2	72.3	78.3	75.2	69.5
3	75.4	82.5	86.1	74.6
4	72.0	74.3	75.6	75.8

Tablas 9.6 Muestra el mismo experimento que en la Tabla 9.4, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 1* (espacio exterior) del homógrafo *space*. Los resultados de la Tabla 9.4 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

	Applied	Natural	Social	World 1	World 2
Applied	86.9	86.0	52.6	31.7	25.1
Natural	89.1	94.4	30.0	20.9	12.2
Social	68.0	74.7	79.9	52.3	48.5
World 1	32.2	41.3	77.1	84.5	89.3
World 2	49.3	48.3	68.7	68.5	82.9

Tabla 9.7 Ambigüedad simple y cruzada de cinco muestras del BNC con el homógrafo *drug*. Las cinco muestras pertenecen a cuatro dominios diferentes y a los medios *books* y *periodicals*. La columna de la izquierda representa los dominios de las muestras de entrenamiento y la fila de arriba representan los dominios de las muestras de test. Los resultados de la diagonal principal son los de la ambigüedad simple o autoambigüedad.

Un agrupamiento similar puede lograrse a partir de los resultados de la Tabla 9.7 para el par *BNC/drug*. En este caso los dos grupos estarían formados por las muestras de dominios *applied science* y *natural science* el primero y por las muestras de dominios *Social Science* y *world affairs* el segundo. La ambigüedad cruzada intragrupo es de 72.5 % y la ambigüedad cruzada intergrupo es de 40.6 %. En este caso por tanto, la caída de la ambigüedad cruzada es de más de un 30% que indica un cambio de dominio todavía más dramático que en el par *BNC/space*.

Analizando los resultados de la Tabla 9.4 para el par *WSJ/space* se podrían distinguir dos grupos formados por las muestras *1* y *4* el primero y por las muestras *2* y *3* el segundo. Sin embargo, en este caso la ambigüedad cruzada intragrupo es 81.7 % mientras que la ambigüedad cruzada intergrupo es 79.8 %. Esto quiere decir que la caída es de menos del 2 %, lo cual indica que no hay un cambio de dominio significativo.

La Tabla 9.10 para el par *WSJ/drug* arroja los siguientes resultados: los grupos estarían formados por las muestras *2* y *3* el primero y las muestras *4* y *5* el segundo; la ambigüedad cruzada intragrupo es 82.5 % y la intergrupo es 81.4 %; la caída es de 1.1%.

Estos resultados de la ambigüedad cruzada indican que la distribución de las *fuentes de dominio* en el WSJ es muy diferente de la distribución de las fuentes de dominio en el BNC. Las fuentes de dominio parecen estar concentradas en ciertas partes del BNC mientras que están distribuidas uniformemente a lo largo de todo el WSJ.

	Applied	**Natural**	**Social**	**World 1**	**World 2**
Applied	91.0	90.0	83.3	82.1	82.7
Natural	98.6	97.6	94.4	99.2	98.1
Social	67.3	70.3	78.4	70.7	62.5
World 1	21.7	27.7	25.7	52.8	25.0
World 2	47.2	44.1	40.3	46.4	69.2

Tabla 9.8 Muestra el mismo experimento que en la Tabla 9.7, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 0* (medicina) del homógrafo *drug*. Los resultados de la Tabla 9.7 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

	Applied	**Natural**	**Social**	**World 1**	**World 2**
Applied	62.1	73.8	46.5	20.8	18.1
Natural	30.6	84.5	17.0	04.0	01.9
Social	72.5	88.1	80.1	48.2	46.1
World 1	63.7	82.7	87.4	91.3	97.1
World 2	62.1	61.3	74.4	73.2	84.6

Tabla 9.9 Muestra el mismo experimento que en la Tabla 9.7, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 1* (narcótico) del homógrafo *drug*. Los resultados de la Tabla 9.7 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

MUESTRA	**2**	**3**	**4**	**5**
2	78.4	75.4	78.0	76.1
3	78.5	86.8	84.2	81.9
4	81.5	87.1	92.6	85.8
5	78.2	83.7	90.0	87.1

Tabla 9.10 Ambigüedad simple y cruzada de cuatro muestras del WSJ con el homógrafo *drug*. Las cuatro muestras están formadas por documentos contiguos del corpus, sin distinguir explícitamente entre dominios. La columna de la izquierda representa las muestras de entrenamiento y la fila de arriba indica las muestras de test. Los resultados de la diagonal principal son los de la ambigüedad simple o autoambigüedad.

MUESTRA	2	3	4	5
2	78.0	75.6	79.7	78.3
3	79.1	84.2	85.0	85.0
4	82.3	83.7	92.4	85.4
5	73.4	78.3	81.0	81.9

Tabla 9.11 Muestra el mismo experimento que en la Tabla 9.10, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 0* (medicina) del homógrafo *drug*. Los resultados de la Tabla 9.10 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

MUESTRA	2	3	4	5
2	79.5	75.3	76.9	72.4
3	77.0	90.5	83.7	76.9
4	79.6	92.1	92.8	86.4
5	90.3	91.3	95.5	95.5

Tabla 9.12 Muestra el mismo experimento que en la Tabla 9.10, pero los resultados de ambigüedad simple y cruzada corresponden al *sentido 1* (narcótico) del homógrafo *drug*. Los resultados de la Tabla 9.10 son globales, es decir, corresponden a los dos sentidos, sin distinción entre ellos.

	base		base
applied	82.2 (sentido 0)	**1**	57.9 (sentido 1)
natural_1	53.1 (sentido 0)	**2**	72.1 (sentido 1)
natural_2	67.6 (sentido 1)	**3**	72.6 (sentido 1)
world	79.6 (sentido 0)	**4**	55.0 (sentido 1)

Tabla 9.13 Porcentaje del sentido más frecuente a priori (línea de base de referencia) en las cuatro muestras del BNC y del WSJ con el homógrafo *space*. Entre paréntesis se indica cuál es el sentido más frecuente en cada caso.

	base		base
applied	85.9 (sentido 0)	**WSJ_2**	71.5
natural	75.3 (sentido 0)	**WSJ_3**	58.7
social	83.3 (sentido 1)	**WSJ_4**	61.8
world 1	82.2 (sentido 1)	**WSJ_5**	61.7
world 2	89.3 (sentido 1)		

Tabla 9.14 Porcentaje del sentido más frecuente a priori (línea de base de referencia) en las cinco muestras del BNC y las cuatro muestras del WSJ con el homógrafo *drug*. Entre paréntesis se indica cuál es el sentido más frecuente en cada caso.

9.4.2 Ambigüedad compuesta: el efecto de la composición de muestras

Si las fuentes de dominio están ubicadas en algunas regiones del BNC y se encuentran distribuidas uniformemente a lo largo del WSJ podría ser interesante estudiar el efecto de juntar dos muestras diferentes de los corpus sobre la ambigüedad de la muestra resultante. A esta ambigüedad la hemos llamado ambigüedad compuesta.

Las Tablas 9.15, 9.16, 9.17 y 9.18 muestran los resultados de la ambigüedad compuesta al componer muestras dos a dos en todos los casos posibles, dentro de cada uno de los cuatro pares corpus/homógrafo.

Las Tablas 9.19 y 9.21 ofrecen un sumario de estos resultados para el homógrafo *space* y las Tablas 9.20 y 9.22 hacen lo mismo para el homógrafo *drug*.

En la Tabla 9.19 vemos que la ambigüedad compuesta media para dos muestras en el BNC es menor que la misma medida media intragrupos, pero mayor que la medida intergrupos, considerando los mismos grupos que en la sección anterior.

Esto quiere decir que la mezcla de dominios diferentes en el BNC produce un incremento de la ambigüedad, mientras que la mezcla de dominios similares hace todo lo contrario, disminuye la ambigüedad. Este último efecto se debe al incremento del tamaño de los conjuntos de entrenamiento en los experimentos supervisados.

Por el contrario, para el WSJ los resultados no hacen distinción de dominios, la ambigüedad intragrupos y la ambigüedad intergrupos es la misma e igual a la ambigüedad media total, y sólo está el efecto de la disminución de la ambigüedad como consecuencia del aumento del tamaño de los conjuntos de entrenamiento en un experimento supervisado.

	base	**total**	**sentido 0**	**sentido 1**
a+n_1	66.7 (sentido 0)	85.3	92.2	71.3
a+n_2	55.3 (sentido 0)	86.9	93.9	78.4
a+w	80.7 (sentido 0)	88.1	91.8	72.8
n_1+n_2	57.3 (sentido 1)	82.4	82.9	82.0
n_1+w	67.3 (sentido 0)	76.9	79.1	72.4
n_2+w	57.4 (sentido 0)	81.6	82.9	79.8
a+n_1+n_2+w	61.7 (sentido 0)	84.7	90.2	75.8

Tabla 9.15 Ambigüedad compuesta para todas las combinaciones posibles de yuxtaposiciones dos a dos de las cuatro muestras del BNC con el homógrafo *space*. La última fila muestra la composición de las cuatro muestras en una sola.

	base	**total**	**sentido 0**	**sentido 1**
1+2	65.2 (sentido 1)	85.3	91.4	82.0
1+3	64.2 (sentido 1)	87.5	91.4	85.3
1+4	56.7 (sentido 1)	87.1	94.2	81.7
2+3	72.3 (sentido 1)	86.6	85.4	87.1
2+4	65.3 (sentido 1)	88.1	87.6	88.3
3+4	64.2 (sentido 1)	86.5	86.8	86.3
1+2+3+4	64.8 (sentido 1)	87.5	94.5	83.7

Tabla 9.16 Ambigüedad compuesta para todas las combinaciones posibles de yuxtaposiciones dos a dos de las cuatro muestras del WSJ con el homógrafo *space*. La última fila muestra la composición de las cuatro muestras en una sola.

	base	total	sentido 0	sentido 1
a+n	81.3	92.9	96.4	77.7
a+s	51.7	82.4	86.3	78.2
a+w_1	53.7	73.3	79.8	65.7
a+w_2	53.5	69.3	88.9	52.1
n+s	57.4	78.4	91.2	69.0
n+w_1	55.6	75.6	87.3	66.3
n+w_2	62.6	57.3	96.1	34.2
s+w_1	82.8	82.2	73.9	83.9
s+w_2	86.4	68.9	84.7	66.4
w_1+w_2	86.1	88.3	64.3	92.2
a+n+s+w_1+w_2	60.3	77.1	90.1	68.7

Tabla 9.17 Ambigüedad compuesta para todas las combinaciones posibles de yuxtaposiciones dos a dos de las cinco muestras del BNC con el homógrafo *drug*. La última fila muestra la composición de las cinco muestras en una sola.

	base	total	sentido 0	sentido 1
2+3	64.2	87.8	86.1	90.8
2+4	50.5	91.2	89.9	92.6
2+5	66.1	85.3	82.5	90.8
3+4	52.8	92.0	90.8	93.0
3+5	60.1	89.9	88.4	92.1
4+5	52.1	92.0	89.5	94.2
2+3+4+5	55.2	91.0	90.3	91.8

Tabla 9.18 Ambigüedad compuesta para todas las combinaciones posibles de yuxtaposiciones dos a dos de las cuatro muestras del WSJ con el homógrafo *drug*. La última fila muestra la composición de las cuatro muestras en una sola.

La Tabla 9.20 muestra resultados análogos para el homógrafo *drug* pero nuevamente, como en el caso de las ambigüedades cruzadas, con mayor énfasis: la diferencia entre la ambigüedad intragrupo e intergrupo en el BNC sube hasta el 10.4%, mientras con *space* estaba en el 2.6% solamente. Este resultado indica que las variaciones de dominio en *BNC/drug* son mayores que en *BNC/space*. Esta hipótesis coincide con la precisión lograda por el algoritmo de Yarowsky en el BNC por esos dos homógrafos presentada en la Tabla 8.4: la precisión es un 3.6% más alta para *space* que para *drug*, y la diferencia alcanza el 15.4% si tenemos en cuenta la línea de referencia del porcentaje del sentido más frecuente.

	total	intragrupos	intergrupos
BNC	83.5	85.3	82.7
WSJ	86.9	86.9	86.9

Tabla 9.19 Ambigüedad compuesta media para la composición de muestras dos a dos del corpus BNC y del corpus WSJ con el homógrafo *space*. Se muestra la ambigüedad compuesta media total y la media dentro del mismo grupo y para muestras de grupos diferentes. Los grupos o clusters son los obtenidos en los experimentos de ambigüedad cruzada.

	total	intragrupos	intergrupos
BNC	76.9	83.1	72.7
WSJ	89.7	89.9	89.6

Tabla 9.20 Ambigüedad compuesta media para la composición de muestras dos a dos del corpus BNC y del corpus WSJ con el homógrafo *drug*. Se muestra la ambigüedad compuesta media total y la media dentro del mismo grupo y para muestras de grupos diferentes. Los grupos o clusters son los obtenidos en los experimentos de ambigüedad cruzada.

9.4.3 El efecto contradictorio del tamaño del corpus

Las Tablas 9.21 y 9.22 muestran el efecto que tiene sobre la ambigüedad el incremento del tamaño del corpus, para los homógrafos *space* y *drug* respectivamente.

El hecho de que los cambios de dominio sean mayores en *BNC/drug* que en *BNC/space* explica que el efecto de incremento de la precisión producido por el aumento del tamaño de los conjuntos de entrenamiento no sea suficientemente fuerte como para aumentar la precisión total cuando se yuxtaponen todas las muestras, Tabla 9.22, algo que por el contrario sí sucede en la Tabla 9.21.

Por lo tanto estamos ante un efecto contradictorio del incremento del tamaño del corpus si se trata de un corpus de texto general, como el BNC, o si se trata de un corpus periodístico de noticias, como el WSJ. En el primero, incrementar el tamaño implica la mezcla de dominios distintos y aunque también implica el aumento del tamaño de los ficheros de entrenamiento, esto último no es suficiente para impedir la caída de la precisión. En el segundo, el aumento del tamaño del corpus, si va acompañado de un aumento del tamaño del fichero de entrenamiento permite *siempre* un incremento de la precisión hasta llegar a un estado de saturación.

Sin embargo, como se ha mencionado en la sección 9.4.1, las diferencias entre los corpus BNC y WSJ podrían a primera vista deberse a la manera en que se han tratado o, en otras palabras, podría de hecho haber cambios de dominio en el WSJ, pero no se los distingue agrupándolos en documentos del mismo dominio/tipo. Es decir, las fuentes de dominio en el BNC estarían formando grupos o protuberancias, mientras que en el WSJ seguirían una distribución homogénea. Pero si el WSJ también tiene cambios de dominio, ¿cuál sería la diferencia entre una muestra grande del WSJ y una muestra grande del BNC, *ambas* con fluctuaciones de dominio aunque distribuidas de forma diferente o, *por qué* las precisiones resultantes son tan diferentes en ese caso?

En una muestra de tamaño pequeño la precisión tiende a ser similar en ambos corpus porque la supuesta precisión alta en el BNC debida a la pureza de su fuente de dominio se ve compensada por el pequeño tamaño del conjunto de entrenamiento dando lugar a una precisión relativamente baja. Para el WSJ las fuentes de dominio podrían estar

mezcladas a pesar del tamaño pequeño de la muestra porque los documentos son muy pequeños y la muestra podría contener documentos de dominios diferentes. Como el conjunto de entrenamiento también sería pequeño, también se obtendría una precisión baja.

A pesar de todo, este no puede ser el panorama completo. Tiene que haber alguna otra razón para explicar el aumento de precisión que acompaña un incremento del fichero de entrenamiento en el WSJ y tiene que ser distinta de la distribución estadística de las fuentes de dominio. Entre las hipótesis más plausibles está que un corpus periodístico como el WSJ constituye un tipo muy especial de corpus ya que utiliza un lenguaje estereotipado, repetitivo y homogéneo que carece de la variedad, no forzosamente en vocabulario sino en uso, del lenguaje utilizado en los corpus de texto general como el BNC, y que se refiere una y otra vez a las mismas situaciones del mundo real.

	AVG (1)	**AVG (2)**	**AVG (4)**
BNC	84.0	83.5	84.7
WSJ	84.1	86.9	87.5

Tabla 9.21 Ambigüedad media para muestras individuales, composición de dos muestras y composición de las cuatro muestras para el homógrafo *space*.

	AVG (1)	**AVG (2)**	**AVG (4)**	**AVG (5)**
BNC	85.7	76.9		77.1
WSJ	86.2	89.7	91.0	

Tabla 9.22 Ambigüedad media para muestras individuales, composición de dos muestras y composición de las cuatro muestras (WSJ) o de las cinco muestras (BNC) para el homógrafo *drug*.

9.4.4 Los corpus utilizados en las competiciones Senseval

Como se ha señalado en el capítulo 2 el certamen Senseval, que empezó en 1998 y ya ha celebrado cuatro ediciones en los años 1998, 2001, 2004 y 2007 [Kilgarriff 1998] [Kilgarriff y Palmer 2000] [Edmonds y Cotton 2001] [Mihalcea y Edmonds 2004], se ha convertido en el foro más importante dedicado a la evaluación y a la comparación de sistemas de DSP de todo tipo. Los certámenes Senseval han constituido un éxito sin paliativos, y la precisión lograda en ellos por los diferentes tipos de sistemas no ha dejado de incrementarse en cada edición, si bien, y precisamente debido a este éxito, se considera que los resultados de precisión han 'tocado techo' y ya en la última edición de 2007, que ha cambiado su nombre a Semeval, se ha intentado empezar a explorar nuevos horizontes.

Los sistemas de DSP basados en corpus supervisados son los que han logrado los niveles de precisión más altos en estos certámenes [Palmer et al. 2006]. Sin embargo, este hecho también constituye uno de los puntos débiles de Senseval, por lo menos

desde un punto de vista, ya que la importancia de los sistemas basados en corpus supervisados fuerza el uso de subconjuntos de corpus de entrenamiento etiquetados manualmente relativamente grandes. Esto hace que la competición en su conjunto se encuentre sometida al Cuello de Botella de la Adquisición de Conocimiento (CBAC) ya que estos corpus etiquetados manualmente también se utilizan en la evaluación de sistemas basados en corpus no supervisados como medio de compararlos directamente con los supervisados.

Como consecuencia, se ha prestado muy poca atención a la naturaleza de los corpus de entrenamiento y prueba (test) o a cualquier tipo de comparación relativamente sistemática de las diferencias intrínsecas de los corpus.

Una simple observación de los corpus objetivos utilizados en los tres primeros ejercicios lo confirma. En Senseval-1, "lexicógrafos profesionales etiquetaron oraciones (…) que se habían extraído del corpus Hector (un pionero del BNC)" [Palmer et al. 2006]. En Senseval-2, la tarea 'todas las palabras' en idioma inglés "incluyó 5 000 palabras de texto compacto extraído del corpus Penn Treebank II *Wall Street Journal*, pero se suplementó con datos del British National Corpus siempre que hubo un número insuficiente de ejemplos del Treebank" [Palmer et al. 2006]. El corpus Penn Treebank II está compuesto básicamente por artículos del Dow Jones Newswire, del Brown Corpus, de abstracts del US Dept. of Energy y de textos del MUC-3 (artículos del Federal News Service). El Brown Corpus es un corpus dividido informalmente en categorías que mezclan diversas variaciones de dominio y genio. Una parte importante del mismo contiene secciones extraídas de la prensa.

Senseval-3 intentó evitar los anotadores especializados y costosos económicamente haciendo uso del proyecto OMWE [Chklovski y Mihalcea 2002]. El proyecto Open Mind Word Expert invita a los usuarios normales de la Web a desambiguar palabras en contexto creando así un corpus etiquetado de sentidos relativamente grande. Sin embargo, como dice [Palmer et al. 2006] "el control de calidad está pendiente ya que los usuarios normales de la Web son noveles para esa tarea en comparación con los anotadores entrenados en lingüística que se usaron en los esfuerzos de etiquetado previos (Semcor, DSO y Senseval)".

Como se ha observado en la sección anterior, la razón que hay detrás de los resultados sorprendentemente altos de precisión de los algoritmos de DSP aplicados sobre corpus periodísticos es su condición de corpus especiales en el sentido de utilizar un lenguaje estereotipado y repetitivo que se refiere una y otra vez a las mismas situaciones del mundo real. Dado el uso extensivo y más bien desordenado de corpus periodísticos en las competiciones Senseval que se ha puesto de manifiesto en los párrafos precedentes, y no solamente en tareas de DSP, sino en muchas otras como por ejemplo Recuperación de la Información, creemos necesario advertir contra su uso excesivo en futuros experimentos y tener en cuenta que se trata de un tipo de corpus de características muy especiales. En general se podría decir que la gran variedad de dominio y ambigüedad de

los corpus reales (o de texto general) o un estudio relativamente sistemático de ella es un asunto que no ha sido abordado suficientemente en Senseval y, quizás como consecuencia, en la literatura sobre DSP en general.

Capítulo 10.
El autoarranque del algoritmo de Yarowsky en corpus reales

El algoritmo de Yarowsky resuelve el problema de la Desambiguación del Sentido de las Palabras (DSP) aplicado a las palabras con ambigüedad homográfica, que es el nivel de distinción de sentidos necesario en las aplicaciones reales de Procesamiento de Lenguaje Natural (PLN). Al mismo tiempo, al ser un algoritmo casi no supervisado, es decir, que prácticamente no precisa de entrenamiento, también resuelve el problema del Cuello de Botella de la Adquisición de Conocimiento (CBAC). Esto significa que, por ejemplo, se puede aplicar fácilmente sobre corpus escritos en un idioma extranjero no conocido, del que no se disponga de corpus de entrenamiento etiquetados manualmente, y estos corpus sí serían necesarios para cualquier algoritmo supervisado.

Sin embargo, como se ha señalado en los capítulos 8 y 9, estos resultados plenamente satisfactorios del algoritmo de Yarowsky sólo se producen para ciertos corpus de una naturaleza muy especial como son los corpus de noticias periodísticas. En otras circunstancias la precisión del algoritmo decae considerablemente hasta niveles inaceptables para aplicaciones reales, como cuando se aplica sobre corpus de texto general que presentan fluctuaciones de dominio [Sánchez de Madariaga, Paice, Rayson y Fernández del Castillo, 2008].

En este capítulo se hace una revisión de la vigencia actual de los algoritmos semisupervisados, tanto desde el punto de vista del alivio del CBAC en un entorno multilingüe, como pueda ser la Web, como desde el punto de vista de su utilidad para tratar con las fluctuaciones de dominio en los corpus generales. A continuación se presenta una nueva metodología de autoarranque del algoritmo de Yarowsky sobre corpus de texto general [Sánchez de Madariaga y Fernández del Castillo, 2008] que permite alcanzar unos niveles de precisión bastante más elevados que los del algoritmo standard de 1995 [Yarowsky 1995], si bien no se logran los mismos resultados que en

los corpus periodísticos, debido a las ambigüedades intrínsecas de los corpus de texto general.

10.1 La vigencia de los algoritmos de aprendizaje semisupervisados

Los investigadores y desarrolladores en el campo del PLN se enfrentan cada vez más con la necesidad de desarrollar componentes de tecnología lingüística en lenguajes nuevos [Hwa et al., 2005]. En la actualidad existen en todo el mundo unos 200 lenguajes diferentes considerados con suficiente relevancia a este respecto. El éxito reciente de los métodos basados en corpus, incluidos en la disciplina conocida como Lingüística de Corpus (LC), en inglés Corpus Linguistics (CL), combinado con la anterior observación, no ha hecho sino agravar el problema del CBAC. El intento de tratar este problema ha llevado al desarrollo de algoritmos supervisados débilmente, tales como el aprendizaje activo [Hermjakob y Mooney, 1997] [Tang, Luo y Roukos, 2002] [Baldridge y Osborne, 2003] [Hwa, 2004], el autoentrenamiento y el coentrenamiento [Sarkar, 2001] [Steedman, Osborne, Sarkar, Clark, Hwa, Hockenheimer, Ruhlen, Baker y Crim, 2003] y el autoarranque mediante la proyección a través de textos paralelos [Yarowsky y Ngai, 2001] [Merlo, Stevenson, Tsang y Allaria, 2002] [Yarowsky, Ngai y Wicentowski, 2001].

El aprendizaje activo (AA), en inglés active learning (AL), es una técnica que intenta reducir el coste de anotar corpus etiquetados mediante la selección de los siguientes ejemplos que es mejor anotar: esos ejemplos son aquellos sobre los que el sistema de aprendizaje tiene mayor incertidumbre (en inglés este criterio se llama *uncertainty sampling* [Cohn et al., 1995] [Osborne y Baldridge, 2004]). Sin embargo, en contra de lo esperado, la generación de corpus etiquetados para un modelo dado siguiendo esta técnica y su ulterior reutilización con otro modelo, puede producir resultados despreciables e incluso negativos [Baldridge y Osborne, 2004]. Otras técnicas relacionadas con esta como la anotación activa, en inglés *active annotation*, [Vlachos, 2006] se han introducido para intentar tratar el problema de la reusabilidad de los datos obtenidos a través del aprendizaje activo. A pesar de que la reducción en el coste de la anotación es comparable a la obtenida con la primera técnica y de que los corpus producidos son más transportables, esta segunda técnica precisa de la intervención manual de una persona anotadora.

El autoarranque mediante la proyección a través de textos paralelos trata de etiquetar un corpus nuevo en un lenguaje dado mediante la proyección de conocimiento lingüístico presente en un corpus de texto paralelo etiquetado escrito en un lenguaje del que se disponga de abundantes recursos lingüísticos, como por ejemplo el inglés. A esta técnica se la denomina proyección de la anotación usando texto paralelo y ha sido llevada a cabo para muchas tareas usuales de PLN [Hwa et al., 2005]. Evidentemente, la mera necesidad de disponer de textos paralelos en ambos lenguajes es una desventaja importante de este método.

El autoentrenamiento (*self-training* en inglés) y el coentrenamiento (en inglés *co-training*) son los métodos semisupervisados introducidos por los algoritmos de autoarranque de Yarowsky [Yarowsky, 1995] y de Blum y Mitchell [Blum y Mitchell, 1998] respectivamente que se han descrito en los capítulos 5, 3 y 2. Los algoritmos de autoarranque semisupervisados han recibido mucha atención en los últimos años y se han aplicado a muchas técnicas de PLN diferentes, y también muchas veces estas técnicas de PLN se han utilizado como casos prácticos para el estudio de los algoritmos de autoarranque.

Pierce y Cardie [Pierce y Cardie, 2001] han utilizado como caso práctico de tarea de PLN un analizador de frases nominales (en inglés *noun phrase bracketing*) para el estudio del comportamiento del aprendizaje mediante el coentrenamiento. Para ello compararon el algoritmo de autoarranque con un algoritmo completamente supervisado que alcanzaba una precisión de 95.17%. El algoritmo de autoarranque logró una precisión de 93.3%, pero este en principio buen resultado depende del parámetro llamado L (la cantidad inicial de ejemplos etiquetados suministrados al algoritmo de entrenamiento) y el nivel óptimo de este parámetro no es un dato global para todas las palabras objetivo, ni conocido a priori para ninguna de ellas. En general, llegan a la conclusión de que el algoritmo de coentrenamiento (y por extensión el de autoentrenamiento) es bastante sensible al ajuste de sus parámetros (el referido L y el número de iteraciones, entre otros). Otra observación importante fue que la precisión alcanzada no se mantenía con el progreso del coentrenamiento, esto es, a medida que el número de iteraciones crecía, y atribuyeron este declive a la degradación en la calidad de los ejemplos que se iban etiquetando según progresaba el algoritmo. Pierce y Cardie han propuesto combinar los métodos de aprendizaje semiautomático con los métodos de aprendizaje activo y han introducido el coentrenaiento corregido como un método "moderadamente supervisado" que simula automáticamente a un hipotético anotador personal que corrigiera cada ejemplo etiquetado nuevamente que se añade a los datos ya etiquetados. Este coentrenamiento corregido alcanzó un 94.5% de precisión y por tanto logró la precisión del algoritmo plenamente supervisado después de ajustar globalmente sólo uno de sus parámetros. Es decir, este método solucionó el problema del ajuste de parámetros y también el de la proliferación de nuevos ejemplos mal etiquetados, pero este último a un coste demasiado alto, que contradice el espíritu del aprendizaje semisupervisado. La intervención manual de una persona hace que de hecho sea un algoritmo casi supervisado.

El trabajo de Pierce y Cardie [Pierce y Cardie, 2001] ilustra con claridad los dos principales problemas exclusivos de los algoritmos semisupervisados: su susceptibilidad al ajuste de varios parámetros y la proliferación de ejemplos mal etiquetados. Mihalcea [Mihalcea, 2004] ha utilizado una tarea de DSP para estudiar el autoentrenamiento y el coentrenamiento y ha obtenido una precisión de 65.61% y de 65.75% respectivamente usando una tarea de tipo 'muestra léxica' estilo Senseval con varios sentidos por cada palabra ambigua. Por supuesto se debe notar que esos resultados de precisión

relativamente bajos se deben a que se utilizó un nivel de granularidad relativamente fina comparados con los sólo dos sentidos por palabra que se utilizan a nivel de ambigüedad homográfica. Es decir, esos niveles de precisión son los habituales para ese tipo concreto de experimentos. Además, el algoritmo supervisado incrustado dentro del algoritmo semisupervisado sólo llegó a una precisión de 53.84%. Sin embargo, estos resultados vuelven a padecer el mismo problema que los de Pierce y Cardie, ya que corresponden a un ajuste óptimo de los tres parámetros del algoritmo, lo cual significa un ajuste diferente para la precisión óptima de cada experimento, es decir, para cada palabra objetivo de la tarea de 'muestra léxica', algo muy poco realista en una aplicación práctica. La precisión alcanzada por estos algoritmos, con un ajuste global de parámetros, baja hasta 55.67%. La razón de la pérdida de precisión está en la aparición de ejemplos mal etiquetados como consecuencia del ajuste no óptimo de los parámetros en todos los experimentos. Mihalcea [Mihalcea, 2004] ha propuesto un nuevo método de autoarranque basado en votaciones por mayoría: el clasificador en cada iteración se sustituye por una votación por mayoría entre todos los clasificadores de todas las iteraciones anteriores. El efecto producido ha sido beneficioso en ambos aspectos y logró un intervalo más largo de iteraciones durante las cuales la precisión también fue más alta, de un 58.35%. Ng y Cardie [Ng y Cardie, 2004] han usado votación por mayoría simple [Breitman, 1996] en una tarea de resolución de correferencia de frases nominales con autoentrenamiento y coentrenamiento y han concluido que el primero superó al segundo en precisión y también fue comparativamente menos sensible al ajuste de parámetros.

Steedman et al. [Steedman et al., 2003] han aplicado autoentrenamiento y coentrenamiento al problema del autoarranque de analizadores sintácticos (parsers) estadísticos. Un parser estadístico normal puede trabajar con una medida F[21] de un 89% si utiliza un número suficientemente grande de oraciones de entrenamiento. Steedman et al. han logrado una medida F de 79.0% en sus experimentos con el autoarranque de parsers estadísticos, y han atribuido el resultado más bajo a que la tarea del análisis sintáctico es bastante más complicada que muchas otras. A pesar de todo, también han proyectado sus resultados a un número mucho mayor de ejemplos de entrenamiento (400 000) y el resultado ha sido de 90.4%, lo cual indica que un parser estadístico autoarrancado puede superar en eficacia a un parser estadístico estándar que trabaje al nivel del estado del arte actual.

10.1.1 Aprendizaje semisupervisado y variación de dominio

Steedman et al. también se refieren a un tercer problema de los algoritmos de autoarranque, que es originado por la posible variación de dominio en y entre los corpus (véase también la sección 2.6). Este problema estaba por ejemplo detrás uno de los

[21] La llamada medida F es una media armónica ponderada de la precisión y el recall, que se obtiene aplicando la siguiente fórmula: $F = 2(precision * recall)/(precision + recall)$.

resultados obtenidos por ellos en el experimento de la sección anterior según el cual una medida F de un 79.0% alcanzada en un determinado corpus caía a un 76.8% si el entrenamiento se había efectuado en un corpus distinto.

He y Gildea [He y Gildea, 2004] también se han referido a este problema al realizar experimentos de autoentrenamiento y coentrenamiento con una tarea de etiquetado de roles semánticos (*semantic role labelling* en inglés). En su trabajo también se refieren al hecho de que es poco práctico intentar tener datos anotados manualmente para todos y cada uno de los dominios que puedan surgir. Sin embargo, el problema de la fluctuación de dominios en los corpus no es de ningún modo un problema exclusivo de los algoritmos de aprendizaje semisupervisado, sino que también afecta a los algoritmos supervisados [Sánchez de Madariaga, Paice, Rayson y Fernández del Castillo, 2008]. De hecho, He y Gildea han asegurado que el motivo inicial de su investigación en ese trabajo ha sido utilizar algoritmos de aprendizaje supervisados débilmente para ajustar los clasificadores entrenados con datos etiquetados en dominios dados a su aplicación sobre datos de otros dominios nuevos, lanzando la hipótesis del potencial de este tipo de algoritmos para tratar el problema de la variación de dominios.

10.1.2 Otras aproximaciones posibles al CBAC

Un enfoque ligeramente diferente al CBAC podría ser intentar utilizar los recursos disponibles para una tarea determinada para resolver el problema planteado por otra tarea relacionada. Por ejemplo, podríamos intentar resolver el problema de la DSP usando recursos disponibles para la tarea de etiquetado de roles semánticos.

En este trabajo hemos comprobado su validez utilizando material de la última edición de Semeval/Senseval, celebrada en 2007. En concreto, se ha utilizado material de la tarea 17 de ese certamen, llamada *English Lexical Sample WSD, English All Words WSD and Semantic Role Labelling* [Pradhan et al., 2007]. Entre este material hay disponible corpus de entrenamiento etiquetados manualmente con sentidos de palabras ambiguas para la tarea de DSP (WSD) y con roles semánticos para la tarea de SRL (semantic role labelling). Ambos conjuntos de etiquetas se refieren a las mismas ocurrencias de las palabras objetivo en los mismos corpus. A partir de estos datos, hemos calculado los resultados que se hubieran obtenido mediante la utilización de un anotador de roles semánticos para desambiguar los sentidos de los homógrafos *space* y *drug*, utilizados a lo largo de todo este trabajo. Hemos hecho corresponder los sentidos distinguidos en Semeval/Senseval para esas dos palabras objetivo exactamente con los sentidos utilizados en este trabajo, y las etiquetas utilizadas en esos datos en la parte de roles semánticos son los roles habituales como *agente, tema, experimentador, causa*, etc.

Los experimentos han dado como resultado que para el homógrafo *space* un anotador de roles semánticos perfecto, es decir, construido manualmente, obtendría una precisión de

un 80% (para un recall de 100%) si se utilizara para determinar los dos sentidos de esa palabra objetivo en una tarea de DSP. La precisión de referencia de base a priori en el corpus del experimento, dada por la proporción de ocurrencias del sentido más frecuente, era de un 64.0%. En el caso del homógrafo *drug* la precisión ha sido de 85.3% para un 100% de recall y una precisión de referencia del sentido más frecuente de 82.3%. El número de etiquetas de roles semánticos que se han utilizado en el experimento ha sido de once. [Sánchez de Madariaga y Fernández del Castillo, 2008].

Es importante destacar que el corpus sobre el que se hizo el experimento fue el WSJ, donde el algoritmo de DSP de Yarowsky y muchos de los algoritmos de DSP supervisados logran una precisión en torno al 95% para un 100% de recall. Evidentemente también hay que tener en cuenta que en este experimento se ha utilizado un anotador de roles semánticos 'perfecto', porque ha sido creado manualmente por expertos para una de las tareas de Semeval 2007, y que en una situación práctica se utilizaría un anotador real que por supuesto trabajaría con unos niveles de precisión más bajos.

Estos resultados nos indican que la utilización de corpus etiquetados en el campo de los roles semánticos aplicados a la tarea de DSP no constituiría una solución plena para este problema.

Tampoco parece que otros tipos de anotadores semánticos como los llamados reconocedores de entidades nombradas, en inglés *named-entity recognizers*, que distinguen entre ubicaciones, personas, organizaciones, meses, etc. vayan a mejorar los resultados obtenidos por el anotador de roles semánticos anterior, al menos para las palabras *space* y *drug* en el corpus WSJ.

Otra posibilidad de abordar el problema del CBAC sería utilizar el gran volumen de conocimiento lingüístico disponible en las bases de datos léxicas ya existentes para; primero, etiquetar el corpus objetivo, y después, intentar desambiguar entre los posibles sentidos de una palabra dada. Además de que esta posibilidad todavía no ha sido puesta en práctica [Rayson et al., 2004] se debe notar que en realidad estaría resolviendo un problema muy diferente del que resuelven los algoritmos de DSP semisupervisados desde la perspectiva del CBAC, ya que por ejemplo tendría que trasladarse todo ese conocimiento léxico a un sistema en un lenguaje diferente, algo que no es bajo ningún concepto una tarea inmediata, incluso cuando se dispone de abundantes recursos lingüísticos automatizados en ambos idiomas [Löfberg et al., 2003].

10.2 Una metodología de autoarranque del algoritmo de Yarowsky en corpus de texto general

Los corpus de texto general están compuestos por muchos documentos que tratan sobre temas muy diferentes, es decir, su rango de dominios puede ser muy extenso. Esta

variación en sus dominios y el hecho de que no utilicen un lenguaje especial, a diferencia de lo que hacen los corpus periodísticos, es la causa de que los algoritmos de DSP logren en aquellos corpus resultados de precisión bastante inferiores a los que obtienen en éstos.

En el caso del algoritmo de Yarowsky, su autoarranque sobre un corpus de texto general podría abordarse intentando el autoarranque de cada documento del corpus aisladamente, de la forma habitual. Se supone que cada documento del corpus trata sobre un dominio determinado y carece de fluctuaciones de éstos, al menos de carácter fuerte, con lo que el algoritmo standard obtendría unos resultados altos de precisión.

Sin embargo, este enfoque cuenta con inconvenientes serios. En primer lugar, para autoarrancar el algoritmo en cada documento habría que seleccionar las semillas iniciales para todos los documentos. Si se hace manualmente, se está usando, de hecho, un algoritmo supervisado, con lo que se perderían las ventajas del algoritmo de Yarowsky. Pero tampoco se puede hacer automáticamente, por varias razones. Primero, porque los cambios de dominio impiden un método de selección automático eficaz sobre todos ellos. Además, hay que tener en cuenta que los métodos automáticos como los descritos en el capítulo 5 utilizan algún tipo de fuente de conocimiento lingüístico externo, lo que transgrede el carácter semisupervisado de todo el algoritmo. Esto ocurre no solamente por los cambios de dominio en sí, sino también por el hecho de que también hay un grado de ambigüedad intrínseco de los documentos que es muy diferente de unos y otros. Finalmente, el algoritmo es muy susceptible a la proporción de semillas iniciales correctas, de forma que una muy ligera bajada en esta proporción puede dar lugar a una caída muy fuerte de la eficacia final de todo el algoritmo.

Todo lo anterior indica que sencillamente no se puede aplicar el mismo conjunto de semillas iniciales a todo el corpus, incluso si se intenta autoarrancar cada documento del corpus por separado.

Se necesita un procedimiento para generar conjuntos de colocaciones para el autoarranque que cambien dinámicamente a lo largo de los documentos y dominios del corpus. En esto el algoritmo de Yarowsky tiene la ventaja de utilizar como algoritmo de aprendizaje automático la Lista de Decisión, que es suficientemente flexible para conseguir esta adaptación.

Además se necesita algún procedimiento para validar esas listas de decisión que van cambiando dinámicamente, para evitar la aplicación de listas de decisión falsas o degeneradas a otros documentos del corpus. Este procedimiento utiliza la propiedad original del algoritmo de *un sentido por discurso* como se explica en la siguiente sección.

10.2.1 La aplicación de la propiedad *un sentido por discurso*

La altísima precisión de la propiedad *un sentido por discurso* en ambigüedades de dos sentidos [Yarowsky 1995] permite utilizar un enfoque 'digital' en el algoritmo de autoarranque. En concreto, el algoritmo se aplica por separado sobre cada documento y el sistema decide si esta aplicación ha tenido éxito o no. Esta decisión sirve para validar la lista de decisión generada en esa aplicación y así poder aplicarla sobre otros documentos.

Para tomar esta decisión el sistema utiliza un umbral empírico predefinido y aplica la siguiente fórmula una vez terminada la aplicación del algoritmo standard sobre un documento determinado:

$$th = \frac{\sum s_0 - \sum s_1}{n_0 - n_1}$$

En la fórmula n_0 y n_1 representan el número total de contextos etiquetados correspondientes a cada sentido de la palabra objetivo (sentidos 0 y 1) después de la aplicación del algoritmo al documento; s_0 y s_1 representan los valores de la *similitud logarítmica* (log-likelihood) obtenidos por esos contextos etiquetados.

Si el valor *th* correspondiente al documento sobre el que acaba de ser aplicado el algoritmo supera el umbral predefinido, se considera que la aplicación ha tenido éxito, todos los contextos del documento se etiquetan con la etiqueta del sentido correspondiente y la lista de decisión resultante se guarda en una estructura de datos para su futura aplicación a otros documentos que aún no hayan sido etiquetados. Si el valor *th* no supera el umbral, se considera que la aplicación ha fracasado, no se etiqueta ningún contexto del documento y la lista de decisión resultante se considera defectuosa y se desecha.

Como consecuencia, si una aplicación tiene éxito con un documento, se está determinando el sentido del documento según la palabra u homógrafo objetivo, algo que podríamos llamar el *sentido homográfico* o *tema homográfico* del documento. Esta determinación puede resultar muy interesante, al menos desde el punto de vista de la Recuperación de la Información (RI).

10.2.2 El algoritmo de autoarranque

El algoritmo de autoarranque necesita sólo como semillas iniciales dos contextos etiquetados manual y correctamente. Cada uno de estos dos contextos debe pertenecer a un documento diferente y los *temas homográficos* de los dos documentos deben ser distintos (cada uno debe corresponder a uno de los dos sentidos del homógrafo) y bien

definidos. Los dos documentos se yuxtaponen en un único documento inicial, que tiene los únicos dos contextos etiquetados manualmente al principio. El tamaño de estos dos documentos iniciales no necesita ser muy grande, basta con que tengan unos 10 contextos cada uno. Téngase en cuenta que estamos hablando en realidad de pares documento/homógrafo, es decir, de conjuntos de contextos o, si se quiere, de ocurrencias del homógrafo en el documento del corpus.

Este documento se autoarranca con el algoritmo de Yarowsky a partir de los dos únicos contextos etiquetados que funcionan como semillas y esta primera ejecución del algoritmo debe tener éxito en el sentido de que la gran mayoría de contextos del documento inicial de sentido 0 deben quedar etiquetados con sentido 0 y lo mismo debe ocurrir con el documento inicial de sentido 1. Esta condición se puede comprobar muy fácilmente de forma automática, y su éxito es una precondición para el éxito del autoarranque de todo el corpus.

La lista de decisión resultante del autoarranque del documento inicial, que es un clasificador de contextos, se aplica a cada documento por separado sin seguir ningún orden de documentos especial. Esto quiere decir que se aplica el algoritmo de Yarowsky a cada documento sin autoarrancarlo a partir de semillas etiquetadas inicialmente, sino utilizando una lista de decisión preconstruida.

Si esta aplicación tiene éxito en la decisión del tema homográfico del documento según el procedimiento descrito en la sección anterior, la lista de decisión queda actualizada con nuevas colocaciones como resultado de la ejecución normal del algoritmo de Yarowsky sobre el documento, y además la lista de decisión actualizada se considera útil y formada por nuevas colocaciones correctas.

Sin embargo sólo se guardan las colocaciones con los valores de similitud más altos, es decir, unos pocos de los primeros elementos de la lista de decisión nueva. Esto asegura que la lista de decisión vaya cambiando rápidamente. Una de las razones del fracaso del algoritmo de Yarowsky original en corpus con dominios variables es que las colocaciones se mezclan en una única lista de decisión de ámbito todo-el-corpus. Esta lista de decisión única contiene colocaciones no sólo de un dominio sino de muchos dominios mezclados, y al final es incapaz de clasificar los contextos correctamente. El hacer a la lista cambiar con cada nuevo documento de dominio diferente tiene el efecto de no mezclar dominios mientras se autoarrancan documentos con dominios similares. Como consecuencia de esto el algoritmoº se tiene que aplicar a todos los documentos del corpus que todavía no se hayan decidido varias veces hasta que se alcance un estado estable en el que no se decida el tema de ningún nuevo documento. Este estado se suele alcanzar después de varias iteraciones por todo el corpus.

Otra consecuencia de esto es que, una vez que una lista de decisión ha sido generada, se debe aplicar a todos los documentos todavía no decididos *antes de que sea a su vez actualizada por ninguno de éstos previamente*. Esto implica que se debe utilizar algún

tipo de estructura de datos para almacenar la lista de decisión inmediatamente después de su creación y evitar que sea actualizada antes de que se intente su uso en todos los documentos sin decidir todavía. Esta política de mantenimiento sugiere como estructura de datos más apropiada una cola FIFO.

Además, como cuando el algoritmo tiene éxito con un documento determinado y decide su tema homográfico es bastante probable que la lista de decisión que se genera esté desviada hacia uno de los dos sentidos, el sentido mayoritario en ese documento, de hecho se utilizan dos colas FIFO diferentes, una para cada sentido, como se puede ver en la Figura 10.1. Las dos listas que están en ambos frentes de las dos colas se combinan en una sola lista de decisión como se ve en la Figura 10.2. Esta lista es la que se aplica a todos los documentos que todavía están por decidir. El proceso se repite siguiendo un esquema de *todas con todas* hasta que una de las dos colas se quede vacía. Naturalmente esto ocurre porque las listas se suprimen del frente de la cola una vez que se han probado en todo el corpus en combinación con todas las demás listas de la otra cola. En la Figura 10.3 se puede ver una definición formal de todo el algoritmo. Como se puede observar en esa figura el algoritmo de más bajo nivel utiliza en realidad una tercera cola auxiliar.

La aplicación de este algoritmo al subconjunto del British National Corpus (BNC) compuesto por los documentos clasificados como de todos los dominios excepto del dominio '*imaginative*' y clasificados a la vez como del medio '*books*', supone en total aproximadamente el 50% de todos los documentos del corpus. Utilizando como palabras objetivo los homógrafos *drug*, *plant* y *space* da como resultado habitualmente un número no muy alto de documentos etiquetados, es decir, con tema decidido, después de varias vueltas alrededor de todo el corpus y con una precisión muy alta (entre el 95% y el 100%) y un bajo *recall* (porcentaje de contextos etiquetados). La aplicación ulterior del mismo algoritmo a los documentos restantes sigue dando como resultado una precisión muy parecida pero un *recall* ligeramente más bajo cada vez. Además, después de tres o cuatro aplicaciones de todo el algoritmo se llega a un estado en el que ya no se etiqueta ningún nuevo documento. Esto quiere decir que hay una serie de documentos lo suficientemente ambiguos como para que ninguna lista de decisión entrenada en otros documentos sea capaz de clasificarlos según su tema.

En la Tabla 10.1 se pueden ver los resultados de precisión al 100% de *recall* que se obtienen después de aplicar el algoritmo de autoarranque cuatro veces con cada uno de

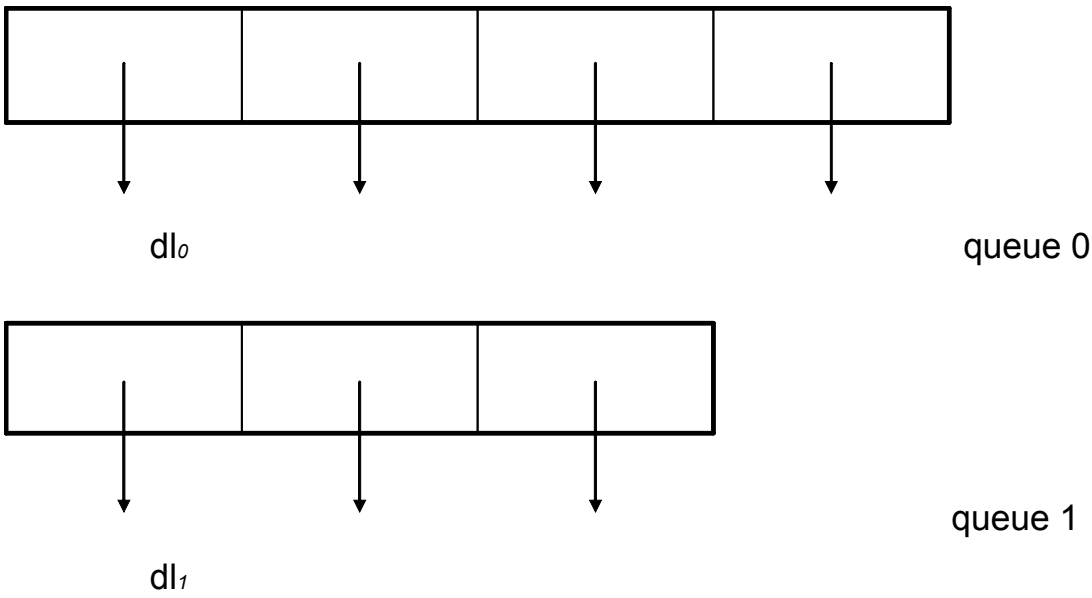


Figura 10.1 .Representa las dos colas FIFO que se utilizan para cada sentido del homógrafo bajo desambiguación (sentidos 0 y 1). Las listas de decisión que están en el frente de cada cola se denominan dl_0 y dl_1 y se combinan en una única lista de decisión como se muestra en la Figura 10.2. Las listas de decisión se combinan siguiendo un régimen de *todas-con-todas* y el proceso termina cuando una de las dos colas se queda vacía. Nótese que durante este proceso se pueden añadir nuevas listas de decisión al final de ambas colas.

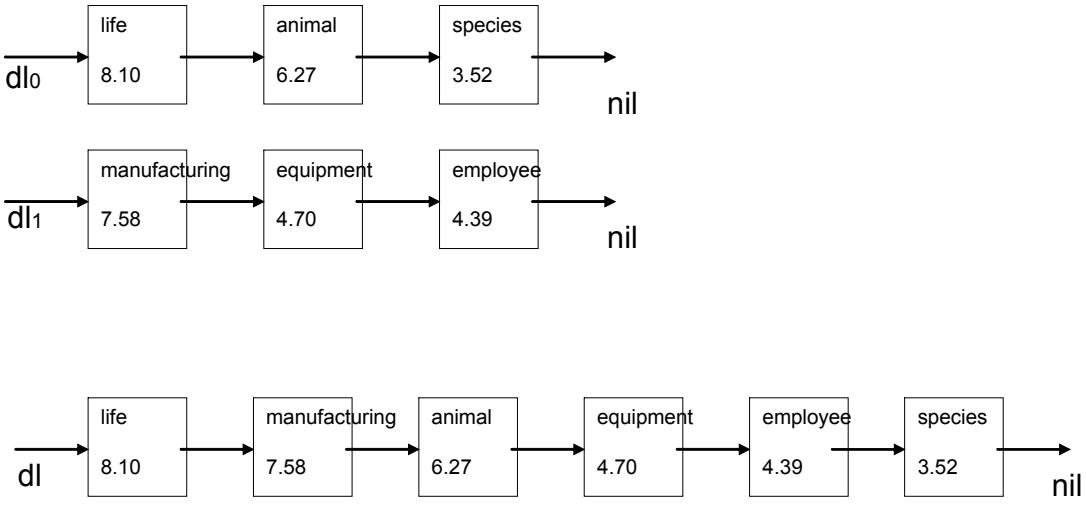


Figura 10.2. Representa la lista de decisión *dl* generada al combinar las listas de decisión dl_0 y dl_1 que han sido extraídas del frente de las colas 0 y 1 representadas en la Figura 10.1. La lista de decisión nueva queda ordenada por el rango de los valores de similitud logarítmica alcanzados por las colocaciones al aplicar el algoritmo de Yarowsky con una lista de decisión previa sobre un documento nuevo. Las listas de decisión nuevas sólo se mantienen si la aplicación del algoritmo al nuevo documento tiene éxito en la decisión sobre el tema de éste. Las colocaciones que se muestran en la figura pertenecen al homógrafo *plant* y son ficticias (extraídas de Yarowsky 1995).

1. Construir un documento especial inicial con dos contextos etiquetados manualmente
2. Autoarrancar el algoritmo de Yarowsky en ese documento
3. Aplicar la lista de decisión resultante a todos los documentos del corpus
    3.1. Si se decide el tema de un documento
        3.1.1. Etiquetar el documento con el sentido de su tema
        3.1.2. Mantener sólo los tres primeros elementos de la lista de decisión
        3.1.3. Insertar la lista de decisión resultante en su cola correspondiente
4. Mientras la cola 0 no esté vacía
    4.1. Llamar a la lista en el frente de la cola 0 *dl0*
    4.2. Mientras la cola 1 no esté vacía
        4.2.1. Llamar a la lista en el frente de la cola 1 *dl1*
        4.2.2. Mezclar *dl1* con *dl0* en *dl*
4.2.3. Aplicar la lista *dl* a todos los documentos del corpus no etiquetados
            4.2.3.1. Si se decide el tema de un documento
                4.2.3.1.1.Etiquetar el documento con el sentido de su tema
4.2.3.1.2. Mantener sólo los tres primeros elementos de la nueva lista de decisión resultante
                4.2.3.1.3. Insertar la lista de decisión resultante en su cola correspondiente
        4.2.4. encolar *dl1* en cola_auxiliar
        4.2.5. desencolar cola 1
    4.3. copiar cola_auxiliar en cola 1
    4.4. desencolar cola 0

Figura 10.3. El nuevo algoritmo de autoarranque.

Palabra	Base	autoarranque 1	autoarranque 2	autoarranque 3	autoarranque 4
Drug	62.5	70.8	77.1	81.2	85.4
Plant	66.3	90.2	90.2	90.3	90.6
Space	50.7	74.1	77.9	78.9	79.9
MEDIA	59.8	78.4	81.7	83.5	85.3

Tabla 10.1. Resultados obtenidos por la nueva metodología de autoarranque aplicada sobre el British National Corpus después de cuatro autoarranques con tres homógrafos diferentes. Los resultados muestran la precisión total en todo el corpus (siempre al 100% de recall) en los cuatro autoarranques.

los tres homógrafos *drug*, *plant* y *space* como palabras objetivo. A los contextos que no obtuvieron clasificación, es decir, que no se etiquetaron, entre los que se incluyen todos los documentos que aportaron un único contexto y varios de los documentos con más de un contexto, se les asignó simplemente la etiqueta de sentido que les atribuye el algoritmo de Yarowsky standard al aplicarse sobre todo el corpus de la forma habitual, como en la Tabla 8.4. Esta información se incluye en la Tabla 10.1 y en la Figura 10.4 desde el primer momento, es decir, desde el primer autoarranque se tiene en cuenta un 100% de recall. Como en las Tablas 8.3 y 8.4, la columna 'base' de la Tabla 10.1 recoge la línea de base de referencia dada por el porcentaje de contextos del sentido mayoritario de cada homógrafo.

Como se puede ver en la Figura 10.4, el comportamiento de los valores de precisión alcanzados después de varios autoarranques es bastante diferente en función del

homógrafo desambiguado. En el caso del homógrafo *drug* la precisión aumenta casi linealmente con los sucesivos autoarranques, mientras que con *plant* el resultado final después de cuatro autoarranques se alcanza ya casi desde el primer autoarranque. El homógrafo *space* representa un caso de comportamiento intermedio.

Esta diferencia en los comportamientos se puede explicar por el grado de ambigüedad de los homógrafos: *plant* sería el homógrafo menos ambiguo de los tres, al menos en las muestras del corpus BNC, y sólo un autoarranque es suficiente para decidir el sentido homográfico de casi todos los documentos. En el otro extremo estaría *drug* como el más ambiguo de los tres: relativamente pocos documentos se deciden en el primer autoarranque, lo que querría decir que los documentos *BNC/drug* son intrínsecamente más ambiguos y por tanto más difíciles de decidir. Sin embargo, esto no impide que decidan muchos más documentos en los siguientes autoarranques. En el caso intermedio *space* se decide un número medio de documentos en el primer autoarranque y lo mismo ocurre en los autoarranques sucesivos.

Esta hipótesis del grado de ambigüedad de los homógrafos queda confirmada por los resultados de precisión obtenidos al desambiguar los mismos tres homógrafos aplicando el algoritmo de Yarowsky standard sobre todo el corpus, reflejados en la Tabla 8.4. En ella se puede ver que *drug* es el más ambiguo al presentar la precisión más baja, *plant* es el menos ambiguo con la precisión más alta y *space* está en el medio con una precisión intermedia. Esto significa que cuanto mayor es la eficacia del algoritmo de Yarowsky standard para un homógrafo determinado, menos necesaria es la nueva metodología de autoarranque.

Estos resultados se refuerzan con los de la Tabla 5.2 sobre el corpus utilizado por Yarowsky en 1995 (véase la sección 5.7) y los de de la Tabla 8.3 sobre el corpus WSJ. En ambos casos los peores resultados son los de *space*, pero muy cerca de *drug*, y los mejores son los de *plant*. El hecho de que *space* esté por debajo de *drug* puede deberse al diferente tipo de corpus, un corpus periodístico en el caso de Yarowsky y del WSJ frente a un corpus equilibrado de texto general en el caso del BNC. También hay que tener en cuenta que los dos homógrafos están muy próximos entre sí y la diferencia es de sólo un 1.4%.

Por lo tanto, esta hipótesis sugiere que hay un grado intrínseco de ambigüedad en un documento considerado como un conjunto de apariciones de la palabra objetivo y sus contextos, es decir, considerado como un par específico documento/homógrafo. Sin embargo, se podría argumentar que una parte de ese grado de ambigüedad es inherente al par corpus/homógrafo y se encuentra localizado en ciertos contextos especialmente ambiguos, mientras que otra parte se le podría adjudicar a la propia metodología de autoarranque en sí, y por tanto podría estar sujeta a algún tipo de optimización. Esta fracción de la ambigüedad representada por una parte de los documentos sobre los que finalmente no se ha decidido su tema homográfico, se debería a aquellos documentos que no tienen por qué tener una serie de contextos especialmente ambiguos, algo que los

incluiría en la otra fracción, pero sí que tienen un cambio de dominio, o un cambio de tema homográfico, dentro de ellos. Este cambio de dominio haría que la metodología de autoarranque fuera incapaz de decidir su tema homográfico debido a que se encuentra mezclado. Sería necesaria una futura línea de investigación para separar estos documentos entre sus partes de tema homográfico homogéneo y como consecuencia lograr mejorar el nivel de precisión logrado por el algoritmo.

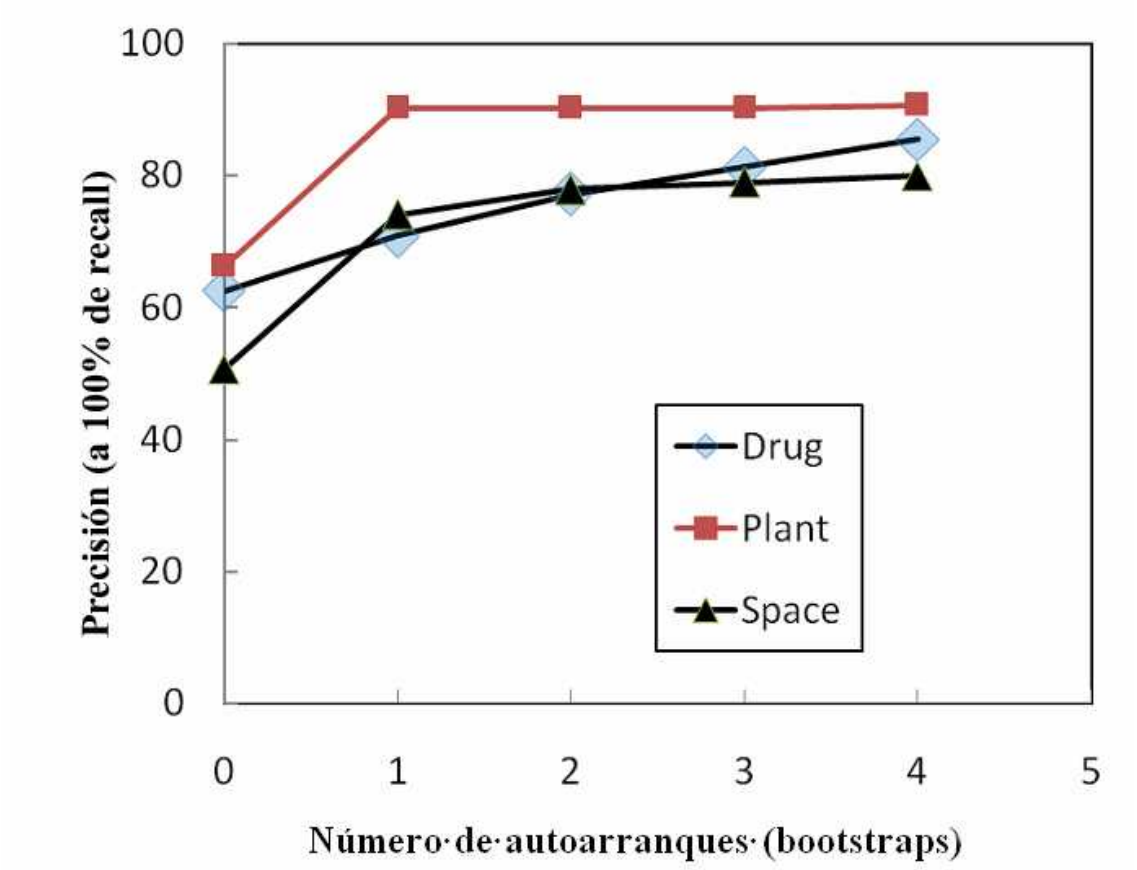


Figura 10.4 .Muestra la precisión (al 100% de *recall*) obtenida por la nueva metodología de autoarranque aplicada sobre el British National Corpus (Tabla 10.1). El valor de la precisión crece con cada Nuevo autoarranque. El experimento se llevó a cabo para cuatro autoarranques con tres homógrafos diferentes.

10.2.3 Trascendencia del algoritmo de autoarranque

Como se apuntó en las secciones 10.1 y 10.1.1, los algoritmos de aprendizaje semisupervisados padecen dos problemas exclusivos y relacionados entre si: el ajuste de parámetros y la proliferación de ejemplos mal etiquetados; también presentan un problema común a todos los algoritmos de DSP: los cambios de dominio. El algoritmo de autoarranque introducido en este capítulo podría constituir la base de una metodología sólida para superar estos tres inconvenientes.

La utilización de un algoritmo de decisiones binarias hace que el sistema sea auto correctivo, ya que el módulo binario es capaz de decidir si se mantiene o no un ejemplo

recién etiquetado de un documento. A este respecto, en realidad no es el propio ejemplo lo que se guarda o no, sino más bien las colocaciones generadas por él en su lista de decisión. Esto hace que el sistema se libere de la proliferación de ejemplos mal etiquetados que es probablemente el problema más importante de los algoritmos semisupervisados. Además este control es automático, sin necesidad de ninguna intervención exterior manual, siguiendo el espíritu de los algoritmos semisupervisados.

Este sistema también puede ser muy útil en la cuestión del ajuste de parámetros, porque la proporción de éxitos en las decisiones binarias sirve como información de realimentación de cómo de bueno es el ajuste actual, con lo cual se obtiene una forma de afinarlo automáticamente. En general, esta proporción de éxitos no debería ser demasiado alta, lo cual significaría que se están tomando decisiones equivocadas, pero tampoco demasiado baja, pues esto indicaría que el sistema está siendo demasiado estricto y muchos documentos no se están decidiendo, con el consiguiente derroche de recursos. La investigación de la proporción óptima de éxitos en estas decisiones dadas sólo las entradas invariantes del algoritmo puede conformar una línea de investigación futura que surge con naturalidad a partir del presente trabajo..

Finalmente, como han sugerido He y Gildea (véase la sección 10.1.1) los algoritmos semisupervisados parecen buenos candidatos para abordar el problema de las fluctuaciones de dominio presentes en casi todos los corpus de texto general. En [Sánchez de Madariaga, Paice, Rayson y Fernández del Castillo, 2008] se ha demostrado que este es un problema genérico que afecta de la misma forma a todos los corpus de texto general (no periodístico) y no sólo a los algoritmos de DSP semisupervisados, sino también a los supervisados. El algoritmo semisupervisado que se propone en este capítulo es un algoritmo 'perfecto a nivel de la propiedad OSPD' en el sentido de que si los documentos del corpus están segmentados correctamente según sus temas, el algoritmo es capaz de producir resultados tan buenos como lo permite la propiedad OSPD, que alcanza una precisión de 99.8% y una aplicabilidad de 50.1% (véase la sección 5.2). Aquí el único obstáculo se debe precisamente a la aplicabilidad: en aquellos documentos donde la palabra objetivo ocurre sólo una vez; y también a un número bastante pequeño de documentos especialmente ambiguos que, aunque puedan ser monotemáticos, exhiben tanta ambigüedad que el algoritmo es incapaz de decidir su tema homográfico.

Se debe notar que, como consecuencia de su carácter binario, ningún otro algoritmo le puede superar por encima del nivel de la restricción OSPD, siempre que la segmentación temática del documento sea correcta, y con la única posible excepción de un algoritmo de DSP hipotético tan eficaz que fuera capaz de desambiguar, dinámicamente, ese pequeño conjunto de documentos especialmente ambiguos que aparecen en casi todos los corpus.

Capítulo 11. Conclusiones y líneas de investigación futuras

11.1. Conclusiones

De entre los resultados del presente trabajo se desprenden una serie de conclusiones relevantes que se muestran a continuación. Se las presenta agrupadas, asociadas al elemento sobre el que mayor incidencia tienen:

El Corpus

Los corpus de texto general presentan fluctuaciones de sus *fuentes de dominio*. Este es un hecho comúnmente aceptado en la literatura sobre PLN en general y sobre DSP en particular. En un intento por explicar el comportamiento inferior en resultados de los algoritmos semisupervisados de autoarranque, y posteriormente también de los algoritmos plenamente supervisados, en los corpus de texto general que en los corpus de texto periodístico, se ha investigado la distribución estadística de las fuentes de dominio en ambos tipos de corpus.

Esta investigación ha probado que la distribución de las fuentes de dominio es diferente en los dos tipos de corpus: está concentrada tópicamente en distintas partes en un corpus de texto general (como el BNC) y está repartida homogéneamente a lo largo de todo el corpus en un corpus de texto periodístico (como el corpus WSJ). Este resultado está respaldado por tres tipos de medidas de la 'ambigüedad' de muestras de los corpus: ambigüedad simple o autoambigüedad de las muestras, calculada como media de la precisión de cuatro experimentos de DSP supervisados, cada uno con un algoritmo empotrado de aprendizaje automático cualitativamente muy diferente, a saber: red neuronal RBF, red bayesiana, tabla de decisión y árbol J48; ambigüedad cruzada, cuya diferencia con la autoambigüedad consiste en que los conjuntos de entrenamiento y test del experimento de DSP supervisado pertenecen a muestras del corpus diferentes, en vez de pertenecer ambos a la misma muestra; y ambigüedad compuesta, en la que la

medida de ambigüedad se realiza sobre composiciones de muestras dos a dos o todas juntas en un solo corpus conjunto.

La autoambigüedad lo demuestra en cierta medida en que sus resultados de precisión son más altos que los de la ambigüedad cruzada, algo que cabía esperar con bastante naturalidad, y en que esta diferencia es mayor para el BNC que para el WSJ. Un resultado colateral, pero también interesante, es que la diferencia es mayor para unas palabras objetivo (palabras bajo desambiguación) que para otras, indicando que las palabras tienen un grado de ambigüedad diferente entre sí. En los experimentos llevados a cabo en este trabajo las palabras objetivo fueron *drug* y *space*, que son palabras homográficas, es decir, fuertemente polisémicas, tanto en inglés británico (BNC) como en inglés norteamericano (WSJ). Los experimentos indican que *drug* es un homógrafo más ambiguo que *space*.

Los resultados de la ambigüedad cruzada prueban con mucho mayor énfasis la diferencia entre las distribuciones de las fuentes de dominio en ambos tipos de corpus. En el BNC se han tomado muestras según el dominio y en el WSJ se han tomado muestras de documentos consecutivos cronológicamente. En ambos corpus, y para los dos homógrafos, la medida de ambigüedad cruzada ha dado lugar a la formación de dos clusters de muestras similares. La diferencia entre la ambigüedad cruzada media de las muestras del mismo cluster (lo que en inglés se llamaría intra-cluster) y de las muestras de clusters diferentes (en inglés inter-cluster) es mucho mayor en el BNC que en el WSJ. En concreto, para el BNC y *space* la diferencia es de un 10% y para BNC/*drug* es de un 32%; mientras que para WSJ/*space* la diferencia es de sólo un 2% y para WSJ/*drug* de un 1%. Es decir, hay un gran cambio de dominio entre las muestras del BNC de diferentes clusters mientras que ese cambio no existe entre las muestras del WSJ. Esto indica que la distribución de las fuentes de dominio está concentrada en ciertas partes en el BNC y es homogénea en todo el WSJ.

La ambigüedad de la composición de muestras dos a dos confirma la existencia de estos cambios de dominio en el BNC, ya que al componer muestras de clusters diferentes se están mezclando dominios distintos y la precisión resultante baja (es decir, la ambigüedad total sube). Para BNC/*space* y BNC/*drug*, la diferencia entre la ambigüedad compuesta media entre clusters diferentes y dentro del mismo cluster es de 2.6% y 10.4% respectivamente, mientras que para WSJ/*space* y WSJ/*drug* estas diferencias se reducen a 0.0% y 0.3%, respectivamente.

Estos resultados confirman también que *drug* tiene un grado mayor de ambigüedad que *space*, algo que una vez más también se puede verificar con los propios resultados del algoritmo de Yarowsky sobre todo el BNC: la precisión de *space* es un 3.6% más alta que la de *drug*, y esta diferencia sube hasta un 15.4% si se tiene en cuenta la referencia de base del sentido más frecuente.

Los resultados de la ambigüedad aditiva de muestras compuestas dos a dos y de todas las muestras juntas también permiten estudiar la influencia del incremento del tamaño del corpus sobre la precisión o la ambigüedad. Se puede deducir que coexisten dos fenómenos contradictorios mientras aumenta el tamaño de los corpus: por un lado se están incrementando también los conjuntos de entrenamiento de los algoritmos de DSP supervisados, lo cual contribuye a que aumente la precisión del experimento; pero por otro lado se está mezclando dominios diferentes, sobre todo en el caso del BNC, lo cual contribuye en sentido contrario al resultado de precisión del experimento. En total se tiene que para BNC/*drug* hay una caída considerable de precisión, para BNC/*space* una caída leve, debida a que *space* es menos ambiguo que *drug*, y para WSJ/*drug* y WSJ/*space* casi no hay mezcla de dominios y el incremento de los conjuntos de entrenamiento hace que haya en total un incremento considerable de la precisión en ambos casos.

Por todo esto, se puede concluir que la distribución estadística de las fuentes de dominio es muy diferente en un corpus de texto general, donde están concentradas en ciertos lugares y en un corpus de texto periodístico, donde se distribuyen homogéneamente. Sin embargo, esto no tiene por qué significar que no haya variaciones de dominio en los corpus de texto periodístico. En el caso del corpus WSJ frente al BNC podría decirse que la principal diferencia a primera vista entre los dos corpus reside en que los documentos del BNC, mucho más grandes, están clasificados según dominios (y según algunos otros criterios, como medios de publicación), mientras que los documentos del WSJ no están clasificados. Pero seguramente que sí hay variaciones de dominio: es evidente que los artículos periodísticos, aunque muy breves, sí tratan sobre temas o dominios diferentes, como política, ciencia o tecnología. Por lo tanto, las variaciones de dominio existen en los dos tipos de corpus, lo único que los diferencia es su distribución estadística. Entonces, si la única diferencia es la distribución inicial, y al juntar todas las muestras todos esos dominios se entremezclan, debe haber alguna razón distinta para que el comportamiento de la medida de la ambigüedad en ambos tipos de corpus sea tan diferente con el aumento de tamaño del corpus.

De estos datos se sigue que la razón detrás de este comportamiento diferente está en que los corpus de noticias periodísticas constituyen un tipo muy especial de corpus, que utiliza un lenguaje también especial, estereotipado y repetitivo, que se refiere una y otra vez a las mismas situaciones del mundo real durante períodos relativamente largos de tiempo, que pueden durar meses y años. Esto hace que al final el lenguaje empleado sea repetitivo, si no en vocabulario, sí en usos y referencias a situaciones similares que acaban siendo estereotipadas.

A este respecto vale la pena referirse brevemente a los corpus utilizados en los certámenes Senseval/Semeval. Como ya se ha indicado, estas competiciones constituyen hasta ahora, en sus cuatro ediciones, un éxito sin paliativos, en cuanto a evaluación y mejora de los algoritmos de DSP, que no han hecho sino aumentar su eficacia medida como precisión desde el primer momento. Sin embargo, sería

interesante prestar atención al tipo de corpus utilizados en ellas. En general, se puede afirmar que los corpus se han seleccionado de una forma bastante informal y casi siempre con una muy alta proporción de componentes de texto periodístico. A la luz de los experimentos de este trabajo, esto significaría que los resultados obtenidos por los algoritmos sobre esos corpus no son completamente representativos del texto escrito en general y además esos resultados seguramente son más altos de los que se obtendrían en corpus realmente representativos, similares por ejemplo al BNC. Por lo tanto parece necesario el prestar mayor atención a la selección de los corpus y a un estudio relativamente sistemático de la naturaleza de los diferentes corpus, no sólo en los foros Senseval/Semeval sino también en la literatura sobre DSP en general.

Las fluctuaciones de las fuentes de dominio afectan tanto a los algoritmos de DSP supervisados como a los algoritmos de DSP de autoarranque semisupervisados, si bien las diferencias entre utilizar un corpus de texto general y un corpus de texto periodístico son más acentuadas para los últimos. Es un hecho aceptado en la literatura que la principal ventaja a priori de los algoritmos semisupervisados sobre los plenamente supervisados está en la liberación del grave problema representado por el CBAC. Desde su aparición en 1995 Yarowsky reclamó para su algoritmo la igualdad de condiciones en cuanto a precisión frente a los algoritmos supervisados, al menos para desambiguación de palabras con polisemia fuerte o de nivel homográfico, es decir, de dos sentidos. Sin embargo, la literatura hasta ahora no se había dado cuenta de que el resultado tan alto de los algoritmos semisupervisados, en competencia directa con los supervisados, se debía en gran medida al uso generalizado de corpus de texto periodístico.

El algoritmo

El nuevo algoritmo de autoarranque semisupervisado propuesto en este trabajo ha tenido como primer objetivo la resolución del problema del CBAC en las aplicaciones que utilizan corpus de texto general, es decir, la competencia en igualdad de condiciones en términos de precisión entre los algoritmos semisupervisados y los plenamente supervisados en ese tipo de corpus de la vida real. Un segundo objetivo ha sido aprovechar lo especial del mecanismo de autoarranque de los algoritmos semisupervisados, que les otorga una gran flexibilidad, para llegar a superar la precisión de los algoritmos plenamente supervisados, al menos en la tarea concreta de la DSP, haciendo un fuerte uso de la propiedad OSPD.

El nuevo algoritmo también ha tenido que hacer frente a los problemas exclusivos de este tipo de algoritmos, el ajuste de parámetros y la proliferación de etiquetados falsos, para poder funcionar a un nivel práctico.

La causa principal de la bajada en la precisión lograda por todos los algoritmos de DSP sobre corpus de texto general, independientemente de los corpus de texto periodístico, que pueden considerarse especiales y excepcionales, es la mezcla de

dominios diferentes provocada cuando el tamaño del corpus es suficientemente grande como para que se sobrepase el límite entre dominios diferentes.

Por lo tanto idealmente se debería intentar evitar esa mezcla de dominios, ejecutando el algoritmo de forma aislada en cada trozo de corpus con un dominio homogéneo. Si esa posibilidad parece muy difícil sin verse afectado por el CBAC en un algoritmo supervisado, e incluso en un algoritmo semisupervisado como el de Yarowsky, ya que habría que encontrar semillas iniciales para muchas partes del corpus, con lo cual estaríamos ante un algoritmo supervisado *de facto*, se pueden aprovechar las características especiales del mecanismo de autoarranque de un algoritmo semisupervisado en combinación con la propiedad OSPD para lograrla.

En el nuevo algoritmo esto se ha conseguido lanzando un autoarranque independiente en cada parte de corpus, no necesariamente separado en segmentos temáticamente homogéneos al 100%, y decidiendo de una forma binaria su *tema homográfico* de acuerdo con la propiedad OSPD cumplida por los dos sentidos del homógrafo correspondiente. Para llevar a cabo esta decisión se ha aplicado una sencilla fórmula que utiliza las puntuaciones logradas por las ocurrencias del homógrafo en el documento después del autoarranque del algoritmo y cuyo resultado se compara de forma binaria con un umbral predefinido.

El hecho de que se produzcan autoarranques independientes en trozos de corpus del tamaño deseado, que puede ser un documento en el caso del BNC, hace que durante esos autoarranques no se produzca la mezcla de dominios heterogéneos con la consiguiente pérdida de precisión. Además, la utilización del clasificador (lista de decisión) obtenido en un determinado documento en el autoarranque ulterior de otros documentos hace que sólo sea necesario inicializar el algoritmo con semillas una sola vez, e incluso con sólo dos etiquetas, una para cada sentido, de forma que el algoritmo sigue estando muy débilmente supervisado y se pueda suprimir el problema del CBAC. Finalmente, la alta fiabilidad de la propiedad OSPD (alrededor del 99% en corpus de texto periodístico, quizás algo menor en texto general) hace que la precisión final del algoritmo sea como mínimo comparable a la de sus competidores plenamente supervisados, incluso teniendo en cuenta que finalmente no se pueden decidir por este método los temas homográficos de todos los documentos o trozos de corpus tratados individualmente, bien porque esos trozos no decididos no sean completamente homogéneos respecto a su tema homográfico, o bien porque se trate de documentos especialmente ambiguos, que serían difícilmente determinables por cualquier algoritmo desambiguador, por muy eficaz que fuese.

La utilización de las listas de decisión logradas en un documento individual cuyo tema homográfico haya sido decidido con éxito en el autoarranque de los documentos que todavía no hayan sido decididos, hace que esas listas de decisión individuales y dinámicas vayan cambiando paulatinamente de dominio, de forma que se pueda autoarrancar muchos documentos del corpus, aunque existan variaciones, primero

leves pero al final drásticas, entre los dominios de esos documentos. De esta forma estamos aprovechando la flexibilidad del mecanismo de autoarranque para lograr desambiguar un corpus con variaciones de dominio sin mezclarlos. En la Tabla 11.1 se puede ver un resumen de algunos resultados de precisión obtenidos por el algoritmo de Yarowsky y algoritmos supervisados en el WSJ y el BNC y por el nuevo algoritmo semisupervisado en el BNC. Los resultados se han calculado en todos los casos como media a partir de los resultados obtenidos para los homógrafos *space* y *drug* y son por tanto directamente comparables. Como se puede observar, el algoritmo de Yarowsky en el WSJ obtiene una precisión del 90.9%. Como se ha señalado en los capítulos precedentes, este resultado, que coincide exactamente con el de Yarowsky en 1995, se obtiene sin aplicar la propiedad OSPD, y si ésta se aplicara, habría que añadir un 4% de precisión aproximadamente, lo que daría un 94.9%, es decir, alrededor de un 95%. Convengamos en que ese valor de precisión es óptimo para aplicaciones reales y es por tanto equivalente a un 100%. También se podría argumentar con bastante credibilidad que el 89.3% obtenido en ese mismo corpus por la media de varios algoritmos plenamente supervisados se podría incrementar hasta un 95% si se eligieran los algoritmos supervisados apropiados, en vez de calcular una media. En cambio, en el BNC la media de los algoritmos supervisados, en vez de subir con el tamaño del corpus como en el WSJ, disminuye, hasta un 80.9%. Si le incrementamos análogamente un 4 ó un 5%, llegamos hasta un 85% aproximadamente. Pero también se podría argumentar que si se optimizara la segmentación de los documentos del BNC en trozos temáticamente homogéneos, se le podría añadir un pequeño porcentaje de un 4 ó 5% a la precisión obtenida en el BNC por el nuevo algoritmo de autoarranque semisupervisado, que ha obtenido un 82.7%. En otras palabras, los resultados de esa tabla para el nuevo algoritmo y los algoritmos supervisados son comparables, es decir, el nuevo algoritmo ha superado de hecho, o puede superar potencialmente, la precisión de los algoritmos supervisados, gracias al uso masivo de la propiedad OSPD en un entorno con fuertes fluctuaciones de dominio.

media space/drug	***media documentos individuales***	***corpus total***
BNC		
Yarowsky		**68.5**
algoritmo nuevo		**82.7**
supervisado	**84.9**	**80.9**
WSJ		
Yarowsky		**90.9**
supervisado	**85.2**	**89.3**

Tabla 11.1 Precisión media de la desambiguación de los homógrafos *space* y *drug* en los corpus BNC y WSJ. Se muestra la precisión del algoritmo de Yarowsky en ambos corpus, la precisión media de cuatro algoritmos supervisados (la precisión media de los documentos individuales de los corpus y la precisión compuesta en los corpus en su totalidad) y la precisión media lograda por el nuevo algoritmo de autoarranque en el BNC para esos dos homógrafos.

Soluciones a problemas con el autoarranque

Finalmente, el nuevo algoritmo también debe abordar los problemas propios de los mecanismos de autoarranque en un entorno de aplicaciones prácticas. El problema de la proliferación de etiquetados falsos se ataja de forma muy importante mediante el algoritmo de decisión binaria, porque cuando no se decide el tema homográfico de un documento, porque no se tiene suficiente seguridad sobre ello, la lista de decisión que se ha formado al autoarrancar ese documento no se mantiene, es decir, se descarta. Esto hace que muchas colocaciones falsas no se apliquen en el intento de autoarrancar nuevos documentos, evitando el problema de la proliferación de etiquetados falsos, que es paralelo al de la generación de colocaciones falsas. Es decir, el algoritmo tiene un sistema para cortar inmediatamente cualquier desviación desde el primer momento y sólo mantener aquellas colocaciones que han acreditado con su éxito en la decisión del tema homográfico de un documento que no son falsas y por tanto no darán lugar a futuros etiquetados falsos ni por supuesto decisiones erróneas del tema de otro documento.

El problema del ajuste de parámetros, que es la causa principal del problema de proliferación de etiquetados falsos en un algoritmo de autoarranque como el de Yarowsky o el de Blum y Mitchell, se ve muy simplificado en el nuevo algoritmo por el coto que el propio algoritmo le pone al segundo problema mediante el mecanismo de decisión binaria. De esta forma, el primer problema se reduce de varios parámetros en aquéllos algoritmos a sólo un parámetro en realidad en éste, a saber, el ajuste del umbral de decisión sobre el tema homográfico de los documentos. Sin embargo, el nuevo algoritmo también dispone de un instrumento eficaz para ajustar automáticamente este parámetro; en realidad se necesita que ese umbral no sea demasiado bajo, porque estaríamos siendo laxos y tomando decisiones erróneas, en un proceso realimentado, pero tampoco demasiado alto, porque estaríamos siendo demasiado estrictos y no tomando decisiones que serían bastante evidentes, perjudicando el recall y malgastando recursos. Por tanto, como necesitamos un punto de equilibrio entre ambos extremos, podemos utilizar como información de entrada la realimentación sobre la proporción de documentos decididos en el pasado en relación al número total de documentos y al número de intentos de decisión. La investigación sobre este punto de equilibrio podría constituir una futura línea de investigación.

Hay que tener en cuenta que la precisión de 85.3% de media para los homógrafos *drug*, *plant* y *space* en el BNC se produce para un *recall* del 100% y que, aunque signifique un incremento de un 10.6% sobre el algoritmo de Yarowsky en el mismo corpus, en muchos documentos el algoritmo en realidad no se puede aplicar: se trata de los documentos donde no hay más de dos ocurrencias de la palabra objetivo, y se recurre al algoritmo de Yarowsky; en los documentos donde el nuevo algoritmo sí es aplicable la precisión siempre es cercana al 95%.

También sería interesante resaltar que este algoritmo es un algoritmo *perfecto al nivel de la restricción OSPD*, es decir, si la *segmentación temática homográfica* de los

documentos fuera perfecta, ningún otro algoritmo le podría superar por encima de la precisión de la propiedad OSPD, que es muy alta (alrededor de 99% en corpus de texto periodístico, quizás un poco menor en texto general). Esto apunta una futura línea de trabajo sobre el estudio de la forma de obtener una segmentación óptima de los documentos según los temas de cada homógrafo en particular, para optimizar el algoritmo. La determinación de los temas homográficos de documentos o de trozos de texto según diferentes homógrafos puede ser muy interesante desde el punto de vista de la Recuperación de la Información (RI).

Los experimentos con los tres homógrafos importantes que se han usado para evaluar el nuevo algoritmo, *drug*, *plant* y *space*, han probado también que el comportamiento medido como precisión después de varios autoarranques varía considerablemente de un homógrafo a otro. La precisión de *drug* crece casi linealmente con cada nuevo autoarranque, mientras que *plant* alcanza su precisión final casi desde el primer autoarranque y *space* representa un caso intermedio. Todo esto ha indicado que *plant* es el homógrafo menos ambiguo, *drug* es el más ambiguo y *space* está en el medio, resultado que se ha visto confirmado por la precisión de estos homógrafos con el algoritmo de Yarowsky original en el mismo corpus. Es decir, los homógrafos tendrían un grado de ambigüedad intrínseco que va desde homógrafos muy fuertes como *plant* hasta homógrafos más ambiguos como *drug*. Según ese grado de ambigüedad se deduce que cuanto menos ambiguo es el homógrafo, menos necesario sería el nuevo algoritmo de autoarranque, ya que el algoritmo de autoarranque estándar se comportaría mejor, y en caso de necesitar aplicarlo, sería necesario un menor número de autoarranques sucesivos.

11.2. Líneas de investigación futuras

La necesidad de alcanzar un punto de equilibrio para el umbral de decisión del tema homográfico de un documento es una cuestión interesante que surge con naturalidad como línea de investigación inmediata. Este umbral no debe ser demasiado bajo, lo que provocaría decisiones equivocadas que además afectarían al desarrollo correcto ulterior del algoritmo, pero tampoco demasiado alto, ya que al ser demasiado exigentes en la prueba del tema del documento se correría el riesgo de dejar de tomar decisiones que hubieran sido correctas siendo algo menos estrictos. Esto naturalmente afecta al *recall* final del algoritmo y en realidad se estarían malgastando recursos.

Como en todo momento se conoce el número de decisiones tomadas en cada uno de los dos sentidos, el número de decisiones intentadas y el número total de documentos del corpus, se puede utilizar esta información actualizada para decidir si se está siendo demasiado laxo o demasiado estricto, y corregir el umbral en la dirección adecuada. Esta determinación del umbral a partir de esas proporciones sería en principio un proceso empírico, y debería probarse en diferentes corpus de texto general para intentar encontrar unas regularidades constantes en todos los corpus. Más adelante se podría

extrapolar a otros corpus más específicos y comprobar si esas regularidades se siguen cumpliendo o si habría que fijar criterios especiales para tipos de corpus específicos.

Otra línea de trabajo que surge lógicamente es la optimización del algoritmo de decisión binario en un corpus determinado hasta el nivel de la restricción OSPD dividiendo los documentos en segmentos homogéneos desde el punto de vista de los dos sentidos o temas homográficos de la palabra objetivo. Es decir, si logramos que no haya ningún trozo de corpus (documento) autoarrancado individualmente que contenga variaciones del tema homográfico, habremos conseguido que casi todos los documentos se decidan correctamente. Sólo quedaría un número muy reducido de documentos no decididos debido a su elevado grado de ambigüedad intrínseca, no a la mezcla de temas homográficos. Esto haría que el algoritmo estuviera optimizado al nivel de la propiedad OSPD salvo esos documentos excepcionales, y el resultado total de precisión sería muy elevado.

La determinación de los temas homográficos de una muestra de texto general para varios homógrafos diferentes podría ser muy interesante desde el punto de vista de la Recuperación de la Información. Hay que tener en cuenta que el número de homógrafos en un lenguaje natural es muy alto; como hemos visto, los homógrafos tienen un determinado grado de ambigüedad intrínseco, y en realidad el número de homógrafos es arbitrariamente alto. En [Durkin y Manning 1989] se presenta una lista de 150 homógrafos importantes del idioma inglés. Por tanto, podría ser muy interesante poder determinar los temas homográficos de muestras de texto o documentos dados desde el punto de vista de muchos homógrafos diferentes. De entre todos esos temas homográficos, los correspondientes a unos pocos homógrafos elegidos adecuadamente en relación a una consulta dada podría ser una información valiosísima para determinar la relevancia de ese documento respecto a esa consulta desde el punto de vista de la Recuperación de la Información.

Capítulo 12. Bibliografía

[Abney 2002] Abney, S. 2002. Bootstrapping. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA, 360-367.

[Abney 2004] Abney, S. 2004. Understanding the Yarowsky algorithm. *Computational Linguistics*, 30(3): 365-395.

[Agirre y Edmonds 2006] Agirre, E. y P. Edmonds. 2006. Introduction to *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (Eds.) Springer Dordrecht The Netherlands.

[Agirre y Lopez de Lacalle 2004] Agirre, E. y O. López de Lacalle. 2004. Publicly available topic signatures for all WordNet nominal senses. *Proceedings of the 4th Internacional Conference on Language Resources and Evaluations (LREC)*. Lisbon, Portugal.

[Agirre y Martinez 2000] Agirre, E. y D. Martínez. 2000. Exploring automatic word sense disambiguation with decisión lists and the Web. *Proceedings of the Semantic Annotation and Intelligent Annotation Workshop*, organizad by COLING. Luxembourg, 11-19.

[Agirre y Martinez 2001b] Agirre, E. y D. Martínez. 2001b. Learning class-to-class selectional preferentes. *Proceedings of the ACL/EACL Workshop on Computacional Natural Language Learning (CoNLL)*, Toulouse, France.

[Agirre y Martinez 2004a] Agirre, E. y D. Martínez. 2004a. The Basque Country University system: English and Basque tasks. *Proceedings of Senseval-3: Third*

Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain, 44-48.

[Agirre y Martinez 2004b] Agirre, E. y D. Martínez. 2004b. Smoothing and word sense disambiguation. *Proceedings of España for Natural Language Processing (EsTAL)*, Alicante, Spain, 360-371.

[Agirre y Martinez 2004c] Agirre, E. y D. Martínez. 2004c. Unsupervised WSD base don automatically retrieved examples: the importante of bias. *Proceedings of the 10th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain, 25-32.

[Agirre y Stevenson 2006] Agirre, E. y M. Stevenson. 2006. Knowledge sources for WSD. *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (eds.). Dordrecht. Springer.

[Agirre et al. 2000] Agirre, E., O. Ansa, E. Hovy y D. Martínez. 2000. Enriching very large ontologies using the WWW. *Proceedings of the Ontology Learning Workshop, European Conference on Artificial Intelligence (ECAI)*, Berlin, Germany.

[Agirre et al. 2001] Agirre, E., O. Ansa, D. Martínez y E. Hovy. 2001. Enriching WordNet concepts with Tepic signaturas. *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA.

[Antal 1965] Antal, L. 1965. *Content, Meaning and Understanding*. The Hague: Mouton.

[Atkins 1993] Atkins, S. 1993. Tools for computer-aided corpus lexicography: The Hector Project. *Acta Linguistica Hungarica*, 41: 5-72.

[Baker et al. 2003] Baker, C.F., C.J. Fillmore y B. Cronin. 2003. The structure of the FrameNet database. *Internacional Journal of Lexicography*, 16(3): 281-296.

[Baldridge y Osborne 2003] Baldridge, J. & Osborne, M. (2003). Active learning for HPSG parse selection. *Proceedings of the 7th Conference on Natural Language Learning*, Edmonton, Canada.

[Baldridge y Osborne 2004] Baldridge, J. & Osborne, M. (2004). Active learning and the total cost of annotation. *Proceedings of EMNLP 2004*, Barcelona, Spain.

[Ballesteros y Croft 1997] Ballesteros, L. y W.B. Croft. 1997. Phrasal translation and query expansion techniques for cross-language information retrieval. *Proceedings of the 20th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia, PA, 84-91.

[Banerjee y Pedersen 2003] Banerjee, S. y T. Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. *Proceedings of the 18th Internacional Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 805-810.

[Basili et al. 1997] Basili, R., M. d. Rocca y M.T. Pazienza. 1997. Contextual word sense tuning and disambiguation. *Applied Artificial Intelligence*, 11: 235-262.

[Basili et al. 2004] Basili, R., D.H. Hansen, P. Paggio, M.T. Pazienza y F.M. Zanzotto. 2004. Ontological resources and question answering. *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*, Boston.

[Berger 1996] Berger, A., S.D. Pietra y V.D. Pietra. 1996. A maximum entropy approach to natural language processing. *Computacional Linguistics*, 22(1): 39-72.

[Berners-Lee et al. 2001] Berners-Lee, T., J. Hendler y O. Lassila. 2001. The Semantic Web. *Scientific American*, 284(5): 34-43.

[Bhattacharya et al. 2004] Bhattacharya, I., L. Getoor y Y. Bengio. 2004. Unsupervised word sense disambiguation using bilingual probabilistic models. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain, 288-295.

[Bloehdorn y Hotho 2004] Bloehdorn, S. y A. Hotho. 2004. Text classification by boosting weak learners based on terms and concepts. *Proceedings of the Fourth IEEE Internacional Conference on Data Mining*, Brighton, UK, 331-334.

[Blum y Mitchell 1998] Blum, A. y T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. *Proceedings of the 11th Annual Conference on Computational Learning Theory (CoLT)*, 92-100.

[Boser et al. 1992] Boser, B.E., I.M. Guyon y V.N. Vapnik. 1992. A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual Workshop on Computacional Learning Theory (CoLT)*, Pittsburg, USA, 144-152.

[Breiman 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.

[Brill 1993] Brill, E. 1993. A Corpus-based Approach to Language Learning. Ph.D. Thesis, University of Pennsylvania, 1993.

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

[Brown et al. 1990] Brown, P.F., J. Cocke, S.A. della Pietra, V.J. della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer y P.S Rocín. 1990. A statistical approach to machine translation. *Computacional Linguistics*, 16(2): 79-85.

[Brown 1996] Brown, R.D. 1996. Example-based machine translation in the Pangloss system. *Proceedings of the 16th Internacional Conference on Computacional Linguistics (COLING)*, Copenhagen, Denmark, 169-174.

[Bruce y Wiebe 1994] Bruce, R. y J. Wiebe. 1994. Word sense disambiguation using decomposable models. *Proceedings of the 32nd Annual Meeting of the Association for Computacional Linguistics (ACL)*, Las Cruces, USA, 139-146.

[Bruce y Wiebe 1999] Bruce, R. y J. Wiebe. 1999. Recognizing subjectivity: a case study in manual tagging. *Natural Language Engineering*, 5(2):187-205.

[Buitelaar y Sacaleanu 2001] Buitelaar, P. y B. Sacaleanu. 2001. Ranking and selecting synsets by domain relevance. *Proceedings of the Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA.

[Buitelaar y Sacaleanu 2002] Buitelaar, P. y B. Sacaleanu. 2002. Extending synsets with medical terms. *Proceedings of the First Internacional WordNet Conference*, Mysore, India.

[Buitelaar et al. 2006] Buitelaar, P., B. Magnini, C. Strapparava y P. Vossen. 2006. Domain-specific WSD. *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (Eds.). Dordrecht. Springer.

[Butadinsky y Hirst 2001] Butadinsky, A. y G. Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*, Pittsburg, USA, 29-34.

[Cabezas et al. 2004] Cabezas, C., I. Bhattacharya y P. Resnik. 2004. The University of Maryland Senseval-3 system descriptions. *Proceedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 83-87.

[Cabezas y Resnik 2005] Cabezas, C. y P. Resnik. 2005. *Using WSD Techniques for Lexical Selection in Statistical Machine Translation.* Technical report CS-TR-4736/LAMP-TR-124/UMIACS-TR-2005-42, July 2005. (http://lampsrv01.umiacs.umd.edu/pubs/TechReports/LAMP_124/LAMP_124.pdf).

[Calzolari y Corazzari 2000] Calzolari, N. y O. Corazzari. 2000. Senseval/Romanseval: The framework for italian. *Computers and the Humanities*, 34(1-2): 61-78.

[Carpuat et al. 2004] Carpuat, M., W. Su y D. Wu. 2004. Augmenting ensemble classification for word sense disambiguation with a kernel PCA model. *Proceedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 88-92.

[Carpuat y Wu 2005a] Carpuat, M. y D. Wu. 2005a. Word sense disambiguation vs. Statistical machine translation. *Proceedings of the 43rd Annual Meeting of the Association for Computacional Linguistics (ACL)*, Ann Arbor, MI, June, 387-394.

[Carpuat y Wu 2005b] Carpuat, M. y D. Wu. 2005b. Evaluating the word sense disambiguation performance of statistical machine translation. *Proceedings of the Second Internacional Joint Conference on Natural Language Processing (IJCNLP)*. Jeju, Korea, October.

[Carroll y Briscoe 2001] Carroll, J. y T. Briscoe. 2001. High precision extraction of gramatical relations. *Proceedings of the 7th ACL/SIGPARSE International Workshop on Parking Technologies*, Beijing, China, 78-89.

[Chan y Ng 2005] Chan, Y.S. y H.T. Ng. 2005. Scaling up word sense disambiguation via parallel texts. *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, Edinburgh, UK, 1010-1015.

[Chapman 1977] Chapman, R. L. 1977. *Roget's Internacional Thesaurus, Fourth Edition*. Harper and Row, New Cork, USA.

[Chelba y Jelinek 1998] Chelba, C. y F. Jelinek. 1998. Exploiting syntactic stucture for language modeling. Proceedings of the 36th Annual Meeting of the Association for Computacional Linguistics and 17th Internacional Conference on Computacional Linguistics (ACL-COLING), Morgan Kaufman Publishers, San Francisco, California, 225-231.

[Chen y Nie 2000] Chen, J. y J. Nie. 2000. Cross-language information retrieval between Chinese and English. *Proceedings of the International Conference on Chinese Language Computing*, Chicago, USA.

[Chen y Chang 1998] Chen, J. y J. Chang. 1998. Topical clustering of MRD senses based on information retrieval techniques. *Computational Linguistics*, 24(1): 61-95.

[Chiang 2005] Chiang, D. 2005. A hierarchical phrase-based model for statistical machine translation. *Proceedings of the 43rd Annual Meeting of the Association for Computacional Linguistics (ACL)*, Ann Arbor, MI, 263-270.

[Chklovski y Mihalcea 2002] Chklovski, T. y R. Mihalcea. 2002. Building a sense tagged corpus with Open Mind Word Expert. *Proceedings of the workshop on word sense disambiguation: Recent successes and future directions*, Philadelphia, USA, 116-122.

[Chugur, Gonzalo y Verdejo 2002] Chugur, I., J. Gonzalo y F. verdejo. 2002. Polysemy and sense proximity in the Senseval-2 test suite. *Proceedings of the ACL 2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Pennsylvania.

[Ciaramita y Johnson 2004] Ciaramita, M. y M. Johnson. 2004. Multi-component word sense disambiguation. *Proceedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 97-100.

[Clark et al. 2003] Clark, S., J. Curran y M. Osborne. 2003. Bootstrapping POS taggers using unlabelled data. *Proceedings of the 7th Conference on Natural Language Learning (CoNLL)*, Edmonton, Canada, 164-167.

[Clough y Stevenson 2004] Clough, P. y M. Stevenson. 2004. Cross-language information retrieval using EuroWordNet and word sense disambiguation. *Advances in Information Retrieval, 26th European Conference on IR Research (ECIR 2004)*, Sunderland, UK, 327-337.

[Cohen 1960] Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20: 37-46.

[Cohn, Ghahranami y Jordan 1995] Cohn, D., Ghahranami, Z. y Jordan, M. (1995). Active learning with statistical models. *Advances in Neural Information Processing Systems*. In G. Tesauro, D. Touretzky & T. Leen (Eds.), 7, 705-712. The MIT Press.

[Cruse 1986] Cruse, D. 1986. *Lexical Semantics*. Cambridge University Press, Cambridge, UK.

[Cucchiarelli y Velardi 1998] Cucchiarelli, A. y P. Velardi. 1998. Finding a domain-appropriate sense inventory for semantically tagging a corpus. *Natural Language Engineering*, 4(4):325-344.

[Daelemans et al. 1999] Daelemans, W., A. v.d. Bosch y J. Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34: 11-41.

[Dagan y Itai 1994] Dagan, I. y A. Itai. 1994. Word sense disambiguation using a second language parallel corpus. *Computational Linguistics*, 20(4): 563-596.

[Dagan, Glickman y Magnini 2004] Dagan, I., O. Glickman y B. Magnini. 2004. *Recognising Textual Entailment Challenge*. (http://www.pascal-network.org/Challenges/RTE/).

[Dawkins 1986] Dawkins, R. 1986. *The Blind Watchmaker*. W.W. Norton.

[Deerwester et al. 1990] Deerwester, S., S.T. Dumais, G.W. Furnas, T.K. Landauer y R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6): 391-407.

[Diab y Resnik 2002] Diab, M. y P. Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. *Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA, 255-262.

[Diab 2003] Diab, M. 2003. *Word sense disambiguation within a multilingual framework*. Ph.D. Thesis. Department of Linguistics, University of Maryland, Collage Park, Maryland.

[Diab 2004] Diab, M. 2004. Relieving the data acquisition bottleneck in word sense disambiguation. *Proceedings of the 42nd meeting of the Association for Computational Linguistics (ACL)*, Barcelona, Spain, 303-310.

[Dill et al. 2003] Dill, S., N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin y J.Y. Zien. 2003. SemTag and Seeker: Bootstrapping the Semantic Web via automated semantic annotation. *Proceedings of the Twelfth Internacional Conference on World Wide Web (WWW-2003)*, Budapest, Hungary, 178-186.

[Dolan 1994] Dolan, W. 1994. Word sense ambiguation: Clustering related senses. *Proceedings of the 14th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan, 712-716.

[Dorr 1993] Dorr, B.J. 1993. *Machine Translation: A View from the Lexicon*. Cambridge, MA: MIT Press.

[Dorr y Jones 2000] Dorr, B.J. y D. Jones. 2000. Acquisition of semantic lexicons: using word sense disambiguation to improve precision. *Breadth and Depth of Semantic Lexicons*. Viegas (Ed.). 79-98. kluwer Academia Publishers, Norwell, MA.

[Duda et al. 2001] Duda, R. O., P. E. Hart y D. G. Stork. 2001. *Pattern classification, 2nd Edition*. New Cork: John Wiley & Sons.

[Durkin y Manning 1989] Durkin, K. y J. Manning. 1989. Polisemy and the subjective lexicon: semantic relatedness and the saliente of intraword senses. *Journal of Psycholinguistic Research*, 18: 577-612.

[Dyvik 1998] Dyvik, H. 1998. Translations as semantic mirrors. *Proceedings of the ECAI Workshop on Multilingality in the Lexicon II*, Brighton, UK, 24-44.

[Dyvik 2002] Dyvik, H. 2002. Translations as semantic mirrors: From parallel corpus to WordNet. *Advances in Corpus Linguistics. Papers from the 23rd Internacional Conference on English Language Research on Computerized Corpora (ICAME 23)* Goteborg 22-26 May 2002. K. Aijmer and B. Altenberg (Eds.). 311-326. Rodopi.

[Dyvik 2004] Dyvik, H. 2004. Translations as semantic mirrors: From parallel corpus to WordNet. *Language and Computers*, 1: 311-326.

[Edmonds y Cotton 2001] Edmonds, P. y S. Cotton. 2001. Senseval-2: Overview. *Proceedings of Senseval-2: Second Internacional Workshop on Evaluating Word Sense Disambigution Systems*, Toulouse, France, 1-5.

[Eisner y Krakos 2005] Eisner, J. y Karakos, D. (2005). Bootstrapping without the boot. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 395-402, Vancouver, October.

[Erkan y Radev 2004] Erkan, G. y D.R. Radev. 2004. Lexrank: Graph-based centralita as salience in text summarization. *Journal of Artificial Intelligence Research*, 22: 457-479.

[Escudero et al. 2000a] Escudero, G., L. Màrquez y G. Rigau. 2000ª. Boosting applied to word sense disambiguation. *Proceedings of the 12th European Conference on Machine Learning (ECML)*, Barcelona, Spain, 129-141.

[Escudero et al. 2000b] Escudero, G., L. Màrquez y G. Rigau. 2000b. Naive bayes and ejemplar-based approaches to word sense disambiguation revisited. *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, Berlin, Germany, 421-425.

[Escudero et al. 2000c] Escudero, G., L. Màrquez y G. Rigau. 2000c. On the portability and tuning of supervised Word sense disambiguation systems. *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, Hong Kong, China, 172-180.

[Escudero et al. 2001] Escudero, G., L. Màrquez y G. Rigau. 2001. Using LazyBoosting for Word sense disambiguation. *Proceedings of Senseval-2: Second Internacional Workshop on Evaluating Word Sense Disambiguation Systems*, Toulouse, France.

[Fellbaum 1998] Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Massachussets and London: The MIT Press.

[Fellbaum et al. 2001] Fellbaum, C., M. Palmer, H.T. Dang, L. Delfs y S. Wolf. 2001. Manual and automatic semantic annotation with WordNet. *Proceedings of the Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA.

[Fellbaum et al. 2005] Fellbaum, C., L. Delfs, S. Wolf y M. Palmer. 2005. word meaning in dictionaries, corpora and the speaker's mind. Meaningful Texts: The Extraction of Seemantic Information from Monolingual and Multilingual Corpora. Barnbrook, Danielsson and Mahlberg (Eds.). London: Continuum.

[Fillmore 1971] Fillmore, C. 1971. Types of lexical semantics information. *Semantics: An interdisciplinary reader in philosophy, linguistics and psychology*. Cambridge: Cambridge University Press, 370-392.

[Firth 1957] Firth, J. 1957. A sinopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*, Philological Society, Oxford. Reprinted in *Selected Papers of J. R. Firth 1952-59*. Palmer (Ed.). 168-205. London: Longmans, 1968.

[Francis y Kucera 1964] Francis, W.N. and H. Lucera. 1964/1979. *Manual of information to accompany A Standard Corpus of Present-Day Edited American English*. Brown University, Department of Linguistics.

[Gale et al. 1992a] Gale, W., K. Church y D. Yarowsky. 1992a. Estimating upper and coger bounds on the performance of word-sense disambiguation programs. *Proceedings of the Annual Meeting of the Association for Computacional Linguistics (ACL)* Newark, USA, 249-256.

[Gale et al. 1992b] Gale, W., K. Church y D. Yarowsky. 1992b. One sense per discourse. *Proceedings of the DARPA Speech and Natural Language Workshop*, New York, USA, 233-237.

[Gao et al. 2001] Gao, J., J. Nie, J. Zhang, E. Xun, M. Zhou y C. Huang. 2001. Impro ving query translation for CLIR using statistical models. *Proceedings of the 24th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, Louisiana, 96-104.

[Gonzalo y Verdejo 2006] Gonzalo, J. y F. Verdejo. 2006. Automatic Acquisition of Lexical Information and Examples. *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (Eds.). Dordrecht. Springer.

[Gustman et al. 2002] Gustman, S., D. Soergel, D. Orad, W. Byrne, M. Picheny, B. Ramabhadran y D. Greenberg. 2002. Supporting access to large digital oral history archives. *Proceedings of the Joint Conference on Digital Libraries*, 18-27.

[Guthrie et al. 1991] Guthrie, J.A., L. Guthrie, Y. Wilks y H. Aidinejad. 1991. Subject dependent co-ocurrence and word sense disambiguation. *Proceedings of the 29th Annual Meeting of the Association for Computacional Linguistics*, 146-152.

[Halliday y Hasan 1976] Halliday, M. y R. Hasan. 1976. *Cohesión in English*. London: Longman.

[Hanks 2000] Hanks, P. 2000. Do word meanings exist? Computers and the Humanities, 34(1-2): 205-215.

[Harris 1968] Harris, Z. 1968. *Mathematical structures of language*. New York: Interscience Publishers.

[Hatzivassiloglou et al. 2001] Hatzivassiloglou, V., P.A. Duboué y A. Rzhetsky. 2001. Disambiguating proteins, genes and RNA in text: A machine learning approach. *Proceedings of the Ninth Internacional Conference on Intelligent Systems for Molecular Biology*, Copenhagen, Denmark, 97-106.

[He y Gildea 2004] He, S. y Gildea, D. (2004). Self-training and co-training for semantic role labelling: primary report. *University of Rochester Technical Report 891*. Rochester, New York.

[Hearst y Schütze 1993] Hearst, M. y H. Schütze. 1993. Customizing a lexicon to better suit a computacional task. *Proceedings of the ACL SIGLEX Workshop on the Acquisition of Lexical Knowledge from Text*.

[Hearst y Pedersen 1996] Hearst, M. y J. Pedersen. 1996. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. Proceedings of the 19th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, 76-84.

[Hearst 2000] Hearst, M. 2000. Next generation web search: Setting our sites. *IEEE Data Engineering Bulletin, Special issue on Next Generation Web Search*. Gravano (Ed.).

[Hermjakob y Mooney 1997] Hermjakob, U. y Mooney, R.J.. (1997). Learning parse and translation decisions from examples with rich context. *Proceedings of the Association for Computational Linguistics*, 482-489. Madrid, Spain.

[Hotho et al. 2003] Hotho, A., S. Staab y G. Stumme. 2003. WordNet improves text document clustering. *Proceedings of the Semantic Web Workshop at the 26th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada.

[Hwa 2004] Hwa, R. (2004). Sample selection for statistical parsing. *Computational Linguistics*, 30 (3).

[Hwa, Resnik, Weinberg, Cabezas y Kolak 2005] Hwa, R., Resnik, P., Weinberg, A., Cabezas, C. y Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 11 (3), 311-325.

[Ide 1998] Ide, N. 1998. Cross-lingual sense determination: Can it work? *Computers and the Humanities*, 34(1-2): 223-234.

[Ide y Véronis 1998] Ide, N. y J. Véronis. 1998. Introduction to the special issue on word sense disambiguation. *Computacional Linguistics*, 24(1): 1-40.

[Ide et al. 2001] Ide, N., T. Erjavec y D. Tufis. 2001. automatic sense tagging using parallel corpora. *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium*, Tokyo, Japan, 212-219.

[Ide et al. 2002] Ide, N., T. Erjavec y D. Tufis. 2002. Sense discrimination with parallel corpora. *Proceedings of the ACL-2002 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia, 54-60.

[Ide y Wilks 2006] Ide, N. y Y. Wilks. 2006. Making sense about sense. *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (Eds.). Dordrecht. Springer.

[Jiang y Conrath 1997] Jiang, J. y D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of the Internacional Conference on Research in Computacional Linguistics*, Taipei, Taiwán.

[Jiménez et al. 2005] Jiménez, J., L. Màrquez y G. Rigau. 2005. Automatic translation of WordNet glosses. *Proceedings of the EUROLAN'05 Cross-Language Knowledge Induction Workshop*, Cluj-Napoca, Romania, July.

[Jing y Tzoukermann 1999] Jing, H. y E. Tzoukermann. 1999. Information retrieval based on context distance and morphology. *Proceedings of the 22nd Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 90-96.

[Kehagias et al. 2003] Kehagias, A., V. Petridis, V. Kaburlasos y P. Fragkou. 2003. A comparison of word- and sense-based text categorization using several classification algorithms. *Journal of Intelligent Information Systems*, 21(3): 227-247.

[Kilgarriff 1993] Kilgarrif, A. 1993. Dictionary word sense distinctions: An enquiry into their nature. *Computers and the Humanities*, 26: 356-387.

[Kilgarriff 1997] Kilgarrif, A. 1997. "I don't relieve in word senses". *Computers an the Humanities*, 31(2): 91-113.

[Kilgarrif 1998] Kilgarrif, A. 1998. Senseval: An exercise in evaluating word sense disambiguation programs. *Proceedings of the European Conference on Lexicography (EURALEX)*, 176-184, Liège, Belgium. Also in *Proceedings of the 1st Conference on Language Resources and Evaluation (LREC)*, Granada, Spain, 581-588.

[Kilgarrif 2001] Kilgarrif, A. 2001. English lexical simple task description. Proceedings of Senseval-2: Second Internacional Workshop on Evaluating Word Sense Disambigution Systems, Toulouse, France, 17-20.

[Kilgarrif y Palmer 2000] Kilgarrif, A. y M. Palmer. 2000. Introduction to the special issue on Senseval. *Computers and the Humanities*, 34(1-2): 1-13.

[Kilgarrif y Rosenzweig 2000] Kilgarrif, A. y J. Rosenzweig. 2000. Framework and results for English Senseval. *Computers and the Humanities*, 34(1-2): 15-48.

[Kilgarriff y Tugwell 2001] Kilgarrif, A. y D. Tugwell. 2001. WASP-Bench: An MT lexicographers' workstation supporting state-of-the-art lexical disambiguation. *Proceedings of MT Summit VIII*, Santiago de Compostela, Spain, 187-190.

[Kim et al. 2004] Kim, S., H. Seo y H. Rim. 2004. Information retrieval using word senses: Root sense tagging approach. *Proceedings of the 27th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, UK, 258-265.

[Klein y Murphy 2001] Klein, D. y G. Murphy. 2001. The representation of polysemous words. *Journal of Memory and Language*, 45: 259-82.

[Klein y Murphy 2002] Klein, D. y G. Murphy. 2002. Paper has been my ruin: Conceptual relations of polysemous senses. *Journal of Memory and Language*, 47: 548-570.

[Koehn et al. 2003] Koehn, P., F.J. Och y D. Marcu. 2003. Statistical phrase-based translation. *Proceedings of the Human Language Technology Conference (HLT-NAACL)*, Edmonton, Canada, 48-54.

[Koehn 2004] Koehn, P. 2004. Pharaoh: A Beam Search decoder for phrase-based statistical machine translation models. *Proceedings of the 6th Conference of the Association for Machine Translation in the Ameritas (AMTA)*, Georgetown.

[Kriedler 1998] Kriedler, C. 1998. Introducing English Semantics. London and New Cork: Routledge.

[Krishnamurthy y Nicholls 2000] Krishnamurthy, R. y D. Nicholls. 2000. Peeling an onion: The lexicographer's experience of manual sense-tagging. Computers and the Humanities, 34(1-2): 85-97

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

[Krovetz 1998] Krovetz, R. 1998. More than one sense per discourse. *Proceedings of the Workshop on Evaluating Word Sense Disambiguation Systems* (Senseval-1), Sussex, England.

[Krovetz 2002] Krovetz, R. 2002. On the importante of word sense disambiguation for information retrieval. *Proceedings of the LREC 2002 Workshop on Creating and Using Semantics for Information Retrieval and Filtering, Third Internacional Conference on Language Resources and Evaluation*, Las Palmas, Canary Islands, Spain, June.

[Kumar y Byrne 2002] Kumar, S. y W. Byrne. 2002. Minimum Bayes-risk word alignement of bilingual texts. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, July, Philadelphia, PA.

[Lakoff y Johnson 1980] Lakoff, G. y M. Jonson. 1980. *Metaphors We Live By*. University of Chicago Press.

[Lakoff 1987] Lakoff, G. 1987. *Women, Fire and Dangerous Things*. Chicago: University of Chicago Press.

[Landauer et al. 1998] Landauer, T.K., P.W. Foltz y D. Laham. 1998. An introduction to Latent Semantic Analysis. *Discourse Processes*, 25: 259-284.

[Leacock et al. 1993] Leacock, C., G. Towell y E. Voorhees. 1993. Towards building contextual representations of Word senses using statistical models. *Proceedings of the ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*, 10-20.

[Leacock et al. 1998] Leacock, C., M. Chodorow y G.A. Miller. 1998. Using corpus statistics and WordNet relations for sense identication. *Computacional Linguistics*, 24(1): 147-165.

[Lee y Ng 2002] Lee, Y.K. y H.T. Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. *Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, USA, 41-48.

[Lee et al. 2004] Lee, Y.K., H.T. Ng y T.K. Chia. 2004. Supervised word sense disambiguation with support vector machines and multiple knowledge sources. *Proceedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 137-140.

[Lesk 1986] Lesk, M. 1986. automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. *Proceedings of the ACM-SIGDOC Conference*, Toronto, Canada, 24-26.

[Levin 1993] Levin, B. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago: Univrsity of Chicago Press.

[Li y Li 2004] Li, H. y C. Li. 2004 Word translation disambiguation using bilingual bootstrapping. *Computational Linguistics*, 30(1): 1-22.

[Li y Zheng 2005] Li, Y. y Z. Zheng. 2005. KDD-Cup 2005 presentation, *Eleventh ACM SIGKDD Internacional Conference on Knowledge Discovery and Data Mining (KDD-05)*. (http://kdd05.lac.uic.edu/kddcup.html).

[Lin 1993] Lin, D. 1993. Principle based parking without overgeneration. *Proceedings of the 31st Annual Meeting of the Association for Computacional Linguistics (ACL)*, Columbus, USA, 112-120.

[Lin 1997] Lin, D. 1997. Using syntactic dependency as local context to resolve Word sense ambiguity. *Proceedings of the 35th Annual Meeting of the Association for Computacional Linguistics (ACL)*, Madrid, 64-71.

[Lin 1998] Lin, D. 1998. Automatic retrieval and clustering of similar words. *Proceedings of the 17th Internacional Conference on Computacional Linguistics (COLING-ACL-98)*. Montreal, Canada, 768-774.

[Littman et al. 1998] Littman, M., S. Dumais y T. Landauer. 1998. Automatic cross-language informatio retrieval usin latent semantic indexing. *Cross Language Information Retrieval*. G. Grefenstette (Ed.). 51-62. Kluwer Academia Publishers.

[Liu et al. 2004] Liu, S., F. Liu, C. Yu y W. Meng. 2004. An effective approach to document retrieval via utilizing WordNet and recognizing phrases. *Proceedings of the 27th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, UK, 266-272.

[Löfberg, Archer, Piao, Rayson, McEnery, Varantola y Juntunen 2003] Löfberg, L., Archer, D., Piao, S., Rayson, P., McEnery, T., Varantola, K. y Juntunen, J. (2003). Porting an English semantic tagger to the Finnish language. *Proceedings of the Corpus Linguistics 2003 Conference*. Archer, D., Rayson, P., Wilson, A. and McEnery, T. (Eds.). UCREL technical paper number 16. UCREL, Lancaster University, 457 – 464.

[Löfberg et al. 2004] Löfberg, L., J. Juntunen, A. Nykänen, K. Varantola, P. Rayson y D. Archer. 2004. Using a semantic tagger as a dictionary search tool. *Proceedings of the 11th EURALEX (European Association for Lexicography) International Congress*, Lorient, France, July, 127-134.

[Ma y Liberman 1999] Ma, X. y M. Liberman. 1999. BITS: A method for bilingual text search over the Web. *Proceedings of the Machine Translation Summit VII*, Singapore.

[Magnini y Cavaglià 2000] Magnini, B. y G. Cabaglià. 2000. Integrating subject field codes into WordNet. *Proceedings of the Second Internacional Conference on Language Resources and Evaluation Conference (LREC)*, Athens, Greece, 1413-1418.

[Magnini et al. 2002] Magnini, B., C. Strapparava, G. Pezzulo y A. Gliozzo. 2002. The role of domain information in word sense disambiguation. *Natural Language Engineering*, 8(4): 359-373.

[Màrquez et al. 2006] Màrquez, L., G. Escudero, D. Martinez, G. Rigau. Supervised corpus-based methods for WSD. *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (Eds.). Dordrecht. Springer.

[Martínez y Agirre 2000] Martinez, D. y E. Agirre. 2000. One sense per collocation and genre/Tepic variations. *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, 207-215.

[McCarthy et al. 2001] McCarthy, D., J. Carroll y J. Preiss. 2001. Diambiguating noun and verb senses using automatically acquired selectional preferences. *Proceedings of the ACL/EACL Senseval-2 Workshop*, Toulouse, France.

[McCarthy et al. 2004] McCarthy, D., R. Koeling, J. Weeds y J. Carroll. 2004. Using automatically acquired predominant senses for word sense disambiguation. *Proceedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 151-158.

[Melamed y Resnik 2000] Melamed, I.D. y P. Resnik. 2000. Tagger evaluation given hierarchical tag sets. *Computers and the Humanities*, 34(1-2): 79-84.

[Merlo, Stevenson, Tsang y Allaria 2002] Merlo, P., Stevenson, S., Tsang, V. & Allaria, G. (2002). A multilingual paradigm for automatic verb classification. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, 207-214. Philadelphia, PA.

[Mihalcea y Moldovan 1999] Mihalcea, R. y D. Moldovan. 1999. A method for word sense disambiguation of unrestricted text. *Proceedings of the 37th Annual Meeting of the Association for Computacional Linguistics (ACL)*, Maryland, NY, 152-158.

[Mihalcea y Moldovan 2000] Mihalcea, R. y D. Moldovan. 2000. Semantic indexing using WordNet senses. *Proceedings of the ACL Workshop on Recent Advances in Natural Language Processing and Information Retrieval*. Hong Kong.

[Mihalcea y Moldovan 2001] Mihalcea, R. y D. Moldovan. 2001. Automatic generation of a coarse grained wordnet. *Proceedings of NAACL-2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA, 35-41.

[Mihalcea 2002a] Mihalcea, R. 2002a. Word sense disambiguation with pattern learning and automatic feature selection. *Natural Language Engineering*, 8(4): 348-358.

[Mihalcea 2002b] Mihalcea, R. 2002b. Bootstrapping large sense tagged corpora. *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Las Palmas, Spain.

[Mihalcea y Chklovski 2003] Mihalcea, R. y T. Chklovski. 2003. Open Mind Word Expert: Creating large annotated data collections with Web users' help. *Proceedings of the EACLWorkshop on Linguistically Annotated Corpora*, Budapest, Hungary.

[Mihalcea 2004] Mihalcea, R. 2004. Co-training and self-training for word sense disambiguation. *Proceedings of the Conference on Natural Language Learning (CoNLL)*. Boston, USA, 33-40.

[Mihalcea y Edmonds 2004] Mihalcea, R. y P. Edmonds, eds. 2004. *Procedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain.

[Mihalcea et al. 2004] Mihalcea, R., T. Chlovski y A. Kilgarrif. 2004. The Senseval-3 English lexical sample task. *Procedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 25-28.

[Mihalcea, Tarau y Figa 2004] Mihalcea, R., P. Tarau y E. Figa. 2004. PageRank on semantic networks with application to word sense disambiguation. *Proceedings of the 20th Internacional Conference on Computational Linguistics (COLING)*, Switzerland, Geneva.

[Mihalcea 2006] Mihalcea, R. 2006. Knowledge-based methods for WSD. *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (Eds.). Dordrecht. Springer.

[Miller 1990] Millar, G.A. (Ed.). 1990. Special Issue, WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4).

[Miller y Charles 1991] Millar, G. y W. Charles. 1991. Contextual correlatos of semantic similarity. *Language and cognitive Processes*, 6(1): 1-28.

[Miller et al. 1993] Millar, G. A., C. Leacock, R. Tengi y R. T. Bunker. 1993. A semantic concordance. *Proceedings of the ARPA Workshop on Human Language Technology*, 303-308.

[Moschitti y Basili 2004] Moschitti, A. y R. Basili. 2004. Complex linguistic features for text classification: a comprehensive study. *Advances in Information Retrieval: 26th European Conference on IR Research (ECIR 2004)*, Sunderland, UK, April, 181-196.

[Mosteller y Wallace 1964] Mosteller, F., D. Wallace. 1964. *Inference and Disputed Authorship: The Federalist*, Addison-Wesley, Reading, Massachusetts

[Mooney 1996] Money, R.J. 1996. Comparative experiments on disambiguating Word senses: an illustration of the role of bias in machine learning. *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, USA, 82-91.

[Murata et al. 2001] Murata, M., M. Utiyama, K.Uchimoto, Q. Ma y H. Isahara. 2001. Japanese Word sense disambiguation using the simple Bayes and support vector machine methods. *Proceedings of Senseval-2: Second Internacional Workshop on Evaluating Word Sense Disambiguation Systems*. Toulouse, France, 135-138.

[Nagao 1984] Nagao, M. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. *Proceedings of the Internacional NATO Symposium on Artificial and Human Intelligence*, 173-180.

[Ng y Lee 1996] Ng, H.T. y H.B. Lee. 1996. Integrating multiple knowledge sources for word sense disambiguation: An exemplar-based approach. *Proceedings of the 34th Meeting of the Association for Computational Linguistics (ACL)*, Santa Cruz, CA, USA, 40-47.

[Ng 1997a] Ng, H.T. 1997a. Ejemplar-based word sense disambiguation: some recent improvements. *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Providence, USA, 208-213.

[Ng et al. 1999] Ng, H.T., C.Y. Lim y S.K. Foo. 1999. A case study on inter-annotator agreement for word sense disambiguation. *Proceedings of the ACL SIGLEX Workshop on Standardizing Lexical Resources*, Collage Park, Maryland, USA, 9-13.

[Ng et al. 2003] Ng, H.T., B. Wang y Y.S. Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, 455-462.

[Ng y Cardie 2002] Ng, V. y Cardie, C. (2002). Improving machine learning approaches to coreference resolution. *Fortieth Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*.

[Ng y Cardie 2004] Ng, V. y Cardie, C. (2004). Weakly supervised natural language learning without redundant views. *Proceedings of HLT-NAACL*, 173-180. Edmonton, Canada.

[Nigam y Ghani 2000] Nigam, K. y Ghani, R. (2000). Analyzing the applicability and effectiveness of co-training. Proceedings of CIKM-00, 9th ACM International Conference on Information and Knowledge Management, 86-93. McLean, USA. ACM Press. New York.

[Oard y Dorr 1996] Oard, D.W. y B.J. Dorr. 1996. *A Survey of Multilingual Text Retrieval*. Computer Science Report CS-TR-3615, University of Maryland, USA.

[Och 2002] Och, F.J. 2002. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. Ph. D. Thesis, RWTH Aachen, Germany.

[Oltramari et al. 2004] Oltramari, A., P. Paggio, A. Gangemi, M.T. Pazienza, N. Calzolari, B. Sandford Pedersen y K. Simov (Eds.). 2004. *OntoLex 2004: Ontologies and Lexical Resources in Distributed Environments*, Workshop in association with the 4th Internacional Conference on Language Resources and Evaluation (LREC), Lisbon, May.

[Osborne y Baldridge 2004] Osborne, M. y Baldridge, J. (2004). Ensemble-based active learning for parse selection. *Proceedings of HLT-NAACL*, Boston.

[Palmer, Fellbaum y Dang 2006] Palmer, M., C. Fellbaum y H.T. Dang. 2006. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering*, 12(3).

[Palmer, Ng y Dang 2006] Palmer, M., H.T. Ng y H.T. Dang. 2006. Evaluation of WSD systems. *Word Sense Disambiguation. Algorithms and Applications.* Agirre and Edmonds (Eds.). Dordrecht. Springer.

[Palmer et al. 2001] Palmer, M., C. Fellbaum, S. Cotton, L. Delfs y H.T. Dang. 2001. English tasks: All-words and verb lexical simple. *Proceedings of Senseval-2: Second Internacional Workshop on Evaluating Word Sense Disambigution Systems*, Toulouse, France, 21-24.

[Pasca y Harabagiu 2001a] Pasca, M. y S. Harabagiu. 2001a. High performance question/ansewring. *Proceedings of the 24th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retieval*, New Orleans, Louisisna, 366-374.

[Pasca y Harabagiu 2001b] Pasca, M. y S. Harabagiu. 2001b. The informative role of WordNet in open-domain question answering. *Proceedings of the NAACL-2001 Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA, 138-143.

[Patwardhan et al. 2003] Patwardhan, S., S. Banerjee y T. Pedersen. 2003. Using measures of semantic relatedness for Word sense disambiguation. *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computacional Linguistics (CICLing)*, Mexico City, Mexico.

[Pedersen 2006] Pedersen, T. 2006. Unsupervised corpus-based methods for WSD. *Word Sense Disambiguation. Algorithms and Applications*. Aguirre and Edmonds (Eds.). Dordrecht. Springer.

[Pedersen y Bruce 1997] Pedersen T. y R. Bruce. 1997. A new supervised learning algorithm for Word sense disambiguation. *Proceedings of the 14th Nacional Conference on Artificial Intelligence (AAAI)*, Providence, USA, 604-609.

[Peh y Ng 1997] Peh, L. y H.T. Ng. 1997. Domain-specific semantic class disambiguation using WordNet. Proceedings of the Fifth Workshop on Very Large Corpora. Beijing & Hong Kong, 56-64.

[Pham et al. 2005] Pham, T.P., H.T. Ng y W.S. Lee. 2005. Word sense disambiguation with semi-supervised learning. *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, USA, 1093-1098.

[Pierce y Cardie 2001] Pierce , D. y Cardie, C. (2001). Limitations of co-training for natural language learning from large datasets. *Proceedings of EMNLP*, 1-9.

[Popescu 2004] Popescu, M. 2004. Regularizad least-squares classification for word sense disambiguation. *Proceedings of Senseval-3: Third Internacional Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, 209-212.

[Pradhan, Loper, Dligach y Palmer 2007] Pradhan, S., Loper, E., Dligach, D. y Palmer, M, (2007). SemEval-2007 Task 17: English Lexical Sample, Semantic Role Labelling and All Words. *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics*, 87-92. Prague.

[Procter 1978] Procter, P. ed. 1978. *Longman Dictionary of Contemporary English*. London: Longman Group.

[Pustejovsky 1995] Pustejovsky, J. 1995. *The Generative Lexicon*. MIT Press.

[Qu et al. 2002] Qu, Y., G. grefenstette y D.A. Evans. 2002. Resolving translation ambiguity using monolingual corpora. *Advances in Cross-Language Information Retrieval: Third Workshop of the Cross-Language Evaluation Forum (CLEF 2002)*, Rome, Italy, 223-241.

[Rayson, Archer, Piao y McEnery 2004] Rayson, P., Archer, D., Piao, S. L. & McEnery, T. (2004). The UCREL semantic analysis system. *Proceedings of the workshop on Beyond Named Entity Recognition Semantic Labelling for NLP Tasks, 4th International Conference on Language Resources and Evaluation (LREC 2004)*, 7-12. Lisbon, Portugal.

[Resnik 1997] Resnik, P. 1997. Selectional preferentes and Word sense disambiguation. *Proceedings of the ACL/SIGLEX Workshop on Tagging Text with Lexical Semantics: What, Why and How?*, Washington, DC, USA, 52-57.

[Resnik 1999a] Resnik, P. 1999a. Disambiguating noun groupings with respect to WordNet senses. *Natural Language Processing Using Very Large Corpora*. S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann and D. Yarowsky (Eds.). 77-98. Dordrecht: Kluwer Academia Publishers.

[Resnik 1999b] Resnik, P. 1999b. Mining the Web for bilingual text. Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL), Maryland, USA, 527-534.

[Resnik y Yarowsky 1997a] Resnik, P. y D. Yarowsky. 1997a. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2): 113-133.

[Resnik y Yarowsky 1997b] Resnik, P. y D. Yarowsky. 1997b. A perspective on word sense disambiguation methods and their evaluation. *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What and How?*, Washington, USA, 79-86.

[Resnik y Yarowsky 2000] Resnik, P. y D. Yarowsky. 2000. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Natural Language Engineering*, 5(2): 113-133.

[Resnik y Smith 2003] Resnik, P. y N. Smith. 2003. The Web as a parallel corpus. Computational Linguistics, 29(3): 349-380.

[Resnik 2004] Resnik, P. 2004. *Exploiting hidden meanings: A Class-Based Approach to Lexical Relationships*. Ph.D. Thesis, Department of Computer and Information Science, University of Pennsylvania, USA.

[Resnik 2006] Resnik, P. 2006. WSD in NLP applications. *Word Sense Disambiguation. Algorithms and Applications*. Agirre and Edmonds (Eds.). Dordrecht. Springer.

[Rigau et al. 2002] Rigau, G., B. Magnini, E. Agirre, P. Vossen y J. Carroll. 2002. MEANING: A roadmap to knowledge Technologies. *Proceedings of the COLING Workshop on a Roadmap for Computacional Linguistics*, Taipei, Taiwan.

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

[Rodd et al. 2002] Rodd, J., M. Gareth-Gaskell y W. Marslen-Wilson. 2002. Making sense of semantic ambiguity: Semantic competition in lexical access. *Journal of Memory and Language*, 46: 245-266.

[Rodd et al. 2004] Rodd, J., M. Gareth-Gaskell y W. Marslen-Wilson. 2004. Modelling the effects of semantic ambiguity in word recognition. *Cognitive Science*, 28: 89-104.

[Ruhl 1989] Ruhl, C. 1989. *On Monosemy: A Study in Linguistic Semantics*. Albany: State University of New York Press.

[Sag et al. 2002] Sag, Ivan A., T. Baldwin, F. Bond, A. Copestake y D. Flickinger. 2002. Multiword expressions: A pain inthe neck for NLP. *Proceedings of the 3rd Internacional Conference on Intelligent Text Processing and Computacional Linguistics (CICLing)*, Mexico City, 1-15.

[Salton y Buckley 1988] Salton, G. y C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5): 513-523.

[Sánchez de Madariaga y Fernández del Castillo 2008] Sánchez de Madariaga, R. y Fernández del Castillo, J. R. 2008. The bootstrapping of the Yarowsky algorithm in real corpora. *Information Processing & Management*. Aceptado con modificaciones, 12 de marzo de 2008.

[Sánchez de Madariaga, Paice, Rayson y Fernández del Castillo 2008] Sánchez de Madariaga, R., Paice, C. D., Rayson, P. y Fernández del Castillo, J. R. 2008. Domain differences in the nature of text corpora: an investigation through ambiguity. Conferencia invitada, 26 de mayo de 2008, *Corpus Resaerch Group (CRG)*, Lancaster University.

[Sanderson 1994] Sanderson, M. 1994. Word sense disambiguation and information retrieval. Proceedings of the 17th Annual Internacional ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 142-151.

[Sanderson 2000] Sanderson, M. 2000. Retrieving with good sense. *Information Retrieval*, 2(1): 49-69.

[Santamaría et al. 2003] Santamaría, C., J. Gonzalo y F. Verdejo. 2003. Automatic association of Web directories to word senses. *Computational Linguistics*, 29(3): 485-502.

[Sarkar 2001] Sarkar, A. (2001). Applying co-training methods to statistical parsing. *Proceedings of the Second Meeting of the North American Association for Computational Linguistics*, 175-182. Pittsburgh, PA.

[Sarkar 2008] Sarkar, A. (2008). Semi-supervised learning for statistical machine translation. In C. Goutte, N. Cancedda, M. Dymetman & G. Foster (Eds.), *Learning Machine Translation*. MIT Press. (En prensa).

[Schapire 2003] Schapire, R.E. 2003. The boosting approach to machine learning: An overview. *Nonlinear Estimation and Classification*. Denison, Hansen, Colmes, Mallick, and Yu (Eds.). New York, USA. Springer.

[Schütze y Pedersen 1995] Schütze, H. y J. Pedersen. 1995. Information retrieval based on word senses. *Fourth Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, 161-175.

[Schütze 1998] Schütze, H. 1998. Automatic word sense discrimination. *Computacional Linguistics*, 24(1): 97-123.

[Segond 2000] Segond, F. 2000. Framework and results for French. *Computers and the Humanities*, 34(1-2): 49-60.

[Smith 2006] Smith, N. (2006). *Language and Statistics II*. Carnegie Mellon University. http://www.cs.cmu.edu/~nasmith/LS2.F06/lecture21.pdf

[Sparck Jones 1964] Sparck Jones, K. 1964/1986. *Synonymy and Semantic Classification*. Edinburgh: Edinburgh University Press.

[Sparck Jones 1999] Sparck Jones, K. 1999. What is the role of NLP in text retrieval? *Natural Language Information Retrieval*. Strzalkowski (Ed.). New York: Kluwer Academic Publishers.

[Sproat, Hirschberg y Yarowsky 1992] Sproat, R., J. Hirschberg y D. Yarowsky. 1992. A corpus-based synthesizer. *Proceedings of the International Conference on Spoken Language Processing*, Banff, Alberta.

[Steedman, Osborne, Sarkar, Clark, Hwa, Hockenmaier, Ruhlen, Baker y Crim 2003] Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., Ruhlen, P., Baker, S. & Crim, J. (2003). Bootstrapping statistical parsers from small datasets. *Proceedings of the Tenth Conference of the European Chapter of the Association for Computational Linguistics*, 331-338. Budapest, Hungary.

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

[Stoica y Hearst 2004] Stoica, E. y M. Hearst. 2004. Nearly-automated metadata hierarchy creation. *Proceedings of HLT-NAACL 2004: Short Papers*, Boston, MA, USA, 117-120.

[Strapparava 2004] Strapparava, C., A. Gliozzo y C. Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: IRST at Senseval-3. *Proceedings of Senseval-3: Third Internacional Workshop on the Evaluation of Sistemas for the Semantic Analysis of Text*. Barcelona, Spain, 229-234.

[Suárez y Palomar 2002] Suárez, A. y M. Palomar. 2002. A maximum entropy-based Word sense disambiguation system. *Proceedings of the 19th Internacional Conference on Computacional Linguistics (COLING)*, Taipei, Taiwán, 960-966.

[Tang, Luo y Roukos 2002] Tang, M., Luo, X. & Roukos, S. (2002). Active learning for statistical natural language parsing. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 120-127. Philadelphia, PA.

[Traupman y Wilensky 2003] Traupman, J. y Wilensky, R. (2003). *Experiments in Improving Unsupervised Word Sense Disambiguation*. Technical Report. University of California at Berkeley. http://www.eecs.berkeley.edu/Pubs/TechRpts/2003/CSD-03-1227.pdf

[Tufis et al. 2004a] Tufis, D., D. Cristea y S. Stamou. 2004a. Balkanet: Aims, methods, results and perspectives. A general overview. Romanian Journal of Iformation Science and Technology, 7(1-2): 9-43.

[Tufis et al. 2004b] Tufis, D., R. Ion y N. Ide. 2004b. Fine-grained Word sense disambiguation based on parallel corpora, word alignment, word clustering, and aligned wordnets. *Proceedings of the Twentieth International Conference on Computacional Linguistics (COLING-2004)*. August 2004, Geneva.

[Turcato et al. 2002] Turcato, D., F. Popowich, J. Toole, D. Fass, D. Nicholson y G. Tisher. 2002. Adapting a synonym database to specific domains. *Proceedings of the*

ACL Workshop on Recent Advances in Natural Language Processing and Information Retrieval, Hong Kong.

[Vapnik 1998] Vapnik, V. 1998. A study of polysemy judgements and inter-annotator agreement. Programme and Advanced Papers of Senseval-1: The First Internacional Workshop on the Evaluation of Systems for the Semantic Análisis of Text, Herstmonceux, England, 2-4.

[Véronis 2004] Véronis, J. 2004. HyperLex: Lexical cartography for information retrieval. *Computer, Speech & Language*, 18(3): 223-252.

[Vickrey et al. 2005] Vickrey, D., L. Biewald, M. Teyssier y D. Koller. 2005. Word sense disambiguation for machine translation. *Proceedings of Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*. Vancouver, Canada, October.

[Vlachos 2006] Vlachos, A. (2006). Active annotation. *Proceedings of the EACL Workshop on Adaptive Text Extraction*.

[Voorhees 1999] Voorhees, E.M. 1999. Natural language processing and information retrieval. *Information Extraction: Towards Scalable, Adaptable Systems*. Pazienza (Ed.). 32-48. London: Springer-Verlag. (Lectura Notes in Computer Science 1714).

[Voorhees y Tice 2000a] Voorhees, E.M. y D.M. Tice. 2000. Building a question answering test collection. *Proceedings of the 23rd Annual Internacional ACM SIGIR Conference on Research and Developmnet in Information Retrieval*, Athens, Greece, 200-207.

[Vossen 1998] Vossen, P. (Ed.). 1998. *EuroWordNet. A multilingual database with lexical semantic networks*. Dordrecht: Kluwer Academia Publishers.

[Vossen 2001] Vossen, P. 2001.Extending, trimming and fusing WordNet for technical documents. *Proceedings of the Workshop on WordNet and Other Lexical Resources*, Pittsburgh, PA.

[Vossen et al. 2006] Vossen, P., G. Rigau, I. Alegria, E. agirre, D. Farwell y M. Fuentes. 2006. Meaningful resultas for information retrieval in the MEANING Project. Proceedings of the Third Global WordNet Conference, Jeju Island, Korea.

[Walker y Amsler 1986] Walter, D. y R. Amsler. 1986. The use of machine readable dictionaries in sublanguage analysis. *Analyzing Language in Restricted Domains*. Grishman and Kittredge (Eds.), 69-83. Hillsdale, NJ: Erlbaum.

[Weeber et al. 2001] Weeber, M., J. Mork y A. Aronson. 2001. Developing a test collection for biomedical word sense disambiguation. *Proceedingas of the AMIA 2001 Symposium*.

[Weischedel y Palmer 2004] Weischedel, R. y M. Palmer. 2004. An ontoBank pilot study: Annotating word sense and co-reference. DARPA TIDES PI Meeting, Philadelphia, PA, July 13-15.

[White y O'Conell 1994] White, J. y T. O'Connell. 1994. The ARPA MT evaluation methodologies: Evolution, lessons and future approaches. Technology Partnerships for Crossing the Language Barrier: *Proceedings of the First Conference of the Association for Machine Translation in the Ameritas (AMTA)*, Columbia, MD, USA, 193-205.

[Wierzbicka 1989] Wierzbicka, A. 1989. Semantic primitives and lexical universals. *Quaderni di Semantica*, X(1): 103-121.

[Wilcock et al. 2004] Wilcock, G., P. Buitelaar, A. Pareja-Lora, B. Bryant, J. Lin y N. Ide. 2004. The roles of natural language and XML in the Semantic Web. *Computacional Linguistics and Beyond: Perspectives at the Beginning of the 21st Century*. Huang and Lenders (Eds.). Frontiers in Linguistics Series, Academia Sinica, Taiwán.

[Wilks 1975] Wilks, Y. 1975. A preferential pattern-seeking semantics for natural language inference. *Artificial Intelligence*, 6: 53-74.

[Wilks y Fass 1992] Wilks, Y. y D. Fass. 1992. The preference semantics family. *Computers & Mathematics with Applications*, 23(2-5): 205-221.

[Wilks 1997] Wilks, Y. 1997. Senses and texts. *Computers and the Humanities* 31(2): 77-90.

[Wilks 1998] Wilks, Y. 1998. Is word sense disambiguation just one more NLP task? *Proceedings of the SENSEVAL Conference*, Herstmonceaux, Sussex. Also appears as Technical Report CS-98-12, Department of Computer Science, University of Sheffield.

[Wilks y Stevenson 1998] Wilks, Y. y M. Stevenson. 1998. The grammar of sense: using part of speech tags as a first step in semantic disambiguation. *Journal of Natural Language Engineering*, 4(2): 135-144.

[Wilks y Catizone 2002] Wilks, Y. y R. Catizone. 2002. What is lexical tuning? *Journal of Semantics*, 19(2): 167-190.

[Witten y Frank 2005] Witten, I.H. y E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Second Edition. Morgan Kaufmann.

[Woods 1995] Woods, W.A. 1995. Finding information on the Web: A knowledge representation approach. Presented at the *Fourth Internacional World Wide Web Conference*, Boston, MA.

[Wu et al. 2004] Wu, D., W. Su andM. Carpuat. 2004. A kernel PCA method for superior word sense disambiguation. *Proceedings of the 42nd Annual Meeting of the Association for Computacional Linguistics (ACL)*, Barcelona, Spain, 637-644.

[Yarowsky 1992] Yarowsky, D. 1992. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. *Proceedings of he 14th Intenational Conference on Computacional Linguistic (COLING)*, Nantes, France, 454-460.

[Yarowsky 1993] Yarowsky, D. 1993. One sense per collocation. *Proceedings of the ARPA Human Language Technology Workshop*, Plainsboro, USA, 265-271.

[Yarowsky 1994] Yarowsky, D. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Las Cruces, USA, 88-95.

[Yarowsky 1995] Yarowsky, D. 1995. Unsupervised Word sense disambiguation rivaling supervised methods. *Proceedings of the 33rd Annual Meeting of the Association for Computacional Linguistics (ACL)*, Cambridge, USA, 189-196.

[Yarowsky, Cucerzan, Florian, Schafer y Wicentowski 2001] Yarowsky, D., S. Cucerzan, R. Florian, C. Schafer y R. Wicentowski. 2001. The Johns Hopkins Senseval-2 system descriptions. *Proceedings of Senseval-2: Second Internacional Workshop on Evaluating Word Sense Disambiguation Systems*, Toulouse, France.

[Yarowsky y Ngai 2001] Yarowsky, D. y Ngai, G. (2001). Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. *Proceedings of the Second Meeting of the North American Association for Computational Linguistics*, 200-207.

[Yarowsky, Ngai y Wicentowski 2001] Yarowsky, D., Ngai, G. & Wicentowski, R. (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. *Proceedings of HLT 2001, First International Conference on Human Language Technology Research*.

[Yarowsky y Florian 2002] Yarowsky, D. y R. Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Journal of Natural Language Engineering*, 8(2): 293-310.

[Yee et al. 2003] Yee, P., K. Swearingen, K. Li y M. Hearst. 2003. Faceted metadata for image search and browsing. *Proceedings of the Conference on Human Factors in Computing Systems (ACM CHI)*, Ft. Lauderdale, Florida, USA, 401-408.

[Yeh et al. 2002] Yeh, A., L. Hirschman y A. Morgan. 2002. Background and overview for KDD Cup 2002 Task 1: Information extraction from biomedical articles. *SIGKDD Explorations: Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 4(2): 87-89.

[Zipf 1949] Zipf, G. K. 1949. *Human Behaviour and the Principle of Least Effort: An introduction to human ecology*. Cambridge, MA: Addison-Wesley. Reprinted by New York: Hafner, 1972.