Esta tesis doctoral contiene un índice que enlaza a cada uno de los capítulos de la misma.
Existen asimismo botones de retorno al índice al principio y final de cada uno de los capítulos.

Ir directamente al índice

Para una correcta visualización del texto es necesaria la versión de Adobe Acrobat Reader 7.0 o posteriores

Aquesta tesi doctoral conté un índex que enllaça a cadascun dels capítols.
Existeixen així mateix botons de retorn a l'index al principi i final de cadascun deis capítols .

Anar directament a l'index

Per a una correcta visualització del text és necessària la versió d' Adobe Acrobat Reader 7.0 o posteriors.

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante

Resolución de la ambigüedad semántica de las palabras mediante modelos de probabilidad de máxima entropía

Armando Suárez Cueto

Memoria para optar al grado de Doctor en Informática bajo la dirección de

Dr. Manuel Palomar Sanz
Dr. German Rigau Claramunt

Alicante, 28 de junio de 2004

Cofinanciado por el Gobierno de España (CICyT) con los proyectos número TIC2000-0664-C02-02 y número TIC2003-07158-C04-01, y el Gobierno de la Comunidad Valenciana (OCyT) con el proyecto número CTIDIB-2002-151.

Resumen

La resolución de la ambigüedad semántica de las palabras, como tarea en pleno desarrollo y auge dentro del procesamiento del lenguaje natural, trata de la asignación mecanizada de etiquetas de significado a las palabras de un texto. Estas etiquetas son enlaces a definiciones de un cierto diccionario, por lo general electrónico.

Siendo una tarea difícil y aún por resolver, existen muchas propuestas de solución al problema que se pueden resumir en aproximaciones basadas en el conocimiento y aproximaciones basadas en corpus. Las primeras utilizan conocimiento lingüístico preexistente, mientras que las segundas trabajan con grandes cantidades de ejemplos, un corpus, y aprendizaje automático y estadística. Estos ejemplos pueden estar etiquetados manualmente o no, lo que se conoce como aprendizaje supervisado y no supervisado, respectivamente.

Este trabajo de Tesis Doctoral presenta la aplicación de un método de aprendizaje supervisado al problema de la resolución de la ambigüedad semántica de las palabras: los modelos de probabilidad condicional de máxima entropía (Lau et al., 1993; Berger et al., 1996; Ratnaparkhi, 1998). Un sistema de etiquetado automático de palabras con sentidos basado en este método es competitivo, si se compara con otros sistemas actuales.

La principal preocupación a la hora de afinar un sistema de estas características es la elección de la información que el contexto puede proporcionar para ayudar en la tarea. Este trabajo contiene varias propuestas sobre cómo suministrársela y de qué tipo debe ser. El problema que nos ocupa es muy sensible a la elección de las propiedades lingüísticas y consigue resultados muy dispares cuando se procesa una palabra u otra. Se demuestra empíricamente que la diferenciación del aprendizaje por palabra es recomendable y más eficaz.

Son muchos los investigadores que, a la vista de los resultados obtenidos hasta ahora, proponen la combinación de varios métodos y sistemas. Siguiendo esta línea de investigación, se mostrarán varios experimentos de cooperación entre el método mencionado y un método no supervisado, que corroboran las ventajas de este diseño.

Otro aspecto importante es el tamaño del conjunto de sentidos definidos para cada palabra, ya que existe cierta confusión sobre qué grado de detalle se necesita para, por ejemplo, traducción automática o búsqueda de respuestas. Se mostrarán resultados que demuestran que la reducción de la polisemia ayuda en gran manera a la desambiguación semántica.

Por último, buscando un sistema que asegure una alta precisión aún cuando no asigne el sentido a todas las palabras del texto, y cuya aplicación sea la adquisición automática de corpus anotados semánticamente, se presenta un algoritmo iterativo, al que hemos dado el nombre de *reentrenamiento*, derivado del coentrenamiento (*co-training*) de Blum y Mitchell (1998), que consigue frenar la degradación en la precisión de este tipo de métodos.

Índice general

Índice de cuadros

Índice de figuras

Agradecimientos

No es tanto el cuento de Caperucita como el de Pedro y el Lobo, que de tanto que mi boca anunciaba una Tesis, mis actos corroboraban todo lo contrario.

En mi caso, si no el lobo, sí la presencia constante y ominosa de Manolo Palomar, insistente, obsequioso en miradas de ésas, como que "ya vendrás a mí", por no decir que sus palabras corrían hacia el sur y sus pensamientos e intenciones hacia el norte. Supongo que fue lo que me decidió por fin a admitir que debía realizar una Tesis Doctoral.

Como buen aspirante al grado, mis primeros pasos no fueron vacilantes sino puro desasosiego. De viajes iniciáticos y transoceánicos a mis primeros congresos, éstos de pura intentona, a ver si entraba en el mundillo, pasando por la fase del sí-no-ya-veremos, hasta que mi mentor decide que el "worsensdisambigüeshion" es lo mío, y con esa clarividencia que le reconozco pero que no sé, sinceramente, de dónde le ha venido, pues aquí estoy.

El germen de lo que hasta hoy he redactado, es inocente pero conviene recordarlo, proviene de una pequeña confusión que me llevó a consultar electrónicamente (por lo del correo) a Lluis Padró, que con aviesas intenciones que le presupongo, me recomendó la lectura de todo lo que pudiera encontrar de la *máxima entropía*, y en especial de un tal Ratnaparkhi. He de reconocer que, de máxima entropía, soy hoy la máxima autoridad en mi departamento, que es como decir que soy el campeón del mundo de mi barrio, pero a él le "culpo" de esta posición de "privilegio".

Desde entonces, han pasado muchas cosas y he conocido a mucha gente. Los he visto de todos los colores, unos respetados y consolidados, otros, humildes hasta la invisibilidad, como yo, deambulando por los congresos intentando convencer a otros de que nuestro

trabajo es serio. De todos ellos, muchos me han ayudado y mucho, y agradezco y me acuerdo.

De German Rigau, que de él siempre he atesorado consejo y horizonte, y cierta sorna con la que me observaba un poco de lejos. Parece tópico decirlo así, pero sin su participación este doctorando no habría llegado a ninguna parte. De verdad.

De Lluís Màrquez, que paciencia tiene, y mucha, por leerse el borrador de este documento y aún comentarlo y enriquecerlo. De Lluis Padró, tengo que volver a él, el que su Tesis fuera inusitadamente fácil de leer para un lego como yo, lo que me hizo creer en el milagro de la investigación posible. De Ruslan Mitkov, los quites en las preguntas difíciles de los congresos en los que coincidíamos. De Alfonso Ureña y Manolo García, los buenos momentos en el congreso de Jaén y cuando aquí se han acercado, y su ayuda en todo momento. De Eneko Agirre, primero sus trabajos que siempre me han acercado un poco más a terminar el mío, y después, aunque no sé si en este orden, que siempre ha tenido la palabra amable y alentadora.

De Andrés Montoyo, sobre todo un viaje a Toulouse a 180 por hora, con el móvil en la mano, y de muchas cosas más, todas de investigación, claro. De Maxi, qué voy a decir de Maxi, faro de mis noches, sombra de mis días, empedernido destructor de mis temores, loco desatado a los mandos de una motocicleta. De Rafa, y también Gonzalo, que de tantos años juntos ya parecemos lo que no somos, que sigamos juntos unos añitos más. De Eva, mi muy querida Eva, que los puentes sigan abiertos y firmes. De Patricio y Paloma, que volvamos a Faro y a los langostinos tigre, y a Jabugo y a la gran fuente. De José Luis Vicedo, de quien recuerdo conversaciones que no podría repetir en según qué sitios. De Rafa Muñoz, que "tié un humor que no se pué aguantá". De Jesús y Fernando, que son los peores compañeros de cervezas en el extranjero.

Del resto de compañeros del departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante, a quienes debo apoyo, ayuda y cierta tolerancia al tabaco negro.

De Juanma, que a veces parece que no hablamos más que de lo mismo pero que siempre hay más que no se oye. De Belén, que pese a ser fan del cotilleo más duro, es adorable.

De Begoña, que aún siendo más dura que el pedernal, como tal, hace chispa y me sigue emocionando. Por siempre.

De mi familia, tan distante, que me sigan pareciendo tan cercanos de tan buenos y comprensivos que han sido conmigo.

De mi hermano, que está ahí, aunque no lo parezca.

De mi madre, que me aguanta y me aguanta, sin estar muy claro el porqué, pero que sé que está orgullosa.

Y, por último, de mi padre, que no asistirá a mi defensa, ni lo haría, pero que sé que sigue a mi lado aunque yo no le pueda ver.

Yo no sé si la aportación científica de todos los nombrados ha sido de peso o no en esta Tesis Doctoral, pero de todos ellos y de muchos, muchos más, a todos les debo alguna cosa y puede que éste sea el momento más adecuado para, sin más, darles las gracias.

Prólogo

De lo que significa 'sentido'
o del sentido de 'significado'

«*Juan está esperándonos en el* ***banco*** *de abajo.*»

¿Qué significa banco? ¿Qué sentido tiene dentro de esta frase? Sin consultar el diccionario, a mi cabeza saltan tres o cuatro acepciones posibles: un banco como artilugio en el que nos podemos sentar (banco "de sentarse", que se dice), una institución financiera, el banco de peces o el banco de arena. Difícil va a ser que Juan nos espere sentado sobre un banco de peces. Raro es que nos lo digan así si es que se trata del banco de arena, aunque tampoco es ilegal gramaticalmente hablando. Así que de cuatro sentidos nos quedamos tan sólo con dos, que ya es un avance, pero no hemos respondido aún a la pregunta que encabeza este párrafo.

Es cierto que el ejemplo "tiene trampa". Evidentemente, nos falta el contexto en el que la frase adquiere sentido. Cuando alguien le espeta tal oración, el otro ya debe estar sobre aviso o nos pedirá una mínima aclaración: *¿qué?*, o más elegantemente, *¿perdón?*

Tenemos que cobrar el cheque de la venta del piso. Juan está esperándonos en el banco de abajo.

Perfecto, ya está claro, banco tiene el sentido de institución financiera puesto que inmediatamente antes estábamos hablando de una operación habitual en ese ámbito. No hay problema, "bajamos", de donde sea que nos encontramos subidos, supongamos que en un

tercer piso, al encuentro de Juan; nos lo encontramos sentado en el banco que el ayuntamiento ha tenido a bien incorporar al mobiliario urbano de nuestra calle, y juntos los tres (¿por qué sólo tres?) nos dirigimos a la sucursal de la Caja de Ahorros del Monte de Piedad de Madrid que, como su nombre indica, no es un banco sino una caja de ahorros.

Si de saber qué significa una palabra dentro de una frase hablamos, y si pensamos en nosotros mismos, algún desasosiego puede abrazar nuestro corazón y a nuestra razón zancadillear. En nuestra ingenua soberbia nunca albergamos la menor duda acerca de nuestro dominio del lenguaje, y más concretamente del léxico. Cuando nos enfrentamos a un texto que hemos de comprender al tiempo que leemos (la expresión oral sufre sus propios defectos) las palabras fluyen del papel a nuestro cerebro sin casi darnos cuenta, generando ideas e imágenes que nos aportan la comprensión final. Analizando nuestro personal e intransferible proceso cognitivo me doy cuenta de la pereza que nos asalta cuando aparecen esas palabras que conocemos pero no sabemos. En realidad, si nos requirieran una definición exacta, en un tanto por ciento muy elevado de las veces dudaríamos, si no directamente mentiríamos involuntariamente.

Aún con el diccionario en la mano, entre las muchas o pocas acepciones que en él se atribuyen a una palabra, sea corriente o trasnochada, muy probablemente haya varias que se acerquen lo bastante a lo que nosotros entendemos del contexto de la frase que estamos leyendo: ¿con banco proyectamos la institución financiera en sí o el edificio en el que se encuentra esa sucursal? Ya inmersos en esta fiebre tecnológica que no sabemos si nos apadrina o nos asola, si pretendemos que una máquina haga ese trabajo por nosotros, la primera intuición que acude a nuestra mente es que ella será más capaz y dudará menos.

En efecto, un ordenador (o computador, o computadora), puede llegar a no dudar jamás pero a costa de equivocarse más de la cuenta. Los actuales métodos de desambiguación del sentido de las palabras alcanzan un 70 % de éxito. Aunque este dato proviene de un ejercicio de evaluación muy concreto que se comentará más adelante, decir que en un 30 % de los casos nuestra muy capaz máquina se equivoca es casi como preguntar "*¿para qué sirve?*".

Sin embargo, volviendo la vista hacia nosotros mismos, y reconociendo esa inconsciencia con la que nos enfrentamos al léxico diario, ¿cuál sería nuestra tasa de éxito? Son frecuentes los comentarios de aquellos que se han embarcado en la aventura de etiquetar manualmente un texto, manifestando lo arduo de la tarea y las dificultades que han tenido. A veces varias personas aportan su opinión sobre un determinado ítem y el problema se traduce en ¡ponerse de acuerdo en el sentido que se le debe asignar! Complejas estrategias se diseñan para llegar al consenso más rápido posible, generalmente por votaciones con algún votante que aporte su calidad en las dificultades.

Toda esta disertación tiene como único objetivo el dar un primer aviso acerca de la tarea que en el *procesamiento del lenguaje natural* se conoce como *resolución de la ambigüedad semántica de las palabras*: es una tarea difícil, y lo es porque ni tan siquiera para nosotros es fácil entender cómo concluimos instantáneamente que tal palabra tiene tal sentido. Falta, en mi opinión, ese toque genial que aporte el enfoque definitivo al problema, que plantee el método correcto. En ese camino nos encontramos ahora, buscando, probando y replanteando; y discutiendo, mucho y bien. El trabajo que recoge esta Tesis Doctoral es un intento de aportar nuevas soluciones y nuevos enfoques al problema de decidir mecánicamente qué sentido tienen las palabras.

Un último y discutible apunte: el nombre de la tarea que nos ocupa está muy difundido en su expresión en inglés, "*word sense disambiguation*". Yo he utilizado, hasta ahora, su directa traducción al español, "desambiguación del sentido de las palabras", como también podía haberla nominado como "desambiguación léxica", "desambiguación léxica pura", "desambiguación semántica automática" o "resolución de la ambigüedad semántica de las palabras". En todo caso, la respuesta de un lingüista, alojado en un despacho cercano al mío, no ha sido capaz de resolver sus propias dudas ni las mías, y DSP, DL, DLP o DSA no tienen la fuerza de la difusión y tampoco es nuestro objetivo. Así, espero que me permitirán la licencia de usar las siglas en inglés, WSD, como acrónimo ampliamente utilizado en este universo científico dominado por el idioma anglosajón. Que quien tenga que hacerlo me perdone.

1

Introducción

La *resolución de la ambigüedad semántica de las palabras* (de ahora en adelante WSD, de *Word Sense Disambiguation*) es una tarea que se encuadra dentro de un conjunto más amplio de técnicas llamado *procesamiento del lenguaje natural* (PLN) que, básicamente, trata los fenómenos lingüísticos de toda índole de forma mecanizada mediante ordenadores.

Concretamente, WSD trata de la asignación automática de sentidos a las palabras de un texto. Por tanto, es un intento de trasladar a un entorno mecanizado las acciones de consulta y elección en un diccionario, de consulta de las acepciones de una palabra, y de elección de una de ellas como sentido correcto. El resultado final es el etiquetado de las palabras de un texto con enlaces a definiciones del diccionario.

Dentro de ese conjunto de tareas que define el PLN, WSD es una de las difíciles, y lo es porque ni tan siquiera para nosotros es fácil entender cómo concluimos instantáneamente que tal palabra tiene este o aquel sentido. En el momento presente aún no se ha conseguido, para WSD, lo que para otras tareas como la desambiguación morfosintáctica son niveles aceptables de éxito. El objetivo es poder incorporar la información semántica al resto de tareas para hacer más útil y deseable el procesamiento lingüístico automático. Sin embargo, su aportación aún no es plenamente aprovechable con los niveles de fiabilidad deseados.

WSD es lo suficientemente compleja como para requerir la concurrencia de múltiples aproximaciones, métodos, heurísticas, etc. Dentro de esta complejidad debemos mencionar la fuerte dependencia de la fuente de los textos destinados al aprendizaje y la desambiguación, además de las características propias de cada sentido o concepto que puede obligar al refinamiento del proceso.

Se ha hecho evidente, también, que WSD precisa (o es ayudado en gran manera) de una gran cantidad de preproceso: análisis sintáctico parcial o completo de la frase, entidades, información general de contexto amplio, anáfora, patrones semánticos, etc. En realidad, el problema sigue siendo la disponibilidad de suficiente información como para decidir cuál es el sentido de un determinado contexto.

Parece comúnmente aceptada la clasificación de las aproximaciones a WSD en dos categorías muy generales: **métodos basados en el conocimiento** (*knowledge-based methods*) y **métodos basados en corpus** (*corpus-based methods*). Los primeros hacen uso del conocimiento adquirido en forma de diccionarios, tesauros, lexicones, ontologías, etc. Podemos decir que este conocimiento es preexistente al proceso de desambiguación y, en la mayoría de los casos, adquirido de forma manual. Los segundos extraen el conocimiento de grandes cantidades de ejemplos (de un corpus) mediante métodos estadísticos y aprendizaje automático. Cuando esos ejemplos están anotados previamente con la etiqueta correcta (el sentido, en nuestro caso), se dice que son métodos de **aprendizaje supervisado**, y **no supervisado** cuando no existe tal anotación. Dado el gran número de métodos y soluciones propuestos actualmente, la clasificación suele simplificarse y se habla de métodos supervisados o no, esto es, únicamente si necesitan de un corpus anotado o no.

De entre los métodos supervisados, Nosotros vamos a proponer los **modelos de probabilidad condicional de máxima entropía** (MME), introducidos para el PLN, entre otros, por (Lau et al., 1993; Lau, 1994; Berger et al., 1996; Ratnaparkhi, 1998). Defenderemos su aplicación a WSD, que es competitivo si lo comparamos con otros métodos como *naive* Bayes, listas y árboles de decisión, *boosting*, basados en ejemplos (*exemplar-based*), máquinas de vectores soporte (*support vector machines*), etc.

El **aprendizaje automático**, según (Mooney, 2003), puede interpretarse como el estudio de sistemas computacionales que mejoran la resolución de alguna tarea mediante la incorporación de la experiencia. ME se encuadra (Màrquez, 2000) en los **métodos estocásticos de aprendizaje automático**, grupo en el que también se citan *naive* Bayes, *expectation maximization*, *log-linear models* y modelos ocultos de Markov. También se le puede denominar como un **método de aprendizaje inductivo basado en vectores de atributos**.

El término **atributo** (en inglés, comúnmente, *feature*) hace referencia a una característica simple, en nuestro caso lingüística, que se puede observar y procesar. Supongamos que marcamos una palabra dentro de la frase como objetivo del aprendizaje: atributos podrían ser si es un nombre o un verbo, si forma parte del sujeto o de un complemento, si su lema puede encontrarse en un diccionario, si pertenece a una lista de nombres de empresas, si es un nombre propio, cuál es la palabra inmediatamente anterior, etc. Identificando un conjunto de esos atributos, el resultado final es un vector de valores que, por decirlo así, representa a la palabra dentro de un contexto (unas cuantas palabras a izquierda y derecha, una frase, un párrafo, un documento...). Y la misma palabra puede ser encontrada y procesada en muchos contextos diferentes. No es lo mismo la palabra banco dentro de la frase «el que está sentado en aquel banco es Juan» o en «Juan ingresó el dinero en el banco».

Debido a los avances en hardware y software, el proceso de enormes cantidades de texto se ha hecho posible y ha generado una gran cantidad de recursos en forma de texto anotado y no anotado. En el pasado reciente, la comunidad científica interesada en la resolución automática de la ambigüedad lingüística ha provocado el resurgimiento de los métodos empíricos y estadísticos. Dado que la mayoría de estos problemas pueden verse fácilmente como problemas de clasificación, las soluciones aportadas por el área de inteligencia artificial (y el aprendizaje automático en particular) han cobrado especial relevancia. Como se verá en el próximo capítulo, los métodos de aprendizaje automático supervisados, en cuanto a resultados, aventajan apreciablemente a los no supervisados basados en el conocimiento.

WSD ya es de por si dificultosa para los humanos por lo que la disponibilidad de conjuntos de ejemplos anotados es insuficiente. El siguiente paso, en el que ahora nos encontramos, es la combinación de información anotada y no anotada. Especial interés están adquiriendo, también, los métodos semisupervisados y la aplicación de las técnicas denominadas "de semilla" (*bootstrapping*) al PLN en general y a WSD en particular. La idea básica de tales métodos es simple: en lugar de realizar un aprendizaje basado en una gran cantidad de ejemplos, el proceso se transforma en sucesivos aprendizajes que se alimentan con el conocimiento adquirido en el anterior. El término "semilla" proviene del inicio de tal proceso iterativo, que se supone que

no necesita más que una mínima cantidad de conocimiento previo para comenzar el aprendizaje e ir aumentándolo progresivamente. Se trata, en definitiva, de combinar información anotada y no anotada en el proceso de aprendizaje, buscando disminuir el esfuerzo en la generación o enriquecimiento de tales recursos.

La revisión de las publicaciones de los últimos años no deja lugar a dudas acerca del interés de incorporar información semántica al análisis lingüístico automático, siendo notoria la creciente presencia de secciones dedicadas exclusivamente a WSD. Por poner un ejemplo, consultando una gran base de datos documental, no es lo mismo buscar textos relacionados con los sentidos de planta como "ser vivo" que utilizar, simplemente, la cadena de caracteres "planta" como criterio de búsqueda. Parece lógico pensar que WSD enriquecerá tareas básicas como el análisis morfosintáctico (*parsing*) actual, más o menos completo, incorporándole comprensión mediante las apropiadas etiquetadas de información semántica (*parsing* semántico).

Es este creciente interés el motivo de la creación de SENSEVAL (*International Workshop on Evaluating Word Sense Disambiguation Systems*)[1], un foro de encuentro de investigadores en WSD, donde se presentan y comparan las últimas propuestas.

El trabajo que recoge esta Tesis Doctoral es un intento de aportar nuevas soluciones y nuevos enfoques al problema de decidir mecánicamente qué sentido tienen las palabras en un contexto. Con tal propósito, hemos elegido un método de aprendizaje automático supervisado dado el auge y la importancia que tal aproximación ha adquirido en el área del PLN.

A partir de la construcción de un sistema de WSD supervisado basado en los MME, el objetivo es mejorar su rendimiento. Así, esta Tesis Doctoral tiene dos objetivos fundamentales:

Desarrollo y selección de atributos

Básicamente, cuál es la información necesaria y adecuada para aprender. Cada conjunto de ejemplos tiene su propias características, cada palabra también, y hasta cada sentido. Detectar estas características y aprovecharlas para el apren-

1 www.senseval.org

dizaje requiere, en muchos casos, un análisis de los corpus de aprendizaje previo a la construcción de los clasificadores.

Construcción de un sistema de alta precisión

Como se verá en el próximo capítulo, todavía no existe el sistema aceptablemente fiable para WSD. Nuestro sistema basado en ME, aún cuando es competitivo, puede ser mejorado. Uno de los problemas de WSD es su escasa efectividad para ser usado en otras tareas como búsqueda de respuestas. Una forma de hacerlo, aunque no sea la ideal, es primar la precisión en detrimento de la cobertura, esto es, no clasificar todas las palabras, sólo aquellas para las que tenemos una alta confianza en que la etiqueta a asignar es la correcta.

Nosotros propondremos un algoritmo iterativo, una adaptación de un método incremental en el que se usa como núcleo nuestro propio sistema de WSD, con el objetivo de asegurar esa alta precisión en la clasificación.

1.1 Contribuciones de esta Tesis Doctoral

Tras abordar los objetivos antes mencionados, estos son, en nuestra opinión, los méritos destacables de todo este trabajo de investigación.

- Los resultados del trabajo que originó la redacción de este documento se derivan de la implementación de un sistema de WSD supervisado basado en los modelos de máxima entropía. Así, se dispone de un software de propósito general que en la actualidad está siendo utilizado, aparte de en WSD, en la construcción de sistemas de reconocimiento de entidades y de análisis sintáctico parcial.

  La nuestra es una de las muy escasas aplicaciones de los modelos de ME como núcleo básico de un sistema de WSD, posiblemente por el reconocido coste computacional de estos frente a otros métodos. Sin embargo, nosotros defendemos que para esta tarea

en particular, y dados los conjuntos de datos que se manejan actualmente, tal coste no es, en realidad, mayor que en otros casos, posiblemente debido a que la implementación tenía, inicialmente, como objetivo la desambiguación léxica.

- Una vez que ya disponíamos de nuestro propio sistema, la elección de la información que debíamos suministrarle para realizar un correcto aprendizaje y clasificación fue nuestra siguiente tarea. El estudio de dichas fuentes de información reveló la disparidad de criterios de selección que se puede llegar a dar dependiendo del conjunto de ejemplos de aprendizaje y clasificación de la palabra a desambiguar.

  Es, también, un refrendo de otros trabajos similares sobre la dependencia del origen de los datos de entrenamiento y de la influencia de la selección de atributos. La principal novedad de nuestro enfoque es que realizamos agrupaciones de estos tipos de atributos en vez de explorar la contribución de cada uno por separado, consiguiendo limitar el esfuerzo computacional de tal análisis.

- Redundando en este asunto, también ofrecemos otra alternativa a la definición de atributos mediante una compactación de los mismos, reduciendo drásticamente la cantidad de proceso necesario en el aprendizaje y clasificación, al tiempo que se observa una degradación mínima de los resultados.

  Supongamos que uno de los atributos es que la palabra gran preceda a interés: lo normal, dentro de los textos que se manejan habitualmente, es que este atributo se "active" unas pocas veces, ocurriendo lo mismo con cualquier otra palabra que acompañe a interés. Al final, tendremos muchos atributos distintos que informan de observaciones que se dan muy pocas veces. Se pretende, mediante la redefinición de lo que es un atributo para el aprendizaje automático, disminuir su número al tiempo que incrementamos su frecuencia de activación.

  Si por si solos pueden contribuir positivamente a la tarea, también constituyen una fuente de información complementaria ya que pueden ayudar al resto de atributos a evitar este problema de la escasa frecuencia de los fenómenos lingüísticos utilizados comúnmente (que en inglés se conoce como *data sparseness*).

- Se prueba empíricamente que la selección de atributos puede ser beneficiosa tanto para palabras como para categorías gramaticales (nombres, verbos, adjetivos y adverbios). No obstante, nuevamente la dependencia de los datos de entrenamiento se muestra como un obstáculo difícil de salvar.
- La combinación por votación simple de nuestro sistema con un método no supervisado obtiene resultados tan buenos como otros más sofisticados. Esto nos lleva a pensar que, posiblemente, el problema de la resolución de la ambigüedad semántica de las palabras no radica tanto en los métodos utilizados como en los datos que manejamos, los ejemplos de entrenamiento y los textos donde se aplican, finalmente, para su clasificación.
- El algoritmo de reentrenamiento, alternativo a *co-training*, un conocido método de *bootstrapping*, que hemos desarrollado y que presentamos en esta Tesis Doctoral, aplicado a WSD es capaz de mantener altos niveles de precisión en la clasificación. La novedad de este algoritmo radica en la división del problema de desambiguar una palabra de n sentidos en n subproblemas binarios a los que se aplican varios filtros para asegurar la certeza en la clasificación. La comparación con el algoritmo original de coentrenamiento es ventajosa en cuanto la degradación de la precisión en el proceso iterativo puede detectarse y eliminarse, aún a costa de no clasificar todo el conjunto no etiquetado.

1.2 Esquema de la Tesis

De forma más detallada, esta Tesis Doctoral se desarrolla así:

En el siguiente capítulo 2 se describirá el problema que nos ocupa y que queremos resolver, además de realizar un resumen de las aproximaciones y soluciones más recientes, centrándonos sobre todo en las aportaciones al Senseval-2.

El capítulo 3 detallará los fundamentos de los modelos de máxima entropía, base de nuestro sistema de WSD, que será desarrollado en el capitulo 4.

Volver al índice/Tornar a l'índex

El capítulo 5 mostrará los datos referidos a la evaluación del sistema, y demostrará que es un sistema competitivo si lo comparamos con otros métodos de WSD.

Como ya se ha mencionado, los recursos para WSD, principalmente corpus anotados, precisan de un esfuerzo considerable en su confección. Uno de los objetivos a los que se espera aportar un nuevo avance trata de cómo disminuir el esfuerzo anotador. El capítulo 6 expondrá la aplicación de ME a la generación automática de corpus anotados semánticamente con una alta precisión. Para ello, se utilizará una técnica de aprendizaje incremental basada en los métodos de semilla que llamaremos *reentrenamiento*.

Finalmente, el capítulo 7 establece las conclusiones y los detalles que hemos considerado relevantes y suficientes para la defensa de esta Tesis.

2

El problema y sus soluciones actuales

¿Qué hacemos cuando ignoramos qué sentido tiene una palabra en el texto que estamos leyendo? Si realmente nos interesa, nos acercamos a la estantería, tomamos el diccionario y la consultamos. Para facilitarnos la tarea, las palabras están ordenadas alfabéticamente. Una vez encontrada, examinamos todas sus acepciones buscando aquella que se adapta mejor al contexto en el que nos hemos detenido al leer. Es como decir que la palabra X puede ser A, B, o C.

Muchas de las tareas del PLN se pueden presentar como problemas de clasificación. Clasificar consiste en identificar el tipo de una entidad, determinar la clase a la que pertenece, «*ordenar o disponer por clases*». Por ejemplo, un detector de los límites de una frase asignaría a un signo de puntuación ('.', '!' o '?') el valor $cierto$ o el valor $falso$ indicando si, efectivamente, está marcando el límite de la frase. Así mismo, un sistema WSD se puede definir como un clasificador de los sentidos de las palabras que aparecen en un texto. No obstante, entre las dos tareas existe una pequeña pero fundamental diferencia que reside en la propia definición de clase: mientras que las categorías que maneja la primera son únicamente dos, la cantidad de acepciones en que se puede clasificar una palabra depende de los sentidos que puede adquirir dependiendo del contexto en el que se encuentre.

Así pues, el objetivo general de este trabajo que consiste en la construcción de un sistema informático que asigne su sentido correcto a cada palabra de un texto, se traduce en la obtención de clasificadores para las palabras polisémicas (las monosémicas no necesitan, en principio, ningún proceso de desambiguación) que nos interesen. Subsidiariamente, se pretende que realmente clasifiquen bien, asunto éste ampliamente tratado en capítulos posteriores.

Se expondrá en las siguientes secciones de este capítulo un resumen de la situación actual, con las tendencias principales y estado tecnológico de la tarea que nos concierne.

La exposición se desarrolla a partir de los sistemas presentados en el último SENSEVAL-2 (Preiss y Yarowsky, 2001b) y ajusta el foco en los sistemas supervisados basados en corpus, sin dejar de mencionar algunos sistemas no supervisados. Se intentará reflejar el avance desde entonces hasta hoy en día.

Como ya se ha apuntado, las técnicas "de semilla", *bootstrapping* o de "mejora incremental" están adquiriendo una relevancia creciente con la esperanza de reducir el esfuerzo de generación de recursos anotados semánticamente, ya que el etiquetado automático y supervisado se inicia con un conjunto relativamente pequeño de ejemplos anotados manualmente. En nuestro trabajo se va a proponer un método basado en estas técnicas que asegura clasificaciones con una alta precisión. Así, pues, se analizará también el estado actual de este campo de investigación.

Dada la variedad de métodos de WSD y las notables diferencias en los resultados publicados, debido principalmente a los diferentes conjuntos de evaluación utilizados, la comparación se hace extremadamente difícil. Nosotros estimamos que el único foro, actualmente, en el que se puede establecer qué métodos son los más adecuados para la desambiguación semántica automática es SENSEVAL, un evento que aspira a ser marco de encuentro de la comunidad del WSD donde se pueden evaluar y comparar los sistemas de desambiguación que participan en un ejercicio controlado.

2.1 El problema

El procesamiento del lenguaje natural (PLN) es un área de investigación que, sin ser de reciente creación, ha ido adquiriendo un peso importante por las necesidades de información de la sociedad actual.

La explosiva necesidad de manejar grandes cantidades de información textual, hablada y hasta visual se debe, aunque no de forma exclusiva, al uso generalizado de internet. De repente, todo el mundo tiene a su disposición cantidades ingentes de información que no es

capaz de procesar ni gestionar adecuadamente. Por poner un ejemplo, con la diversidad idiomática de la Unión Europea no es rentable perder tiempo y recursos humanos (que se pueden dedicar a otras tareas) en, por poner un caso, traducir los documentos oficiales de unas lenguas a otras.

Así, la tendencia es a replicar la habilidad humana en la comunicación al tiempo que la mecanización permite el procesamiento de enormes cantidades de información textual. En contra tenemos que la comunicación humana es muy compleja por lo que, ante un problema de tales dimensiones, la mejor estrategia es la división en problemas más livianos y esperar que pequeños esfuerzos sumen una solución global.

Algunos de estos "pequeños" problemas son tareas intermedias: análisis sintáctico, reconocimiento de entidades, determinación del final de frase, reparación de errores sintácticos y gramaticales, resolución de la anáfora, desambiguación del sentido de las palabras, generación, etc. Otros tienen un sentido más práctico, inmediato y aplicado (a veces denominadas como "finales"): recuperación de información, extracción de información, traducción automática, generación automática de resúmenes, sistemas de diálogo, entre otros muchos más.

Para hacerlo más difícil, estos problemas no son independientes sino que unos ayudan a otros incluso de forma cíclica o retroalimentada. Por ejemplo, conocer el sentido de una palabra significa saber traducirla a otro idioma, pero si conozco su traducción puede que tenga determinado su sentido. Según Montoyo (2002), «*para diseñar un sistema de PLN se requiere abundante conocimiento sobre las estructuras del lenguaje, como son el conocimiento morfológico, sintáctico, semántico y pragmático*». Una forma clásica de estructurar estos problemas interdependientes podría incluir las siguientes fases y áreas de conocimiento:

- El conocimiento morfológico proporciona las herramientas para formar palabras, es decir cómo las palabras son construidas a partir de unidades más pequeñas.
- El conocimiento sintáctico establece cómo se deben combinar las palabras para formar oraciones correctas, además de estudiar cómo se relacionan unas con otras.

- El conocimiento semántico es el conjunto de los significados de las palabras y cómo estas se combinan para formar el significado completo de una oración.
- El conocimiento pragmático ayuda a interpretar la oración completa dentro de su contexto.

Todas estas formas de conocimiento lingüístico tienen asociadas sus propias ambigüedades: estructural, léxica, ámbito de cuantificación, función contextual y referencial.

La ambigüedad léxica se presenta cuando, al asociar a cada una de las palabras del texto la información léxico-morfológica, hay palabras que tienen más de un sentido o significado. Se distinguen dos tipos de ambigüedad léxica:

Ambigüedad léxica categorial: se presenta cuando una palabra aparte de tener diferentes significados, éstos pueden desempeñar diferentes categorías sintácticas en la oración. Por ejemplo, la palabra cura puede ser un nombre en la oración «*el cura bendijo los alimentos*», y un verbo en «*el médico cura al paciente en el hospital*».

Ambigüedad léxica semántica: se presenta en aquellas palabras que en función del contexto pueden tener un sentido u otro. De forma más precisa, se puede decir que la ambigüedad léxica semántica puede referirse tanto a *homonimia* como a *polisemia*.

La **polisemia** se da en palabras a las que les corresponden, según el contexto, varios significados. Para deshacer la ambigüedad hay que apoyarse en el contexto, el cual nos indica cuál es el significado pertinente. Por ejemplo, la palabra circuito podría tener cualquiera de los cinco sentidos expuestos a continuación[1]:

1. Terreno comprendido dentro de un perímetro cualquiera.
2. Bojeo o contorno
3. Trayecto en curva cerrada, previamente fijado para carreras de automóviles, motocicletas, bicicletas, etc.

1 Extraídos del Diccionario de la Real Academia, edición electrónica, versión 21.1.0, Espasa Calpe 1995

4. Recorrido previamente fijado que suele terminar en el punto de partida.
5. Conjunto de conductores que recorre una corriente eléctrica, y en el cual hay generalmente intercalados aparatos productores o consumidores de esta corriente.

Sin embargo, en la siguiente frase, es evidente que la palabra tiene el sentido numerado como cinco:

«*Juan ha comprobado el* ***circuito*** *del panel principal de energía.*»

Como la polisemia, la **homonimia** ofrece varios sentidos para una sola palabra. Pero esto viene motivado por la evolución histórica de una lengua, que, con el paso del tiempo, va confundiendo diferentes palabras en una única forma por evolución fonética. Por ejemplo, la palabra bala pueden tener varios sentidos:

bala como munición o

bala como paquete grande de algo.

Las palabras que presentan homonimia se dividen en dos tipos, homófonos y homógrafos. Los **homófonos:** se pronuncian igual, pero se escriben de forma diferente, porque alguno de sus grafemas se corresponden con el mismo fonema, o porque no se corresponden con ninguno. Por ejemplo, las palabras:

tubo	tuvo
onda	honda
ojear	hojear

Los **homógrafos:** se escriben y pronuncian igual. Es necesario acudir al artículo, al plural o al contexto para saber su significado. Por ejemplo, las palabras:

el corte la corte

Como apunte, simplemente queremos hacer notar que el idioma en si no es importante, es decir, WSD se puede acometer en cualquier idioma (que puede tener sus particularidades que la hagan más o menos fácil), incluso se pueden aprovechar varias lenguas a la vez, pero con un "ligero inconveniente" que hay que solucionar previamente: la disponibilidad de recursos en tal idioma, diccionarios, corpus, analizadores, reconocedores de entidades, etc.

En primer lugar es conveniente establecer las definiciones para ciertos términos que van a ser profusamente utilizados, obviamente desde el punto de vista del PLN y el WSD.

2.2 Terminología básica

Esta sección no pretende ser una descripción exhaustiva de todos los términos utilizados generalmente en el PLN, tan sólo se trata de aclarar cierto número de ellos de los que vamos a hacer un uso constante.

Corpus: palabra difícil donde las haya por las dudas que genera el uso del plural. Corpus, según el Diccionario de la Real Academia de la Lengua Española es un «*conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación*». En nuestro caso, estamos hablando de texto, un conjunto de frases, párrafos o documentos textuales. En WSD entendemos por **corpus anotado** aquél que contiene ciertas palabras (nombres, verbos, adjetivos o adverbios) etiquetados con sentidos, según las definiciones de un cierto diccionario. Esta anotación puede ser más compleja si, además, contiene información sintáctica (más o menos completa) o cualquier otro tipo de información que se considere adecuada y en el formato que sea. Por **corpus no anotado** entendemos la falta de información de sentidos[2]. Una breve descripción de algunos de ellos se puede consultar en la sección 2.3.2.

Contexto: entendemos el "contexto lingüístico" como un conjunto de palabras y toda la información sobre ellas de la que se disponga.

2 Evidentemente, anotado o no anotado es una cualidad que depende de la tarea en la que se desee utilizar el recurso. El aprovechamiento de un corpus depende de los objetivos y las tareas programadas.

Los contextos pueden abarcar el espacio que se desee: n palabras a derecha e izquierda de la palabra objetivo (la palabra a desambiguar), la oración, el párrafo, el documento, etcétera. En realidad, pueden combinarse varios de ellos, dependiendo de la información que se desea utilizar y de la fuente de la que se extraen los contextos.

La información que pueden aportar estos contextos comprende todas las características morfológicas, sintácticas y semánticas que ofrece un texto escrito en lenguaje natural y que podemos extraer y caracterizar. El que esta información sea más o menos completa depende de varios factores, no todos ellos estrictamente lingüísticos: la disponibilidad de las herramientas y recursos adecuados (etiquetadores morfo-sintácticos, analizadores parciales o totales, diccionarios, ejemplos anotados...) y más importante, quizás, lo eficaces que puedan ser.

Clase: cada una de las etiquetas o categorías en que podemos clasificar un concepto. En WSD hablamos de clasificar un contexto, una palabra, en uno de sus sentidos: el conjunto de clases posibles es el conjunto de sentidos de una palabra.

Aprendizaje: fase de un método de aprendizaje automático en la que, a partir de un corpus (anotado o no) se obtiene un modelo del lenguaje que nos permita predecir (clasificar) nuevos contextos.

Clasificación: aplicación del modelo del lenguaje previamente adquirido (por aprendizaje automático o por conocimiento) para clasificar o desambiguar un determinado contexto.

Evaluación: la evaluación no es propiamente un objetivo de la implementación de cualquier método, sino el cálculo de las medidas apropiadas que indiquen la bonanza del propio método (precisión, cobertura, etc.). Este es un nuevo obstáculo especialmente en nuestro caso ya que se necesita que el corpus de evaluación esté anotado también.

Atributos (*features*, en inglés): la información que caracteriza un contexto. En los métodos de aprendizaje automático es habitual codificar el contexto extrayendo aquellos datos lingüísticos que se consideran relevantes y útiles. Por ejemplo:

«... dado el gran **interés** *que suscitan los peinados de Beckham... ...»*

Podemos decidir que la información importante de esta frase es el conjunto de palabras y categorías gramaticales de las palabras con carga semántica cercanas a interés, nuestra palabra objetivo:

$$\{dado, gran, suscitan, peinados, verbo, adjetivo, verbo, nombre\}$$

Se espera que, tras procesar un corpus lo suficiente grande, el peso estadístico de estos datos nos proporcione indicios de que la palabra objetivo se clasifica en una determinada clase[3]. Obviamente, la información elegida para caracterizar los contextos puede ser tan completa o compleja como queramos.

Colocación: este término es una traducción directa y poco pensada de la inglesa *collocation*. Son conjuntos de palabras, dos, tres, o las que sean, que con frecuencia aparecen juntas y en un determinado orden. Las palabras compuestas entrarían dentro de esta categoría.

2.2.1 Evaluación y medidas

Medidas

Las medidas usuales en WSD se describen a continuación. Supongamos que un ejercicio consiste en clasificar N contextos, y al hacerlo acertamos A veces y nos equivocamos E veces. No necesariamente realizamos todas las N clasificaciones, puede ser que para alguna de ellas no dispongamos de suficiente información como para decidir la clase a la que pertenece (ecuación 2.1), entonces:

Precisión (*precision*): es una medida que nació para la recuperación de información pero que se aplica en general a cualquier

3 Dada la naturaleza de las tareas de clasificación que se basan en la definición de atributos, se suele decir "clasificar el contexto" o "clasificar la palabra" indistintamente.

tarea de clasificación. Se define como la razón entre aciertos y respuestas (ecuación 2.2).

Cobertura (*recall*): se calcula dividiendo la cantidad de aciertos por la cantidad de respuestas que debería haber dado el sistema de clasificación (ecuación 2.3). Cuando el clasificador responde a todas las posibles respuestas, cobertura y precisión se igualan y las dos equivalen a la **tasa de acierto** (*accuracy*), o **acierto**, simplemente.

F1: es la combinación de las precisión y cobertura en un único valor (ecuación 2.4).

Cobertura absoluta (*coverage*): esta medida informa de cuántos contextos han sido clasificados respecto del total (ecuación 2.5).

$$N >= A + E \tag{2.1}$$

$$P = \frac{A}{(A+E)} \tag{2.2}$$

$$C = \frac{A}{N} \tag{2.3}$$

$$F1 = \frac{2 \times P \times C}{P + C} \tag{2.4}$$

$$CA = \frac{A+E}{N} \tag{2.5}$$

Evaluación y comparación entre sistemas

Habitualmente, los datos se dividen en dos partes, una para entrenamiento y otra para clasificación. Es el tipo de evaluación de competiciones de tipo SENSEVAL, ya que todos los equipos de una cierta tarea tienen los mismos conjuntos de entrenamiento y test, el primero etiquetado y el segundo no. Naturalmente, el objetivo de SENSEVAL es la comparación de sistemas, pero cuando no se parte de las condiciones de esta competición, la comparación pasa por recrearlas.

Para obtener conclusiones fiables de la comparación entre dos sistemas de aprendizaje supervisado se pueden llevar a cabo varias pruebas estadísticas, de las que sólo vamos a nombrar algunas (las

mostradas en el trabajo de Dietterich (1998)). Estos estadísticos se basan en evaluaciones pareadas partiendo de las diferentes particiones de datos en ejemplos de entrenamiento y ejemplos de evaluación. El trabajo de Dietterich trata de comprobar la fiabilidad de cinco estadísticos utilizados habitualmente para determinar que un sistema es más acertado que otro.

Se parte de un conjunto de ejemplos anotados del que se extraen conjuntos de entrenamiento y evaluación de la forma que se muestra a continuación:

Prueba de McNemar: se dividen los datos en dos conjuntos, uno de entrenamiento y otro de evaluación, y se entrenan y evalúan los dos sistemas a comparar sobre éstos. La limitación de este estadístico es su simplicidad, ya que el conjunto total de ejemplos debe ser suficientemente grande para obtener datos fiables de la prueba, al tiempo que no mide la posible variación del resultado si hubiéramos elegido otro conjunto de evaluación. Se basa en la comparación, ejemplo a ejemplo, de todas las clasificaciones hechas por cada sistema, aplicando el estadístico z.

Prueba de la diferencia de dos proporciones: al igual que el anterior, sólo necesita una evaluación pareada sobre un conjunto de entrenamiento y otro de evaluación. La base de la comparación es, ahora, la tasa de error de cada sistema. Sufre las mismas limitaciones que la prueba anterior, además de la dependencia estadística entre los promedios de error obtenidos por uno y otro sistema, ya que han utilizado los mismos datos. En este caso se aplica la χ^2 de Pearson.

Prueba *t*: en este caso, para el estadístico t de Student, se realizan 30 pruebas, cada una de las cuales consiste en la partición aleatoria de los datos en conjuntos de entrenamiento y de evaluación (típicamente, usando dos tercios para entrenar y el tercio restante para evaluar). El cálculo del estadístico se basa en las diferencias entre los promedios de error obtenidos por cada sistema en cada una de las 30 pruebas. En este caso, uno de los problemas apuntados por Dietterich es el posible solapamiento de los conjuntos de evaluación, solucionado con la prueba que se expone seguidamente.

Prueba *t* para una validación cruzada: los datos se procesan usando validación cruzada, *N-fold cross-validation* (nFCV). Los datos se dividen en N partes y, alternativamente, $N-1$ partes se utilizan para aprender y una para evaluar. Las tasas de error de las N evaluaciones de los dos sistemas son la entrada de la prueba. El valor de N suele ser 10. Aquí el problema puede ser justo el contrario al anterior, el solapamiento entre los conjuntos de entrenamiento.

Prueba 5x2cv *t*: para este test se hacen 5 pruebas de 2FCV, que parten de la división aleatoria de los datos en dos conjuntos de igual tamaño, nuevamente uno para entrenar y otro para evaluar.

Dietterich llega a la conclusión de que todos son aproximaciones más que métodos estadísticos rigurosamente correctos, aunque útiles para el desarrollo y mejora de los algoritmos de aprendizaje automático, y que la elección de uno u otro depende casi más del coste computacional. Aún así, recomienda las pruebas 5x2cv y McNemar, reconociendo que su estudio puede ser excesivamente dependiente del conjunto de datos y de los algoritmos utilizados.

2.3 Recursos

2.3.1 Diccionarios, tesauros, corpus, ...

Una cuestión obvia, si hablamos de los sentidos de las palabras, es quién o qué define cuáles son esos sentidos. Todos hemos consultado algún **diccionario** (o lexicón), «*libro en el que se recogen y explican de forma ordenada voces de una o más lenguas, de una ciencia o materia determinada*», para buscar las posibles acepciones de ciertas palabras. Incluso cada uno de nosotros utiliza un "lexicón mental" por el que accedemos instantáneamente a los conceptos que tenemos organizados en nuestra memoria. Básicamente, WSD decide cuál de esos sentidos es el correcto cuando nos encontramos una palabra dentro de un contexto concreto. No obstante, no tiene porqué limitarse a este tipo de clases, dependiendo de cual sea el recurso de referencia que utilicemos. Por ejemplo, un **tesauro** organiza las palabras por un índice que sirve a un propósito determinado, la catalogación de noticias de prensa, por poner un caso. Concretamente, la

norma ISO 5964 define tesauro como «*un vocabulario de un lenguaje de indización controlado, organizado formalmente con objeto de hacer explícitas las relaciones, a priori, entre conceptos (por ejemplo, "más genérico que" o "más específico que")*».

Actualmente, sin ser el único, por su difusión y por ser el diccionario electrónico usado en algunas de las tareas de SENSEVAL, **WordNet** (FellBaum, 1998; Miller et al., 1990) es la fuente de definiciones más común en PLN. Es una base de datos léxica en la que los conceptos se definen por grupos de sinónimos denominados *synsets*, y donde estos mismos conceptos se estructuran mediante relaciones semánticas del tipo "es-un", "es-parte-de", "es-substancia-de", entre otros. Actualmente coexisten varias versiones para diversos idiomas.

WordNet (WN) nació para el inglés y se ha convertido en el repositorio de sentidos para muchas de las tareas del PLN que precisan conocimiento semántico. **EuroWordNet** (EWN) (Vossen, 1998) proporciona una estructura multilingüe a la idea original de WordNet estableciendo el denominado *Inter-lingual-index* (ILI) como el conjunto de conceptos que todas las lenguas involucradas poseen, y enlazando sus *synsets* propios a este índice. Entre otros, cabe mencionar los WN en euskera (Agirre et al., 2002), catalán (Benítez et al., 1998) y español (Vossen et al., 1998).

El resultado es un conjunto de bases de datos de tipo WN que pueden manejarse independientemente unas de otras pero que, al mismo tiempo, se alinean mediante el ILI, cuya base conceptual es la versión 1.5 de WN para el inglés.

Con el paso del tiempo se han ido generando diversos recursos para el PLN en forma de corpus anotados semánticamente, etiquetas de dominio, concordancias, etc. ***Semcor*** 1.6 (FellBaum, 1998) es un subconjunto de artículos del *Brown Corpus* (Kučera y Francis, 1997) anotado con sentidos de WN1.6. ***WordNet Domains*** (Magnini y Strapparava, 2000) es un conjunto de etiquetas de dominio que catalogan sentidos de nombres de WN1.6. Es curioso, por ejemplo, el caso del corpus DSO (Ng y Lee, 1996) que está etiquetado con sentidos de un WN1.5 previo a su versión definitiva, por lo que no coincide exactamente la anotación con el WN disponible. Tampoco es un detalle insalvable y, sin ir más lejos, se dispone de mapas de conexión entre las versiones (Daudé et al., 2000) lo que, con los cui-

dados necesarios, hace que prácticamente todos los recursos puedan compatibilizarse y usarse conjuntamente.

WN no es el único diccionario electrónico. El *Longman's Dictionary of Contemporary English* (LDOCE) tiene una versión electrónica que ha sido utilizada en múltiples trabajos (Cowie et al., 1992; Yarowsky, 1992; Wilks y Stevenson, 1997; Bruce y Wiebe, 1994). El éxito de WN radica, en nuestra opinión, en su estructuración y en la riqueza de las interrelaciones en él representadas. Aparte, claro está, de su libre uso para investigación.

Sentidos de WordNet

WN también puede verse como un diccionario electrónico. Lo que diferencia a WN de los diccionarios convencionales es su estructuración en *synsets*. Es decir, el diccionario almacena conceptos que se representan, en primera instancia, por un conjunto de palabras sinónimas. Además, aunque no en todos los casos, puede ir acompañado de una glosa o texto descriptivo y de uno o varios ejemplos de uso en forma de citas. Cuando buscamos una palabra en WN se nos devolverá un conjunto finito de synsets, cada uno de ellos representando un concepto o idea, una acepción. Está organizado según las cuatro categorías de palabras con contenido léxico: nombres, verbos, adjetivos y adverbios. Los *synsets* están ligados entre sí por varios tipos de relaciones: hiperonimia, meronimia, troponimia, etc., siendo muy pocas las relaciones entre palabras de distinta categoría.

En definitiva cada *synset* representa un sentido, un concepto que se puede expresar con una o muy pocas palabras. La base de datos es muy flexible a la hora de identificar un *synset*: se puede utilizar un código interno (un número entero que coincide con el desplazamiento, *offset*, en el fichero del registro correspondiente), un par (*palabra, número*) queriendo decir que hablamos del "sentido n de la *palabra*", o una clave codificada (los llamados *sensekeys*) que suelen ser más estables entre versiones que los sentidos.

Sin embargo, algunos conceptos representados en WN se diferencian entre ellos por matices tan sutiles que a veces no es fácil ver tal diferencia. La mayor parte de estos casos se pueden achacar a la ambición del proyecto y simplemente son errores que se van depurando en las sucesivas versiones. Este "grano fino" ha sido puesto en

cuestión (Slator y Wilks, 1987; Resnik y Yarowsky, 1999; Mihalcea y Moldovan, 2001a; Palmer et al., 2004), sobre todo porque no está clara su necesidad para ciertas tareas, como pueda ser la recuperación de información.

Puede que el problema de WN sea su falta de sistema y la arbitrariedad con la que en algunos casos se ha optado por una solución y en otros por otra. En muchos casos, el problema viene de una mala representación que podría solucionarse con herencia múltiple.

El siguiente ejemplo es de WN2.0, donde se muestra el primer sentido del nombre en inglés playing (los otros dos sentidos de playing.n son de juego–deporte y teatro):

```
3 senses of playing

Sense 1

playing -- (the act of playing a musical instrument)
  RELATED TO->(verb) play#3
     => play -- (play on an instrument;"The band played
           all night long")
  RELATED TO->(verb) play#7
     => play -- (perform music on (a musical instrument); "He
           plays the flute";"Can you play on this old recorder?")
  RELATED TO->(verb) play#6
     => play, spiel -- (replay (as a melody); "Play it again,
           Sam"; "She played the third movement very
           beautifully")
```

A su vez, los dominios del verbo play son los siguientes (play.v#6 no tiene asignado dominio):

```
Sense 3 play -- (play on an instrument; "The band played all night
long")
       CATEGORY->(noun) music#1

Sense 7 play -- (perform music on (a musical instrument); "He
plays the flute"; "Can you play on this old recorder?")
       CATEGORY->(noun) music#1
       CATEGORY->(noun) music#3
```

En nuestra opinión, las diferencias entre los sentidos play.v#3, play.v#7 y play.v#6 son harto difíciles de ver. Además, ¿por qué a esos sentidos se les ha asociado en un caso un dominio, en otro dos

y en el último ninguno? ¿Bastaría con un sólo sentido de play.v y uno de music.n?

Además, seguramente, el problema es que cada tarea puede requerir un repertorio de sentidos distinto, mientras que la tónica general es utilizar WN en cualquier caso y sin reparar en si la distinción de sentidos es apropiada o no.

Generalización de los sentidos de WordNet

Relacionado con la reducción de la polisemia en WN, se propone utilizar conceptos más generales que actúan como agrupaciones que engloban a varios conceptos de más específicos.

La generalización más inmediata se encuentra en WN como categorías lexicográficas, o como *Top Ontology* en EWN Vossen et al. (1997)[4]. Concretamente, las categorías lexicográficas de las distintas versiones de WN en inglés generan el problema contrario: categorías como 'animal', 'vegetal' o 'manufacturado' son tan generales que su poder de discriminación y su utilidad parecen discutibles. La *Top Ontology* y los conceptos base del proyecto EuroWordNet son un intento de alcanzar un nivel intermedio entre las representaciones de bajo y alto nivel del WN original en el que se basa pero, al mismo tiempo, relativamente difíciles de manejar. Por ejemplo, se han utilizado para extraer conocimiento en forma de patrones o relaciones entre categorías (Saiz-Noeda et al., 2001).

Otros investigadores (Magnini y Strapparava, 2000; Montoyo et al., 2001) proponen la utilización de etiquetas de dominio, que sin ser tan minuciosas como un synset no son tan generales como las categorías lexicográficas. Dependerá de la extensión del conjunto de etiquetas el que tenga mayor o menor detalle. En general, podemos incluir dentro de estas generalizaciones a los sistemas de clasificación ya existentes como IPTC[5], u otros similares. Estos son listas de descriptores utilizados para clasificar noticias de prensa o para catalogación bibliográfica, por ejemplo. En el momento actual, ciertas

4 Puesto que EWN se basa en WN, las categorías lexicográficas también le son aplicables.

5 El *IPTC Subject Reference System* (`www.iptc.org`) se desarrolló para suministrar a los servicios de información un sistema de codificación universal independiente del lenguaje para la clasificación de noticias por temas.

tareas de búsqueda y clasificación automatizada de documentos parecen más abordables desde el punto de vista de estos lexicones, e incluso más eficaces.

Especial interés merece el trabajo (y recurso) *WordNet Domains* (Magnini y Strapparava, 2000; Magnini et al., 2001). Consiste en una compactación de los sentidos de nombres de WN1.6 en categorías y sus desarrolladores tienen la intención de que sea útil tanto para WSD como para toda tarea a la que se quiera incorporar semántica.

Para el problema que nos ocupa, la asignación de un significado a una palabra en función del contexto en el que se encuentra dicha palabra, el primer paso debería ser la elección del conjunto de significados, la definición de las clases. Se piensa que si se encuentra una buena solución para significados muy detallados, esta solución funcionará igualmente, y mejor, con un conjunto de clases de más alto nivel y, por tanto, de menor polisemia. En este trabajo de Tesis se utilizan los *synsets* de WN como el conjunto de clases, pero también se realiza un pequeño estudio basado en *WordNet Domains* (véase la sección 5.4), llegándose a la conclusión de que la reducción de la polisemia ayuda a incrementar la tasa de acierto de nuestro método supervisado.

eXtended WordNet

Es un proyecto del *Human Language Technology Research Institute* (*University of Texas at Dallas*) y los detalles se pueden consultar en (Mihalcea y Moldovan, 2001b). WN ha sido enriquecido mediante la incorporación de la siguiente información:

- Análisis sintáctico de las glosas (separando definiciones y ejemplos).
- Anotación con sentidos de WN1.7 de esas glosas
- Transformación en formas lógicas.
- Relación entre conceptos (incluso entre varias categorías sintácticas) por su asociación a un contexto o dominio particular.

El objetivo del proyecto es crear una herramienta que sea capaz de realizar todas estas tareas sobre cualquier versión presente o futura de WN.

La primera versión está ya disponible en `xwn.hlt.utdallas.edu`

2.3.2 Corpus

Los corpus, los conjuntos de ejemplos anotados que nutren el aprendizaje automático en WSD, son pocos actualmente, y se discute mucho sobre el tamaño mínimo necesario y su calidad. Por eso se habla del "cuello de botella en la adquisición de conocimiento" (Gale et al., 1993), refiriéndose a la falta de recursos anotados actual y a la dificultad para obtenerlos.

Lo que a continuación se muestra es una descripción somera de algunos de estos recursos disponibles. Nos vamos a centrar en los conjuntos especialmente útiles para WSD que son los corpus anotados semánticamente, o que han sido utilizados expresamente para la tarea[6].

SemCor: *Semcor* 1.6 (FellBaum, 1998) es un subconjunto de artículos del *Brown Corpus* (Kučera y Francis, 1997) y la novela "*The Red Badge of Courage*" de Stephen Crane. En total son 23.346 lemas en 234.113 ejemplos. SemCor respeta la organización del corpus origen y los nombres de los ficheros informan del tópico bajo el que se agrupan (reportajes de prensa, editoriales, religión, etc.)[7].

DSO: el corpus DSO (Ng y Lee, 1996) contiene 192.800 ejemplos de los 121 nombres y 70 verbos ambiguos más frecuentes en inglés. Consiste en frases sueltas extraídas del *Brown Corpus* y del *Wall Street Journal*, y la única anotación de cada ejemplo es el sentido de WN1.5[8] de la palabra elegida. El corpus está organizado en ficheros, cada uno de ellos referido a una palabra concreta.

6 Una buena recopilación de este tipo de recursos se puede encontrar en `www.senseval.org`.

7 La estructuración de este corpus se puede consultar en `clwww.essex.ac.uk/w3c/corpus_ling/content/corpora/list/private/brown/brown.html`.

8 Como ya se ha mencionado anteriormente, no se corresponde exactamente con la versión 1.5 final.

Este corpus está disponible en el *Linguistic Data Consortium* (LDC)[9].

Senseval: en realidad nos referimos a SENSEVAL-2, aunque los corpus del primer SENSEVAL también se pueden encontrar en este formato. Las fuentes de los textos para inglés y español fue el *British National Corpus* y la agencia EFE, respectivamente.

line, hard, serve: (Leacock et al., 1993, 1998): mas de 12.000 ejemplos de esas 3 palabras extraídos de *Wall Street Journal*, *American Printing House for the Blind*, *San Jose Mercury*, y anotados con sentidos de WN1.5. Disponibles en `www.d.umn.edu/~tpederse/data.html`

interest corpus: 2,369 ejemplos, anotados con definiciones de LDOCE. La fuente es el *Wall Street Journal*. (Bruce y Wiebe, 1994). `crl.nmsu.edu/cgi-bin/Tools/CLR/clrcat\#I9`

Open Mind Word Expert: 230 lemas en 70.000 ejemplos, incluyendo duplicados pero, puesto que es una página web de libre acceso que permite la anotación de contextos preseleccionados por cualquier persona interesada en participar, estos números se están incrementando diariamente. Está anotado con WN1.7 y las fuentes son *Penn treebank*, *LA Times*, y otros (Chklovski y Mihalcea, 2002), `www.teach-computers.org/word-expert.html`

eXtended WordNet: dado que incorpora la desambiguación de las glosas de WN1.7 es un corpus anotado (con mucha información adicional).

A continuación, y como paso previo al cuerpo central de esta Tesis, vamos a describir los métodos existentes para la resolución de la ambigüedad semántica de las palabras o WSD, haciendo hincapié en los métodos de aprendizaje supervisado y los últimos sistemas presentados en SENSEVAL-2.

9 `www.ldc.upenn.edu`

2.4 Una clasificación de métodos de WSD

Aunque existen otras clasificaciones más pormenorizadas (Ide y Véronis, 1998; Wilks y Stevenson, 1996), dada la profusión de métodos, nosotros nos apoyamos en la muy general que utiliza el propio SENSEVAL: métodos supervisados y no supervisados.

Se entiende por **método supervisado** aquel que utiliza un conjunto de ejemplos previamente etiquetados con sentidos (un corpus anotado) de los que extrae el conocimiento lingüístico.

Por contra, son **no supervisados** los que capturan el conocimiento lingüístico sin necesidad de la anotación previa manual de gran cantidad de ejemplos. Utilizan corpus no anotados, diccionarios electrónicos, tesauros, etc. La ventaja de estos frente a los supervisados es que no dependen de un corpus anotado sino de su habilidad al utilizar los recursos mencionados.

Otra clasificación complementaria es la de métodos **basados en conocimiento** (*knowledge-based*) y **basados en corpus** (*corpus-based*) (Pedersen, 2001a). Los primeros utilizan conocimiento lingüístico previamente adquirido (diccionarios, tesauros, etc.) y los segundos usan técnicas estadísticas y aprendizaje automático para inducir modelos de uso del lenguaje a partir de una gran cantidad de ejemplos de texto. Estos métodos realizan *aprendizaje supervisado* si los ejemplos están etiquetados y *aprendizaje no supervisado* cuando no lo están.

En realidad, la primera es un resumen cómodo de la segunda clasificación (que a su vez lo es de las otras más detalladas). Lo cierto es que se ha llegado a tal punto de complejidad en los sistemas que parece que la única diferencia clara ahora mismo es si utilizan corpus anotado o no. De hecho, la batalla la están ganando ahora mismo, y de momento, los métodos supervisados, con una clara diferencia de acierto respecto de los no supervisados, como se verá a continuación en los comentarios relativos al SENSEVAL-2.

2.5 SENSEVAL-2

El primer SENSEVAL culminó en septiembre de 1998 en un *workshop* en Herstmonceux Castle, en Inglaterra (Kilgarriff y Rosenzweig, 2000). Nació como foro específico de WSD donde los investigadores exponían sus avances en forma de competición al estilo de las *Message Understanding Conference* (MUC), la *Text Retrieval Conference* (TREC) o *Cross-Language Evaluation Forum* (CLEF). Dado el éxito obtenido, el pasado julio de 2001 se celebró la segunda edición bajo en el marco de la conferencia anual de la *Association for Computational Linguistics*, ACL, en Toulouse, Francia, y la tercera tendrá lugar en Barcelona en julio de este año 2004.

En SENSEVAL, con el debido tiempo de antelación, se definen las tareas y se suministran los recursos necesarios para que todos aquellos investigadores o instituciones que quieran participar puedan desarrollar o preparar sus sistemas de desambiguación y, finalmente, bajo unas mismas condiciones para todos, comparar los resultados obtenidos por todos ellos sobre un mismo banco de pruebas de WSD. El objetivo, tal y como remarca Edmonds (2001), no es premiar a un ganador sino «*explorar los aspectos científicos de la desambiguación semántica automática*».

En este último SENSEVAL-2 las tareas propuestas fueron *all-words*, *lexical sample* y *translation*, las dos primeras multiplicadas por varios idiomas mientras que la última sólo para japonés.

La tarea "completa" (*all-words*) tenía por objeto desambiguar la mayor parte de los nombres, verbos y adjetivos de un conjunto de textos seleccionados. Había una tarea de desambiguación completa para cada uno de los idiomas propuestos: checo, holandés, inglés y estonio.

La tarea "muestra léxica" (*lexical sample*) se basa en la preselección de un conjunto limitado de palabras. De cada una de esas palabras se recolecta un conjunto de ejemplos que se anotan manualmente y que sirven de corpus de aprendizaje (si es que se quiere utilizar). Estos ejemplos son, en general, un párrafo más o menos extenso donde se anota una única instancia de la palabra seleccionada con su sentido correcto, y el resto de palabras actúa como su contexto. Como ejercicio, se confecciona otro corpus de evaluación donde,

en otros contextos, se marca la misma palabra pero sin especificar su sentido correcto (el ejercicio, evidentemente, consiste en asignar ese sentido). La tarea se subdivide en las correspondientes para inglés, español, euskera, italiano, japonés, coreano y sueco. La selección de estas palabras y la confección de los corpus de aprendizaje y evaluación son responsabilidad de los comités correspondientes (uno por idioma) que aplican para ello los criterios y objetivos que consideran interesantes.

Los sistemas y resultados del primer SENSEVAL se presentan en los trabajos de Kilgarriff y Palmer (2000) y Kilgarriff y Rosenzweig (2000) y, junto con los datos del último SENSEVAL-2, pueden consultarse en `www.senseval.org`.

2.5.1 Tareas de inglés y español

El cuadro 2.1 muestra los sistemas que se presentaron a la tarea completa de inglés y a las muestras léxicas en inglés y español. El cuadro especifica si los sistemas presentados fueron supervisados o no, mixtos o si presentaron varios de varios tipos. Además se informa de las tareas en las que participaron. Una descripción más completa se puede consultar también en el cuadro A.1, en el anejo A.

Afiliación	Sistemas	Contacto	Tipo	AE	LE	LS
University of Maryland	UMD-SST:	Resnik et al.	S	X	X	X
University Basque Country	ehu-dlist	Agirre & Martinez	S	X	X	
Lab. Informatique d'Avignon	LIA-Sinequa	Crestan et al.	S	X	X	
University of Antwerp	ANTWERP	Hoste	S	X		
University of California	University_California	Chao	S	X		
Stanford University	CS224N	Manning	S		X	X
Univ. of Minnesota Duluth	Duluth1,..., DuluthC	Pedersen	S		X	X
Johns Hopkins University	JHU	Yarowsky et al.	S		X	X
Korea University	Kunlp	Seo, Lee & Rim	S		X	
Tech University of Catalonia	TALP	Escudero & Rigau	S		X	
UNED University	UNED	Fernandez-Amoros	NS	X	X	
CL Research	DIMAP	Litkowski	NS	X	X	
Illinois Institute of Technology	IIT	Haynes	NS	X	X	
Instituto Trentino di Cultura	irst-eng	Magnini	NS	X	X	
University of Sussex	Sussex-sel	Carroll &McCarthy	NS	X		
Universiti Sains Malaysia	usm_english_tagger	Guo	NS	X		
University of Sheffield	University_Sheffield	Preiss	NS	X		
ITRI (University of Brighton)	WASPS-Workbench	Tugwell	NS		X	
Southern Methodist Univ.	SMU	Mihalcea	M	X	X	
University of Alicante	Univ._Alicante_System	Montoyo & Suarez	M		X	X

Tipo: S Supervisado — NS No supervisado — M Mixto

AE: English All-words — LE: English lexical sample — LS: Spanish lexical sample

Cuadro 2.1. Selección de sistemas participantes en SENSEVAL-2

Sin entrar en detalles (se pueden consultar en internet), el resumen de SENSEVAL-2 para las tareas en inglés y español es el que se muestra en el cuadro 2.2. La columna 'acierto' muestra las puntuaciones globales máximas, obtenidas por sistemas supervisados. Como comparación entre las dos aproximaciones al problema, también se muestran las puntuaciones obtenidas por los mejores no supervisados[10].

Tarea	Acierto	No supervisado
English All-words	69,0	57,5
English lexical sample	64,2	58,1
Spanish lexical sample	71,2	-

Cuadro 2.2. Máximas puntuaciones obtenidas en SENSEVAL-2 (datos porcentuales)

Con estos resultados podemos concluir que la precisión en la tarea WSD ronda el 70 %, muy lejos, por tanto, de otros niveles de precisión a los que nos tienen acostumbrados otras tareas intermedias como *POS-tagging*.

2.5.2 Descripción de sistemas seleccionados

Hemos seleccionado algunos de los sistemas participantes en SENSEVAL-2, en su mayoría supervisados, atendiendo a su interés por el nivel de precisión alcanzado o por utilizar algún método que consideramos necesario mencionar.

El sistema *UMD-SST*

Cabezas et al. (2001) propone un método supervisado basado en el aprendizaje mediante **máquinas de vectores soporte** (SVM). Participaron en las tareas para inglés, español, sueco y euskera.

10 En SENSEVAL-2 las medidas utilizadas fueron precisión, cobertura y cobertura absoluta, pero gran parte de los sistemas, entre ellos los que obtuvieron la máxima puntuación, cubrieron el 100 % de los contextos de test, por lo que se muestra únicamente la precisión

Después de caracterizar los contextos de cada una de las palabras a desambiguar con funciones de atributo (*features*) se construyen clasificadores para cada sentido posible, de tal forma que los ejemplos positivos de ese sentido son negativos para los otros (*one against all*). Ante un contexto concreto, se elige el clasificador SVM que «*responde 'sí' con más fuerza*».

SVM es apropiado para aprendizaje automático de clasificadores binarios (Voser et al., 1992; Vapnik, 1995, 1998). En un problema de clasificación cuyos contextos se describen en forma de vectores de atributos, se calcula el hiperplano lineal que separa el conjunto de ejemplos positivos del de negativos con el máximo margen (la distancia del hiperplano al ejemplo más cercano de positivos y negativos). Recientemente, podemos encontrar aplicaciones a la desambiguación en PLN como clasificación de textos (Joachims, 1998), *chunking* (Kudo y Matsumoto, 2001), análisis sintáctico (Collins y Singer, 2002), WSD (Murata et al., 2001; Lee y Ng, 2002) o para la detección de errores en el corpus (Nakagawa et al., 2002).

El sistema *TALP*

El sistema presentado por Escudero et al. (2001) se basa en al algoritmo ***LazyBoosting*** (Escudero et al., 2000a) que es una variante del *AdaBoost.HM* (Schapire y Singer, 1999). Los algoritmos basados en *boosting* realizan la clasificación mediante la combinación de varios clasificadores simples (o débiles, *weak classifiers*) que no necesariamente obtienen una alta tasa de clasificaciones correctas por si solos. *LazyBoosting* reduce el espacio de atributos que maneja *AdaBoost.MH* lo que claramente disminuye el coste computacional al tiempo que no degrada el acierto.

Este módulo, que denominaremos central, fue enriquecido con información de dominio y descomposición jerárquica del problema de clasificación multiclase, inspirado en el trabajo de Yarowsky (2000).

Aparte del contexto local, una ventana de menos y más tres palabras a partir de la palabra objetivo, añadieron información de dominio utilizando las etiquetas del *WordNet Domains*. Para todos los nombres del contexto se tenían en cuenta todas las posibles etiquetas de dominio para todos los posibles sentidos y se anotaban aquellas

que obtenían la máxima puntuación. Finalmente, estas etiquetas se añadían al sistema en forma de atributos.

La desambiguación la descompusieron en dos pasos: clasificadores de categoría semántica y clasificadores de sentidos de WN. El primer nivel de clasificadores utilizaba las categorías lexicográficas de WN como etiquetas de clase, lo que delimitaba los sentidos posibles del segundo nivel de clasificadores.

Dados los resultados obtenidos, los autores llegan a la conclusión de que la información de dominio es productiva mientras que, sorprendentemente, la descomposición jerárquica en dos niveles disminuye el acierto global del sistema (algunas palabras sí incrementaban su precisión). Exponen como posible explicación a este segundo dato la posibilidad de que la separación conceptual de alto nivel es demasiado general como para ser utilizada con éxito y, al mismo tiempo, la distancia semántica entre algunas de estas clases puede no ser lo suficientemente grande.

Los sistemas *UNED*

Fernández-Amorós et al. (2001) presentaron un sistema no supervisado basado la medida denominada "información mutua" (***mutual information***) y en heurísticas. Para realizar la desambiguación de todas las palabras se utiliza una matriz de relevancia entre palabras calculada con los datos de 3200 libros de inglés del *Gutenberg Project* (`promo.net/pg/`). Esta matriz es sensible a las distancias entre las palabras en el corpus. La idea es medir la distancia de las definiciones de WN con el contexto a desambiguar y asignar el sentido más cercano. Con ayuda de la matriz se enriquecen las descripciones de los sentidos con las relaciones de hiponimia (cuando es posible hacerlo).

Mencionan una variante supervisada para la *English lexical sample task* ya que incorporan el conjunto de ejemplos de entrenamiento a las definiciones de los sentidos.

El sistema *SMUaw*

Mihalcea (2002); Mihalcea y Moldovan (2001c, 2000, 1999) basan el proceso de desambiguación en dos subsistemas, un aprendizaje de patrones a partir de WN (de sus glosas), SemCor y GenCor (un corpus anotado semánticamente de forma automática mediante una serie de heurísticas utilizando los dos anteriores como semilla), y un sistema *instance-based* con selección activa de atributos (*active feature selection*).

Los **patrones** son parejas de palabras de tal forma cuando se encuentra (w_{-1}, w) en el texto a desambiguar, w tiene el sentido i si en el aprendizaje estas dos palabras iban siempre asociadas a ese sentido y con una frecuencia suficiente. Se busca la máxima fiabilidad a la hora de asignar la etiqueta, aunque sea a costa de no etiquetar muchas de la ocurrencias, hecho este último que se soluciona con diversas heurísticas o procedimientos en cascada.

El subsistema supervisado se utiliza para las palabras de las que se dispone de suficiente cantidad de ejemplos anotados de entrenamiento. Los algoritmos ***instance-based*** (Aha et al., 1991) pertenecen a la familia de los métodos de aprendizaje inductivo que clasifican nuevos contextos mediante una cierta medida de distancia con los ejemplos (por ejemplo, el vecino más próximo). También se les conoce como *memory-based*, *example-based*, *exemplar-based*, o *similarity-based*: el entrenamiento se reduce a almacenar los ejemplos en memoria y la clasificación de nuevos contextos se realiza por similitud con los ejemplos. Cómo virtudes de estos métodos se cita el que no desecha los ejemplos menos frecuentes, «*no olvida las excepciones*».

Al subsistema anterior se le añade un sistema de selección automática de atributos mediante un proceso incremental basado en pruebas 10FCV sobre los corpus anotados, buscando los atributos más informativos o eficaces para cada palabra. De hecho, la ***selección activa de atributos*** puede verse como un "aprendizaje antes del aprendizaje" ya que, dada una palabra, el módulo *instance-based* sólo trabaja con los atributos que se presume más útiles para cada palabra.

En resumen, el sistema combina las dos aproximaciones, supervisado y no supervisado, dependiendo de si se dispone de un cor-

pus de entrenamiento suficiente o no. Aparte, aplican un preproceso bastante completo de análisis sintáctico, búsqueda de entidades, palabras compuestas, etc.

El sistema *Antwerp*

Hoste et al. (2001) entrenan su sistema con SemCor. El sistema combina varios clasificadores basados en ***memory-based*** *learning (TiMBL)* y ***rule induction*** *(Ripper)*, utilizando información parcial y complementaria de los contextos. La clasificación se realiza por votación y en caso de duda, se devuelve al sentido más frecuente de WordNet. La arquitectura del sistema permite que los procesos de entrenamiento se ejecuten en paralelo.

El sistema *JHU-English*

Yarowsky et al. (2001) presentaron un sistema de votación sobre los resultados obtenidos de varios subsistemas de aprendizaje supervisados. Estos subsistemas se basan en ***decision lists*** (Yarowsky, 2000), ***cosine-based model***, y ***naive-Bayes***. Para cada subsistema las características que se incluyen no son solamente un conjunto de palabras en una ventana fija de contexto, sino que también incluyen una variedad de características sintácticas como sujetos, objetos directos, complementos circunstanciales y varias relaciones modificadoras de nombres y adjetivos.

Los sistemas *CS224N*

Este sistema, coordinado por Ilhan et al. (2001), es una combinación de varios y variados sistemas supervisados. Lo curioso de este sistema es la forma de recolectar los subsistemas, siete elegidos de entre 23 candidatos desarrollados por estudiantes. A éstos se les dio libertad para elegir el método, aunque casi todos eran, principalmente, versiones de ***naive*-Bayes**, algunos había basados en ***vector space***, ***memory based*** y otros. Para cada palabra, todos los sistemas fueron clasificados por acierto en una prueba 5FCV. Finalmente,

también para cada palabra, se descartaban los que obtenían peores resultados.

El ensamblaje de estos subsistemas elegidos se hacía mediante votación por mayoría, votación ponderada y máxima entropía. El sistema de máxima entropía es entrenado con las respuestas de cada clasificador como información para la predicción posterior. El sistema ponderado asigna pesos a cada uno según la probabilidad de que la respuesta de un sistema sea la correcta de todas las propuestas.

Finalmente se realiza un meta-aprendizaje, cada combinación de sistemas es probada sobre el conjunto de entrenamiento para cada palabra en un 5FCV. El resultado del entrenamiento global es un sistema de combinación y un conjunto de subsistemas de clasificación para cada palabra, el mejor en el entrenamiento, que será aplicado al conjunto de evaluación cuando esa misma palabra sea la palabra objetivo.

El sistema *ITC-irst*

Magnini et al. (2001) utilizaron un recurso propio ya mencionado, *WordNet Domains*. El clasificador, en su fase de entrenamiento, aprende valores de frecuencia relativa a cada dominio definido para un lema concreto, utilizando SemCor.

Esta información se estructura en forma de vectores de dominio, representando los dominios relevantes para cada lema o sentido en un contexto. Tienen en cuenta tres tipos de vectores dependiendo de si es un contexto de evaluación o de entrenamiento, y en este último caso, de si se dispone de ejemplos de entrenamiento o no.

La clasificación se lleva a cabo comparando el vector de un lema en un contexto de evaluación con los vectores de entrenamiento correspondientes a cada sentido posible. La comparación es una medida de similitud calculada con el producto escalar entre vectores. El sentido que obtiene el máximo valor es el asignado finalmente.

Los sistemas *ehu-dlist*

Martínez y Agirre (2001) se basan en las **listas de decisión** (*decision lists*) de Yarowsky (1994) para presentar tres sistemas a la tarea

completa de inglés (que entrenaron con Semcor), a la muestra léxica de inglés y a la muestra léxica en euskera.

Todos los sistemas tenían dos versiones, una que utilizaba todos los atributos de aprendizaje, y otra que seleccionaba los mejores en aras de conseguir una precisión del 85% sin importar la cobertura que se alcanzara.

Para las tareas en inglés, tales atributos eran los ya definidos en el trabajo de Yarowsky, mientras que para el euskera se basaron en el análisis morfológico del texto. Esta decisión se fundamenta en las grandes diferencias entre las dos lenguas.

Los sistemas *sussex-sel*

McCarthy et al. (2001) presentaron varios sistemas basados en la adquisición automática de **preferencias de selección** (*selectional preferences*), resolución de la anáfora y la aplicación de la heurística "un sentido por discurso".

Los sistemas *duluth*

Pedersen (2001b, 2002b) propone una metodología de referencia para WSD que se basa en el aprendizaje de árboles de decisión (***decision trees***) y clasificadores *naive-Bayes*. El trabajo presenta varios sistemas combinando los métodos mencionados, pero cada uno entrenado con conjuntos de atributos diferentes.

Posteriormente Pedersen (2002a) realizó un estudio comparativo de los sistemas participantes en SENSEVAL-2 en las tareas *lexical-sample* para inglés y español. Defiende, como en trabajos suyos anteriores, la existencia de instancias difíciles de resolver, aparte de la disponibilidad o no de suficientes ejemplos de entrenamiento, dificultad que cree evidente puesto que la mayoría de los sistemas no pudieron resolverlas satisfactoriamente.

El sistema *KUNLP*

El sistema de la *Korea University* (Seo et al., 2001) utiliza un modelo de información de clasificación (***classification information model***) basado en la teoría de la entropía que calcula lo que ellos denominan *Discrimination Score*, una medida de la confianza en la decisión de un atributo al asignar su clase más probable.

En el sistema de WSD preparado para SENSEVAL-2, la información de contexto que utilizaron fue de tipo local, de dominio y bigramas.

El sistema *Alicante*

Nuestro sistema (Montoyo y Suárez, 2001) era una combinación de dos métodos, ***marcas de especifidad*** (SM) y ***máxima entropía*** (ME). Tal combinación consistía en utilizar SM para clasificar nombres y ME para verbos y adjetivos. El hecho de ser no supervisado el primero y supervisado el segundo tuvo dos consecuencias principales: el sistema al completo fue considerado supervisado y la comparación con otros sistemas se hizo más difícil.

SM fue desarrollado por Montoyo y Palomar (2000) y se inspira en la medida de distancia semántica ***densidad conceptual*** (Agirre y Rigau, 1995, 1996). SM utiliza WN para, mediante sus relaciones de hiperonimia, encontrar subárboles dentro de esa jerarquía. Esos subárboles contienen todas o algunas de las palabras de un contexto que, en uno y sólo uno de sus sentidos, están dentro de ella.

La marca de especifidad es, por tanto, un concepto de WN que tiene como hipónimos algunos de los sentidos posibles en el contexto. La marca con más sentidos define la desambiguación de todas o casi todas las palabras del contexto. Dado que este sistema no es suficiente para algunos contextos, y buscando una mayor cobertura, el sistema se ha ido enriqueciendo con diversas heurísticas (Montoyo y Palomar, 2001) utilizando gran parte de la información que proporciona WN, incluidos los *WordNet Domains*.

SM sólo trabaja bien (siendo no supervisado) con nombres y tampoco tuvimos tiempo de establecer una estrategia de colaboración con suficientes garantías de éxito.

Por otro lado, se hizo un preproceso mínimo de los corpus de entrenamiento y evaluación, apenas una análisis sintáctico parcial con *Tree-tagger* para inglés y *Connexor* para español. Estando los dos módulos en sus estados de desarrollo primarios, nuestra participación fue útil como experiencia para el próximo SENSEVAL-3.

2.6 Apuntes sobre WSD

Un buen punto de partida para comprender el estado actual de la WSD es el artículo de Ide y Véronis (1998), donde se detallan aplicaciones de la tarea, corpus, diccionarios electrónicos, métodos, etc. Particularmente interesante son los comentarios y citas de discusiones acerca del tamaño necesario de los contextos y del tipo de información a usar (aún cuando se encuentra en un apartado dedicado a la relación entre WSD y traducción automática).

Respecto al tamaño del contexto parece que los mismos resultados se obtienen con el contexto local (unas pocas palabras a izquierda y derecha) que con frases o párrafos. De la información útil para WSD recoge la opinión de que información gramatical más compleja (roles, dependencias, etc.) ayudan también a la desambiguación. Este último hecho es refrendado por Martínez et al. (2002) y Agirre y Martínez (2001) que ofrecen una visión muy completa de la complejidad de la tarea.

Basándose en sus propios corpus para tres palabras, Leacock et al. (1998) llegan a la conclusión de que la información local es adecuada para el adjetivo *hard* y el verbo *serve*, mientras que el nombre *line* obtiene mejores resultados con información genérica de tópico o dominio (de contexto más amplio). Preiss (2001) y Mihalcea (2002), también, apuntan a la necesidad de diferenciar el tipo de información según qué palabras o dominio estemos tratando. Del mismo modo, Hoste et al. (2002, 2001) proponen el concepto de "experto", un clasificador que previamente ha seleccionado los mejores tipos de información relevantes para cada palabra, y llegan a la conclusión de que SemCor es insuficiente como conjunto de entrenamiento para una tarea tipo *all words*, tanto por no cubrir todas las posibles palabras a desambiguar como por la escasez de ejemplos para algunas de ellas.

Lee y Ng (2002) exploran diversas fuentes de información (de forma no tan completa como Agirre y Martínez (2001)) y métodos supervisados, pero estimando que las fuentes son dependientes entre si y que no existe "la mejor". También defienden que SVM es mejor método comparado con *naive* Bayes, AdaBoost y árboles de decisión. Otros trabajos relevantes de comparación de métodos son los de Mooney (1996) y Escudero et al. (2000a,b).

Florian y Yarowsky (2002) también utilizan información sintáctica y proponen diversas formas de combinar las salidas de varios clasificadores: votación, promediando las probabilidades y por preferencia de las clases más probables según la mayoría de clasificadores. También van Halteren et al. (2001) exploran distintas estrategias de combinación de métodos (modelos ocultos de markov, basado en ejemplos, reglas de transformación y máxima entropía) por votación y metaclasificadores, además de estudiar el impacto de distintas fuentes de textos anotados y no anotados (*Lancaster-Oslo/Bergen corpus* y *Wall Street Journal*), llegando a la conclusión de que cualquier combinación supera en precisión a un único sistema.

En general, hay una gran preocupación por saber cuáles son las fuentes de información "rentables" para WSD, así como la forma de combinar diversos métodos y clasificadores. Un ejemplo claro es el conjunto de sistemas presentados al último SENSEVAL-2.

No quisiéramos dejar de mencionar algunos trabajos relacionados directamente con WSD que se están desarrollando aquí, en España, aparte de los que participaron en el SENSEVAL-2 y que ya han sido expuestos. Concretamente nos referimos al sistema de la Universidad Politécnica de Valencia, basado en los modelos ocultos de Markov (Markov, 1913; Baum, 1972) que se muestra en la Tesis Doctoral de Molina (2004). Su contribución, entre otras, es un sistema de WSD supervisado basado en los modelos de Markov extendidos, introducidos éstos por Pla (2000), que necesita un corpus anotado complemente con sentidos (como es Semcor). Según su propia evaluación sobre los datos de la tarea SENSEVAL-2 *English All-words*, se trata de un sistema competitivo como esperan confirmar en el próximo SENSEVAL-3.

La Universidad de Jaén (García-Vega et al., 2003) está desarrollando su propio sistema supervisado a partir de redes neuronales artificiales y, más concretamente, aprendizaje por cuantificación vec-

torial (LVQ). Aunque las redes neuronales se han aplicado con éxito en multitud de campos, el interés dentro de la comunidad del PLN ha sido relativamente reciente, más aún para WSD (Cottrell y Small, 1983; Veronis y Ide, 1990).

Estas aproximaciones, más un sistema no supervisado basado en la detección de patrones (Nica et al., 2004b,a), se integrarán en un sistema combinado con nuestros métodos con la intención de participar en el próximo SENSEVAL-3.

2.6.1 Adquisición automática de corpus

Tras el SENSEVAL-2, nadie niega el éxito de los métodos supervisados para WSD, pero sólo si los comparamos con los no supervisados, ya que todavía no se ha conseguido el sistema fiable y robusto que se desea. Por otro lado, la ventaja de los supervisados sobre los no supervisados es algo relativa dada la variedad de métodos y de situaciones de evaluación.

Lo cierto es que la esperanza de los métodos y sistemas supervisados está en conseguir el número suficiente de ejemplos para entrenar (Ng, 1997). A esto hay que añadir los problemas derivados de los distintos orígenes de los datos almacenados en esos corpus. Por ejemplo, el corpus DSO se nutre de ejemplos extraídos del *Wall Street Journal* y del *Brown Corpus*, y los resultados que se obtienen aprendiendo con unos y evaluando sobre los otros indican que el dominio de los ejemplos de entrenamiento es un factor importante a la hora de conseguir precisiones aceptables (Escudero et al., 2000c). Conclusiones similares se extrajeron del estudio sobre Semcor de Agirre y Martínez (2000).

Así, uno de los grandes desafíos actuales es la confección de corpus de calidad, con una mínima cantidad de ejemplos por sentido y correctamente anotados, y en la adquisición automática de los mismos.

Según Gonzalo et al. (2003) las causas de la no utilización de WSD en tareas finales (traducción automática, recuperación de información, búsqueda de respuestas, etc.) son:

- WSD es más difícil que la propia tarea a la que quiere ayudar.

- diferentes tareas demandan diferente grado de distinción entre sentidos
- los sistemas no supervisados aún obtienen un pobre rendimiento
- y los supervisados apenas tienen recursos con los que trabajar, si acaso en inglés.

Como otros autores, defienden el uso de internet como fuente de la que extraer conocimiento lingüístico. El artículo es un resumen de los intentos por usar la web para ayudar a WSD.

Agirre y Martínez (2000) usan internet para enriquecer los sentidos de WN con información de dominio o categoría. Las *topic signatures* consisten en listas de palabras que se relacionan frecuentemente con cada sentido junto con una medida de cuan fuerte es esa asociación. Para ello utilizan los contextos resultado de consultas al buscador *AltaVista*.

Santamaría et al. (2003)[11] utilizan los directorios de los buscadores de internet (*Yahoo*) como etiquetas de categoría que asociar a los sentidos de WordNet. Defienden esta estrategia basándose en que los directorios de estos buscadores están hechos a mano, lo que puede asegurar su calidad y aprovechamiento.

El uso de corpus paralelos en distintas lenguas puede ayudar a la desambiguación semántica ya que si una palabra tiene una traducción muy concreta a otro idioma esto puede restringir la cantidad de sentidos a explorar (Gale et al., 1992a; Brown et al., 1991; Dagan et al., 1994). Nuevamente el problema es la disponibilidad de tales corpus (Ng et al., 2003), además de las propias limitaciones de la traducción automática (Diab y Resnik, 2002).

En cuanto a la adquisición automática de corpus anotados semánticamente, Leacock et al. (1998) utilizan los términos monosémicos relacionados con una palabra para identificar su sentido (un ejemplo sería 'semifinal' como referencia a 'partido' en su sentido deportivo, lo que descartaría el otro sentido relacionado con 'partido político'). Mihalcea y Moldovan (1999) y Mihalcea (2003) utilizan esta idea para realizar consultas a los buscadores de internet y explotan de forma más completa las relaciones de WN.

11 `nlp.uned.es/ODP`

Sin embargo, Agirre y Martínez (2000) intentaron aplicar los muy buenos resultados obtenidos por los anteriores a la desambiguación de SemCor y encontraron que, aun siendo correctos los ejemplos obtenidos automáticamente, las características de éstos no eran apropiadas para el corpus en cuestión. Entramos aquí en otra discusión, si lo que importa es la cantidad o la calidad.

El aprendizaje activo (***active learning***) consiste en seleccionar los ejemplos más informativos para el aprendizaje intentando reducir el coste de adquisición (Rigau et al., 2002; Mihalcea y Moldovan, 2001c). La calidad como propiedad de un ejemplo puede medirse según varios criterios o métodos (Argamon-Endelson y Dagan, 1999; Lewis y Gale, 1994; Fujii et al., 1998; Cohn et al., 1994).

Steedman et al. (2003) combinan dos aproximaciones a la adquisición automática de ejemplos, la selección de ejemplos (***sample selection***) y el coentrenamiento (***co-training***) para analizadores estadísticos. La selección de ejemplos (Thompson et al., 1999; Hwa, 2000; Tang et al., 2002) es una variante del aprendizaje activo y consiste en la búsqueda de ejemplos no anotados con una alta utilidad de aprendizaje (que más probablemente mejorarán el clasificador). El coentrenamiento, un algoritmo iterativo iniciado en una semilla de ejemplos anotados, busca los ejemplos más fiablemente anotados.

En general, se denominan algoritmos de semilla (***bootstrapping algorithms***) aquellos que intentan reducir el coste de adquisición de conocimiento presentando una colección inicial reducida de ejemplos anotados que, generalmente de forma iterativa, van viendo incrementada su cantidad por el propio proceso de desambiguación.

El coentrenamiento (Blum y Mitchell, 1998) es uno de los trabajos más citados entre los que buscan combinar información anotada y no anotada para la desambiguación. Ha sido aplicado a varias tareas como la clasificación de textos, analizadores parciales y completos (Steedman et al., 2003; Sarkar, 2001), y la adquisición de lexicones (Philips y Riloff, 2002). El coentrenamiento ha tenido varias revisiones y extensiones (Abney, 2002; Nigam y Ghani, 2000a,b; Pierce y Cardie, 2001). Aunque el trabajo de Yarowsky (1995), según Blum y Mitchell (1998), es una versión de coentrenamiento, otros autores defienden lo contrario dado que el coentrenamiento y lo que se denominan "clasificadores iterativos' se basan en restricciones diferentes (Abney, 2002; Mihalcea, 2003).

Si la mayor parte de estas aproximaciones iterativas buscan un conjunto de ejemplos correctamente anotados, Diab (2003) defiende que es posible un sistema de *bootstrapping* para WSD utilizando datos de menor calidad. SALAAM (*Sense Assignment Leveraging Alignments and Multilinguality*) es un sistema no supervisado de anotación semántica que se basa en el uso de corpus paralelos en varias lenguas obtenidos por traducción automática. La adquisición de nuevos ejemplos anotados mediante SALAAM y su uso en un método supervisado de aprendizaje automático se muestra como prometedor aunque la evaluación (con datos del SENSEVAL-2 para la muestra léxica en inglés) parece indicar que es bastante sensible a la parametrización del sistema y con comportamientos dispares según de la palabra que se trate.

El proyecto Meaning[12] (Rigau et al., 2002) tiene como objetivo avanzar hacia la "comprensión del lenguaje natural" (como un paso más allá del "procesamiento del lenguaje natural"). Entre sus objetivos principales destaca la búsqueda del método o métodos para obtener, de forma automática, corpus de calidad para WSD.

La idea es simple en el fondo: ya tenemos un buen puñado de técnicas y recursos que por si solos no llegan a lo realmente aceptable pero que es muy posible que interactuando entre ellos sí puedan conseguir el objetivo. Métodos basados en el conocimiento combinados con métodos supervisados ayudados, si cabe, por alineamientos entre distintas lenguas (y, en definitiva, ayudados por todo lo que sea humana y técnicamente posible), todo ello en un proceso cíclico en el que los resultados de unos sean la entrada de los otros.

2.7 Conclusiones

La conclusión principal, y a la espera del inminente SENSEVAL-3, es que falta mucho por hacer.

De las distintas aproximaciones al problema abordadas hasta ahora, los métodos supervisados son los que mejores resultados están obteniendo, si los comparamos con los no supervisados. No obstante, los niveles de precisión alcanzados en WSD (veánse los resultados de SENSEVAL-2) están lejos aún de lo que se puede considerar

12 www.lsi.upc.es/~rigau/meaning/documentation/

aceptable si los comparamos con los de otras tareas como el *POS-tagging* o el *base chunking*. Esto es particularmente importante ya que WSD no es una tarea aislada sino que debe aportar información adicional a las aplicaciones de PLN.

De entre los métodos supervisados no se puede destacar a ninguno en especial. Es posible que el problema radique más en los datos que se manejan, los conjuntos de entrenamiento y evaluación, y los sentidos definidos definidos para las palabras, que en los propios métodos.

WSD es una tarea difícil, incluso más que algunas a las que pretende ayudar, y la tendencia general es intentar la combinación de múltiples aproximaciones, métodos, heurísticas, etc. WSD precisa (o es ayudado en gran manera) de una gran cantidad de preproceso: análisis sintáctico, entidades, colocaciones, información de dominio, anáfora, patrones semánticos, etc. Al mismo tiempo, la selección de los atributos y de los ejemplos útiles para el aprendizaje, la influencia del dominio de los conjuntos de entrenamiento y clasificación, la diferenciación del proceso para cada palabra y sentido, son hechos aún por explotar y aprovechar plenamente.

Además, la tarea ya es dificultosa para los humanos por lo que la disponibilidad de conjuntos de ejemplos anotados es insuficiente. El siguiente paso, en el que ahora nos encontramos, es la disminución del esfuerzo anotador mediante técnicas semisupervisadas o de aprendizaje incremental, tanto para enriquecer los recursos ya existentes como para generar automáticamente nuevos corpus anotados.

Todo ello como apoyo a otras tareas como pueda ser la recuperación de información, la traducción automática, etc. Este apoyo implica, y es otro de los desafíos de WSD, rapidez de respuesta porque si no el papel de la resolución del sentido de las palabras seguirá siendo cuestionada para las tareas mencionadas.

No obstante, y en nuestra defensa, a veces deberíamos cuestionarnos el propio SENSEVAL como medida real del estado tecnológico de WSD (pero tampoco hay muchas más). Por poner un ejemplo de nuestro propio grupo de investigación, existe desconfianza en los sistemas de WSD actuales por parte de algunos desarrolladores de sistemas de recuperación de información. Se basan en los buenos resultados obtenidos sin utilizar desambiguación semántica en sus

propios foros de encuentro, pero tampoco han evaluado suficientemente si el uso que han hecho alguna vez de información semántica ha sido el correcto, tal vez son ellos los que no han sido capaces de aprovechar WSD. Se podría decir que ciertos colegas podrían (sólo podrían) morir de éxito y no ver más allá. Alguien puede argumentar que ciertos buscadores de internet tienen un funcionamiento sobresaliente, pero ya les va pesando la creciente cantidad de información en la red, y muchas veces nos ahogamos en el tamaño de la respuesta a una consulta.

Por contra, Baeza-Yates (2004) habla de la necesidad de incorporar el PLN para conseguir sistemas de recuperación de información de alta calidad, aunque no a cualquier precio (nivel de precisión, eficiencia, ...). Baeza-Yates comenta, cuando describe la ayuda necesaria para lo que se comienza a conocer como *web semántica*, que dos de los problemas de la información semántica son la falta de acuerdo en lo que debe ser un estándar en cuanto a como describirla y la calidad o grado de confianza en una determinada fuente. Concretamente, apunta a que WSD debe cumplir un papel fundamental en este tipo de sistemas.

En un futuro, WSD puede dejar de ser una tarea aislada, e incorporarse a los sistemas de adquisición de conocimiento de forma natural, en un único análisis que será sintáctico y semántico a la vez, siendo responsabilidad de cada uno el aprovechar convenientemente esa información, tal y como pasa actualmente con los analizadores sintácticos y otra herramientas similares.

En definitiva, e inevitablemente, WSD y PLN han de ayudar en tareas básicas de gestión de información, pero que

3

Modelos de probabilidad de máxima entropía

Como se ha explicado anteriormente, métodos de clasificación aplicados al PLN hay muchos. En nuestro caso hemos utilizado los *modelos de probabilidad de máxima entropía*, o modelos de máxima entropía (MME), que definen funciones de clasificación a partir de un conjunto de ejemplos previamente etiquetados. Es, por tanto, un método de aprendizaje supervisado basado en ejemplos.

Uno de los primeros trabajos que utilizan los MME en tareas de PLN, concretamente en reconocimiento del habla (*speech recognition*) es de (Lau et al., 1993; Lau, 1994). En una revisión histórica interesante, (Berger et al., 1996) citan a Laplace considerándolo el padre de la máxima entropía al haber enunciado su "principio de la razón insuficiente" hace más de dos siglos: «*cuando no tenemos información para distinguir entre la probabilidad de dos eventos, la mejor estrategia es considerarlos a los dos equiprobables*». También sostienen que el principio de máxima entropía se puede encontrar incluso en la Biblia y en los escritos de Herodoto. Finalmente, (Jaynes, 1990) dice:

> «*... el hecho de que cierta distribución de probabilidad maximice la entropía sujeta a ciertas restricciones que representan nuestra información incompleta, es la propiedad fundamental que justifica el uso de tal distribución para la inferencia; está de acuerdo con todo aquello que es conocido pero evita cuidadosamente asumir nada que sea desconocido ...*»

En realidad, el modelado basado en el principio de ME es un método general que se puede utilizar en cualquier tarea que se aborde desde la perspectiva del aprendizaje estadístico automático. Los ejemplos se representan de forma codificada mediante funciones que

se activan por la presencia de un determinada propiedad (se explicará más detalladamente en este capítulo; se dice de éste, y de todos los que utilizan esta técnica, que es un método basado en la representación por rasgos o atributos). Los modelos de ME proporcionan un entorno para la integración de, prácticamente, cualquier tipo de información de una forma sencilla. De hecho, la principal preocupación del usuario de esta técnica consiste en la detección de cuáles son las fuentes de información útiles para el problema que pretende resolver; es el método el que se encarga de componer la función de clasificación a partir de la muestra suministrada.

Las definiciones del método y del algoritmo aprendizaje que se muestran en este trabajo están extraídas del trabajo de Tesis Doctoral de Ratnaparkhi (1998), donde se puede encontrar una descripción más completa. En la sección 3.7, se añade un conjunto de referencias comentadas de otros trabajos y mejoras del método.

Ratnaparkhi, en ese mismo trabajo, emplea con éxito ME en varias tareas del PLN: detección de los límites de la frase, etiquetado gramatical (*part-of-speech tagging*), análisis sintáctico (*parsing*), dependencia de los sintagmas preposicionales (*prepositional phrase attachment*) y clasificación de textos.

Así mismo, los MME han sido utilizados, entre otros, en reconocimiento de entidades (Borthwick et al., 1998b,a), traducción automática estadística (Och y Ney, 2002; García-Varea et al., 2001) clasificación de textos (Nigam et al., 1999), extracción de patrones semánticos (Saiz-Noeda et al., 2001), sistemas de búsqueda de respuestas (Ittycheriah et al., 2001, 2000). Como extensión de los modelos de markov, éstos se han combinado con los MME dando lugar a un nuevo modelo general, los modelos de markov de máxima entropía (McCallum et al., 2000), aplicados en este caso a la extracción de información, concretamente a la segmentación de listas de "preguntas más frecuentes" en preguntas y respuestas. Desarrollos posteriores de este modelo pueden encontrarse en (Lafferty et al., 2001), y en (Sutton et al., 2004) donde es usado en tareas de POS-tagging, y detección de sintagmas nominales.

Se puede encontrar también un trabajo sobre WSD (Chao y Dyer, 2002) que coincidió en la misma publicación que nuestro artículo, (Suárez y Palomar, 2002b), en el que se implementan los MME para WSD, pero del que no tenemos noticias de desarrollos posteriores.

También se utilizan en van Halteren et al. (2001) en sistemas de combinación de varios métodos.

A continuación se va a describir el método general para, en el siguiente capítulo, adaptarlo a WSD.

3.1 Representación de la información

Estamos hablando de un método de aprendizaje supervisado que puede utilizar cualquier tipo de información, sólo falta saber cómo suministrarle tal información. Parte de la potencia del método basado en ME se fundamenta en la codificación previa de los datos que se van a usar para construir el clasificador. Esta codificación se realizará con el uso de funciones definidas para la tarea concreta a desarrollar.

El método no necesariamente ha de utilizar una codificación binaria pero, como se justificará más adelante por el uso de un determinado algoritmo de optimización, es la que hemos elegido. En realidad, las conocidas como *features*[1], o como las vamos a denominar a partir de este momento, **atributos**, son funciones que devuelven valores $verdadero$ o $falso$ para indicar la presencia, o no, de un determinado atributo. De hecho, a partir de ahora, emplearemos el término *atributo* para nominar este tipo de funciones.

Por ejemplo, supongamos que queremos obtener un clasificador que nos diga si un estudiante abandonará sus estudios superiores. Podemos caracterizar a los estudiantes por una serie de propiedades que poseerán o no: ha repetido curso en su etapa de enseñanza media, le compraron motocicleta a los dieciséis años y, por último, ha elegido una ingeniería al entrar en la universidad. Obviamente, podíamos haber elegido cualquier otro atributo.

Supongamos que nuestra muestra de ejemplos consiste en tres alumnos que tienen las siguientes características:

<"repitió","moto",–><ABANDONO>
<–,"moto",–><ABANDONO>
<–,–,"ingeniería"><NO ABANDONO>

1 Ratnaparkhi también utiliza el término "predicado contextual" (*contextual predicate*).

La interpretación de los datos anteriores, si observamos el caso del $alumno_1$, es que un alumno repitió curso en su período de enseñanza media, le compraron la motocicleta pero no ha elegido una ingeniería como sus estudios superiores y, además, abandonó la carrera elegida antes de terminarla.

Estos datos inducirán los atributos a utilizar en el aprendizaje. Es importante que nos demos cuenta de que los atributos siempre se refieren a una clasificación concreta. Los alumnos son representados por vectores de valores de verdad, cada uno calculado a partir de una función, o *atributo*, que tiene la forma de la ecuación 3.1.

$$f(x,c) = \begin{cases} 1 \text{ si } c = c' \text{ y } cp(x) = cierto \\ 0 \text{ en otro caso} \end{cases} \tag{3.1}$$

En la ecuación 3.1 la condición $c' = c$ es la que liga el atributo a una clase concreta, y cp es un predicado cualquiera. Cada instancia del objeto a clasificar se sustituirá en x, y la clase a la que pertenece en C. Por ejemplo:

$$f_1(x,c) = \begin{cases} 1 \text{ si } c = ABANDONO \text{ y repitió(x)= } cierto \\ 0 \text{ en otro caso} \end{cases}$$

$$f_2(x,c) = \begin{cases} 1 \text{ si } c = ABANDONO \text{ y } moto(x) = cierto \\ 0 \text{ en otro caso} \end{cases}$$

$$f_3(x,c) = \begin{cases} 1 \text{ si } c = NO\ ABANDONO \text{ e ingeniería}(x) = cierto \\ 0 \text{ en otro caso} \end{cases}$$

Si obtenemos los valores de estas funciones para el $alumno_1$:

$$f_1(alumno_1, ABANDONO) = 1$$
$$f_2(alumno_1, ABANDONO) = 1$$
$$f_3(alumno_1, ABANDONO) = 0$$

Así, la representación completa del conjunto de ejemplos sería la siguiente:

$Alumno_1$: 110
$Alumno_2$: 010
$Alumno_3$: 001

Pero volvamos a la discusión sobre las "propiedades" de cada individuo. Es necesario precisar más: denominaremos *contexto* al conjunto de información que consideramos útil para caracterizar a un determinado objeto. En el ejemplo que estamos manejando el contexto que para nosotros es el $alumno_1$ sería <"repitió curso","tuvo moto","no eligió ingeniería">. Resumiendo, el contexto es el conjunto de propiedades que tiene un individuo, tal y como las entendemos habitualmente, y un atributo es una de estas propiedades donde, además, se verifica su pertenencia a una clase o tipo de objeto.

<"repitió","moto",–><ABANDONÓ>:110
<–,"moto",–><ABANDONÓ>:010
<–,–,"ingeniería"><NO ABANDONÓ>:001

3.2 Aprendizaje (o entrenamiento) y clasificación

Conocemos "cómo son" nuestros tres alumnos. Supongamos que, a partir de esta información, somos capaces de obtener una función que nos informe de la probabilidad de que un individuo x pertenezca a una determinada clase c, $p(c|x)$; sería inmediato definir un clasificador, $cl(x)$, que nos elija la clase con mayor valor de probabilidad, o lo que es lo mismo, conociendo la "historia" de cualquier alumno podríamos predecir si abandonará o no la carrera.

$$cl(x) = \arg\max_c p(c|x)$$

$p(c|x)$ = probabilidad de que el alumno x pertenezca a la clase c

La tarea se completa si definimos el modelo de probabilidad por el que vamos a obtener los valores de $p(c|x)$. Lo que hemos llamado aprendizaje es la obtención de dicho modelo, que debe cumplir una serie de restricciones para acomodarse al principio de máxima entropía.

El aprendizaje consiste en coger todos y cada uno de los atributos definidos y asignarles un peso de acuerdo a los datos de ejemplo, en nuestro ejemplo los alumnos conocidos. La función de probabilidad se define con esos pesos y atributos de tal forma que, ante un alumno desconocido, podemos calcular las probabilidades de que pertenezca a cada clase y seleccionar aquella que obtiene el máximo valor.

El aprendizaje es una aplicación que a cada atributo $f_j(x, c)$ le asocia un coeficiente α_j (más adelante se explica la forma de estimar estos coeficientes). La clasificación, para cada nuevo individuo, calcula sus atributos para todas las clases posibles y obtiene los valores de probabilidad correspondientes. Por ejemplo, si de un nuevo alumno sabemos que le compraron moto, si el peso del atributo que liga esta información a la clase "abandonará" es mayor que el que la asocia a la otra clase posible, la predicción será que no terminará los estudios superiores que ha elegido.

Al estimar los parámetros asociados a cada atributo estamos dando más importancia a unos y quitándosela a otros. Es habitual que los métodos basados en atributos utilicen una fase previa al aprendizaje en la que determinan cuáles son las apropiadas para aprender sin introducir "ruido" (atributos irrelevantes, redundantes, erróneos, ...). Suele ser el caso de atributos que se activan muy pocas veces entre todos los ejemplos de aprendizaje (por desgracia, WSD adolece de abundancia de este tipo de atributos).

Finalmente, el problema a solucionar, cómo utilizar toda esta información resumida en forma de función de clasificación para etiquetar a un nuevo alumno. Supongamos que, de él, conocemos que repitió curso, que no le compraron motocicleta alguna y que ha elegido una ingeniería, esto es:

$Alumno_x$:<"repitió",–,"ingeniería"><?>:?

El proceso de clasificación consiste en comprobar el valor de $p(c|x)$ para todas las clases:

$$p(ABANDONO|A_x) = p(100) = a$$
$$p(NO\ ABANDONO|A_x) = p(001) = b$$

Si $a > b$ el alumno A_x será clasificado como que "abandonará" la carrera antes de finalizarla, y viceversa.

Obviamente, el ejemplo utilizado hasta ahora es necesariamente breve en el número de atributos elegidos y, muy posiblemente, deficiente en calidad y claridad. Para formalizar los objetivos y el método de los MME, las siguientes secciones desarrollan los fundamentos y el procedimiento de aprendizaje.

3.3 Modelos de máxima entropía condicional

En esta sección vamos a describir detalladamente los fundamentos de los modelos de máxima entropía condicional. La base de este método exponencial es que no hay nada más allá de los propios datos de entrenamiento (representado en la ecuación 3.2, que se muestra más adelante). Trabajar con un modelo de probabilidad de máxima entropía es asumir la máxima incertidumbre: «*si escogemos un modelo con menor entropía, estaremos añadiéndole restricciones que no están justificadas por la evidencia empírica*» (Manning y Schütze, 1999).

Según la descripción de Ratnaparkhi (1998), partimos de un conjunto de clases C, un conjunto de contextos X y un conjunto de N ejemplos de aprendizaje, y se han definido K atributos (es decir, K funciones del tipo de la ecuación 3.1) para llegar a obtener un modelo de probabilidad óptimo p^*.

El conjunto de todos los modelos de probabilidad posibles, P (ecuación 3.2), está formado por todos aquellos que cumplen que las frecuencias esperadas de los atributos, $E_p f_j$, son iguales a las observadas, $E_{\tilde{p}} f_j$. Es decir, las frecuencias de activación de cada atributo en estos modelos de probabilidad son las mismas que en el conjunto de entrenamiento.

$$P = \{p|E_p f_j = E_{\tilde{p}} f_j, j = \{1..K\}\} \tag{3.2}$$

El modelo de probabilidad óptimo, p^* (ecuación 3.3) es el de mayor entropía condicional promediada en el conjunto de aprendizaje, $H(p)$.

$$p^* = \arg\max_{p \in P} H(p) \tag{3.3}$$

La entropía condicional, $H(p)$ (ecuación 3.4), se calcula en base a las frecuencias en la muestra de cada contexto x, $\tilde{p}(x)$, y de la probabilidad condicional de cada clase en esos contextos $p(c|x)$.

$$H(p) = -\sum_{c,x} \tilde{p}(x)p(c|x)\log p(c|x) \tag{3.4}$$

$E_{\tilde{p}}f_j$ (ecuación 3.5) es la frecuencia observada en el conjunto de entrenamiento del atributo f_j, donde $\tilde{p}(c,x)$ es la frecuencia de un par (c,x) dentro de la muestra, siendo $c \in C$ y $x \in X$. Nótese que al utilizar atributos binarios la expresión se convierte en una frecuencia estadística; en el caso de atributos enteros o reales, tal correspondencia no existe.

$$E_{\tilde{p}}f_j = \sum_{c,x} \tilde{p}(c,x)f_j(c,x) = \frac{1}{N}\sum_{i=1}^{N} f_j(c_i,x_i) \tag{3.5}$$

Por contra, $E_p f_j$ (ecuación 3.6) es la frecuencia esperada del atributo dada por el modelo de probabilidad, y se calcula a partir del espacio completo de posibles contextos y clases, no únicamente de los observados[2] (aunque a efectos prácticos, como veremos, sólo vamos a utilizar los observados).

$$E_p f_j = \sum_{c,x} \tilde{p}(x)p(c|x)f_j(c,x) \tag{3.6}$$

Finalmente, la probabilidad condicional, en el modelo, de una clase c dado un contexto x (ecuación 3.7) es el valor normalizado del productorio de los coeficientes estimados α_j de aquellos atributos que se activan en el contexto x. El valor $Z(x)$ (ecuación 3.8) se utiliza para asegurar que la suma de las probabilidades de todas las clases condicionadas a x es igual a 1.

$$p(c|x) = \frac{1}{Z(x)}\prod_{j=1}^{K} \alpha_j^{f_j(x,c)} \tag{3.7}$$

2 En un corpus no nos vamos a encontrar, normalmente, $|X| \times |C|$ ejemplos; en otras palabras, determinados contextos se darán con algunas, no con todas las clases.

$$Z(x) = \sum_{c} \prod_{j=1}^{K} \alpha_j^{f_j(x,c)} \tag{3.8}$$

Por claridad, la figura 3.1 agrupa todas las restricciones del modelo, tal y como se han expuesto en esta sección.

$$P = \{p | E_p f_j = E_{\tilde{p}} f_j, j = \{1..K\}\}$$
$$p^* = \arg\max_{p \in P} H(p)$$
$$H(p) = -\sum_{c,x} \tilde{p}(x) p(c|x) \log p(c|x)$$
$$E_{\tilde{p}} f_j = \sum_{c,x} \tilde{p}(c,x) f_j(c,x) = \frac{1}{N} \sum_{i=1}^{N} f_j(c_i, x_i)$$
$$E_p f_j = \sum_{c,x} \tilde{p}(x) p(c|x) f_j(c,x)$$
$$p(c|x) = \frac{1}{Z(x)} \prod_{j=1}^{K} \alpha_j^{f_j(x,c)}$$
$$Z(x) = \sum_{c} \prod_{j=1}^{K} \alpha_j^{f_j(x,c)}$$
$$K = \textit{cantidad de atributos}$$
$$N = \textit{cantidad de ejemplos de aprendizaje}$$

Figura 3.1. Definición de los MME

3.4 Algoritmo de aprendizaje

Una vez que se ha definido el modelo de probabilidad a obtener, se presenta un algoritmo estándar para la estimación de los coeficientes α_j, lo que en la práctica significa el propio procedimiento de aprendizaje por el que llegaremos al modelo óptimo, p^*.

El procedimiento utilizado es el denominado *Generalized Iterative Scaling* (Darroch y Ratcliff, 1972), o GIS. Éste es un procedimiento

iterativo que necesita que la suma de los atributos para todos los contextos y clases sea igual a una constante, D (ecuación 3.9). Este valor es sencillo de calcular puesto que basta con hallar el máximo valor de la suma de los atributos de los contextos de aprendizaje[3]

$$D = \max_{x \in X, c \in C} \sum_{j=1}^{K} f_j(x, c) \tag{3.9}$$

Esto obliga, para corregir aquellos casos en los que la suma no alcanza este valor, a añadir un atributo de "ajuste", $f_l(x, c)$, a todos los contextos (ecuación 3.10).

$$f_l(x, c) = D - \sum_{j=1}^{K} f_j(x, c) \tag{3.10}$$

Nótese que éste no siempre será necesario ya que si todos los contextos suman la misma cantidad de atributos activos, el atributo de "ajuste" valdrá 0 en todos los casos; además, este atributo, al contrario que los que hemos definido nosotros, puede ser mayor que 1, dependiendo del contexto al que se aplica[4].

Se describe a continuación el algoritmo iterativo cuyo resultado final es p^*:

$$\begin{aligned} \alpha_j^{(0)} &= 1 \\ \alpha_j^{(n+1)} &= \alpha_j^{(n)} \left(\frac{E_{\tilde{p}} f_j}{E_{p^{(n)}} f_j}\right)^{\frac{1}{D}} \\ &donde \\ E_{p^{(n)}} f_j &= \sum_{c,x} \tilde{p}(x) p^{(n)}(c|x) f_j(c, x) \\ p^{(n)}(c|x) &= \frac{1}{Z(x)} \prod_{j=1}^{K} (\alpha_j^{(n)})^{f_j(x,c)} \end{aligned} \tag{3.11}$$

3 Ratnaparkhi advierte de que se está realizando una adaptación de la teoría puesto que la constante debería ser calculada sobre el espacio completo de $X \times C$, no únicamente sobre los ejemplos del conjunto de aprendizaje; no obstante, la constante calculada de este modo suele ser suficiente.

4 Otro algoritmo que será mencionado más adelante, el *Improved Iterative Scaling*, no utiliza tal ajuste.

La primera iteración asigna a todos los coeficientes el valor 1 (en realidad, vale cualquier valor de inicio). El procedimiento termina cuando se satisface el criterio de convergencia, cuando el modelo de probabilidad es el óptimo o está lo suficientemente cercano del óptimo. Este criterio, en su formulación más simple, es un valor de mínima variación entre los coeficientes de una iteración y la siguiente, o simplemente un número máximo de iteraciones.

En principio, se supone que el cómputo se ha de realizar sobre el espacio $X \times C$, es decir, para todos los posibles pares (x, c) estén o no en el conjunto de aprendizaje. Sin embargo, en la práctica, las iteraciones sólo van a tener en cuenta los ejemplos: ya hemos visto que el cálculo de $E_{\tilde{p}} f_j$ es sencillo (ecuación 3.5); además, en el caso de $E_{p(n)} f_j$, puesto que para contextos que no pertenecen al conjunto de aprendizaje la probabilidad $\tilde{p}(x)$ es igual a 0, éstos se pueden excluir de la suma. El coste de cada iteración, si suponemos que V es el promedio de atributos activos entre los contextos, es del orden de $O(N|C|V)$.

3.5 Características de los modelos de máxima entropía

Según Ratnaparkhi algunas características destacables de los MME son:

- Integración de fuentes de información heterogéneas
- Selección de atributos integrada en el aprendizaje
- Los atributos no necesitan tener una definición compleja

Ratnaparkhi arguye que frente a métodos como *naive* Bayes y *decomposable probability models*, los MME permiten atributos más flexibles (aunque a costa de la estimación de parámetros); es capaz de usar más información para cada predicción que las listas de decisión; el aprendizaje basado en transformaciones (*transformation-based*) sólo devuelve una clasificación sin probabilidad. Con el único método con el que no establece una clara ventaja es con los árboles de decisión, aunque circunscribe sus comentarios al análisis sintáctico y ciertas aproximaciones citadas por él.

3.6 Límites de los MME

Ratnaparkhi (1998) cita algunas de las limitaciones de esta aproximación a los MME.

Atributos binarios

El uso de atributos binarios no es adecuado para ciertos tipos de información[5], por ejemplo la captura de información en el ámbito del documento cuando la frecuencia de aparición de una palabra es más útil que su presencia o ausencia. La solución mostrada en el trabajo de Ratnaparkhi es simple, como el ejemplo mostrado en la ecuación 3.12, donde se informa de si una palabra w tiene una frecuencia dentro de un contexto x mayor de 10 para la clase c'.

$$f(x,c) = \begin{cases} 1 \text{ si } c = c' \text{ y } frecuencia(w,x) >= 10 \\ 0 \text{ en otro caso} \end{cases} \quad (3.12)$$

Obviamente, si los rangos a observar no están claros o si es preciso utilizar todos los valores del mismo, es posible que debamos plantear la necesidad de cambiar de procedimiento de optimización. Para la tarea que nos ocupa, WSD, aún cuando este tipo de atributos pueden ser definidos, no parece que vayan a ser muchos ni diferirán en demasía del esquema de la ecuación 3.12.

Problemas de convergencia

Es el caso de los contextos poco frecuentes (que incluso pueden ser no ambiguos, precisamente por esa baja frecuencia). Pensemos, por ejemplo, en la clasificación de textos basada en la inclusión de palabras. Supongamos que la palabra w_1 aparece una vez en todo el conjunto de aprendizaje en un contexto de clase c_1, y otra palabra, w_2, aparece muy frecuentemente en documentos de clase c_2 y muy poco en documentos c_1. Lo normal sería que la aparición de w_2 forzara la clasificación del documento como c_2 ya que disponemos de

5 Nuevamente debemos recordar que es el GIS el que necesita este tipo de atributos.

muchos más ejemplos en los que así ocurre. Sin embargo, el modelo de probabilidad clasificará aquellos documentos en los que estén las dos palabras a la vez como c_1.

Sean $f_{w_1c_1}$ y $f_{w_2c_2}$ los dos atributos referidos a las palabras y clases antes mencionadas. Dadas las restricciones del modelo, la frecuencia esperada de un determinado atributo ha de ser igual a la observada (véase ecuación 3.5). La frecuencia observada del primer atributo, $E_{\tilde{p}}f_{w_1c_1}$, será cercana a 0 mientras que la del segundo, $E_{\tilde{p}}f_{w_2c_2}$, se puede acercar a 1; el coeficiente asociado al primero (α_{11}) tenderá a infinito, pero el asociado al segundo (α_{22}) convergerá a un valor finito. En la práctica, esto significa que la activación de esos atributos poco frecuentes dominará la clasificación; el método confiará "infinitamente" más en los contextos poco frecuentes que en los muy frecuentes. La solución que propone Ratnaparkhi es eliminar aquellos atributos cuya activación se encuentra por debajo de un umbral (*feature cutoff*: él utiliza 5 y 10 como valores por debajo de los cuales un atributo se elimina del aprendizaje).

Sin embargo, para la tarea que pretendemos desarrollar, y por el tipo de atributos utilizado, esta posibilidad no nos preocupa en demasía, ya que el caso anterior es difícil que se dé. El ejemplo anterior presenta ese comportamiento "anómalo" cuando el conjunto de aprendizaje es muy grande y las frecuencias de los contextos son extremas. Para WSD, ninguno de los atributos parece que vaya a tener una frecuencia esperada tan cercana a 1 como para indicar la evidencia de la clase a la que está ligado. De hecho, técnicas suavizadoras como la propuesta podrían eliminar prácticamente casi toda la información de aprendizaje.

3.7 Otras lecturas

A continuación se recomiendan algunas lecturas complementarias relativas a los modelos de ME.

De la teoría de los MME

Anterior al trabajo de Ratnaparkhi son los de Lau et al. (1993) y Berger et al. (1996) que presentan el método aplicado al PLN. Otro

trabajo relacionado con los fundamentos de los MME y algunos ejemplos de posibles fuentes de información se puede encontrar en (Rosenfeld, 1996).

Rosenfeld (1997) presenta un modelo del lenguaje no condicional de ME basado en oraciones completas, no en palabras, que son tratadas como conjuntos de atributos, siendo éstos propiedades computables arbitrarias (por ejemplo, las veces que aparece un determinado trigrama dentro de una oración). Defienden que los modelos ME basados en sentencias son eficientes porque no necesitan normalización al tiempo que pueden tratar de forma natural fenómenos lingüísticos relacionados con las oraciones.

El hecho novedoso del trabajo de (Dehaspe, 1997) es que combina los MME y las técnicas de programación lógica inductiva. Entre otras cosas, los ejemplos están representados por programas Prolog sobre los que se evalúan cláusulas Prolog para calcular las restricciones del modelo.

Optimización

Una de las críticas más o menos constantes a los modelos de máxima entropía en general es su carga computacional precisamente en la optimización del modelo de probabilidad. Una variante es el algoritmo denominado *Improved Iterative Scaling* (IIS) (Berger et al., 1996) cuyos autores aducen que converge más rápidamente que el GIS (en menos iteraciones).

Hay que resaltar que, mientras GIS es un procedimiento que se basa en una regla cuyo cálculo es sencillo, IIS obliga a utilizar un método adicional de resolución de ecuaciones (Berger et al. (1996) utilizan el de Newton) en cada iteración para obtener el incremento de cada uno de los coeficientes. Malouf (2002) compara diversos métodos de optimización y muestra que, aún siendo cierta la mayor velocidad de convergencia del IIS, el cálculo más complejo de cada iteración diluye esa ventaja teórica. Sólo en ciertos casos, efectivamente IIS es más rápido que el GIS, a lo que hay que añadir que, por los resultados obtenidos por Malouf, el segundo consigue una tasa de acierto un poco mayor.

La conclusión más importante de este trabajo es que, aún siendo GIS e IIS los algoritmos más utilizados, el problema no deja de ser la optimización de una función para lo que existen otros métodos más adecuados. Así, propone la utilización de paquetes de software que implementan algoritmos de optimización[6]: *conjugate gradient* y *variable metric methods*. En particular, sus resultados parecen indicar que, para tareas de clasificación en PLN, *limited memory variable metric* (Benson y Moré, 2001) es significativamente mejor.

Uno de los aspectos que parecía dar ventaja a IIS frente a GIS es que el primero no necesita un atributo de ajuste o corrección. Curran y Clark (2003) defienden que, adaptando la prueba de Berger para la convergencia del IIS, se llega a la conclusión de que el GIS tampoco la necesita.

Se han publicado otras variantes del GIS y del IIS buscando una mayor rapidez de cálculo que se pueden encontrar, entre otras, en las publicaciones de Wu y Khudanpur (2000) y Goodman (2001, 2002).

De todas formas, el problema que estamos tratando aquí, la desambiguación del sentido de las palabras, y teniendo en cuenta los corpus disponibles, presenta muy poco margen para la comparación entre distintos métodos de optimización. Generalmente, se trabaja con muy pocas clases y la cantidad de ejemplos, salvo excepciones, no es lo suficientemente grande como para encontrar diferencias significativas. Sí es posible que un proceso de desambiguación, llamémosle real, que involucre a muchas palabras, por acumulación, sea significativamente sensible al método elegido.

Suavizado y selección de atributos

Aunque inicialmente se argumentaba que los MME no precisan de suavizado, ya que el propio proceso de optimización selecciona los atributos por su mayor peso frente a los demás, sí se ven afectados por atributos "raros", que se activan pocas veces. En realidad, como a todos los métodos estadísticos de clasificación, y más en PLN, el problema no es tanto el atributo en si, sino el tipo de información que puede no ser adecuado para la tarea.

6 Concretamente, *Toolkit for Advanced Optimization* (TAO) y *Portable, Extensible Toolkit for Scientific Computation* (PETSc), disponibles en `www-fp.mcs.anl.gov/tao/` y `www-unix.mcs.anl.gov/petsc/petsc-2/`, respectivamente.

El suavizado, como práctica posible en cualquier método de aprendizaje automático, quiere evitar el "sobreaprendizaje" (*overfitting*). Este fenómeno estadístico se produce cuando la naturaleza de los datos hace que los fenómenos lingüísticos representados por ellos sean muchos y con una frecuencia de aparición muy baja. Por ejemplo, la palabra tablas puede ir precedida por infinidad de otras palabras: "las tablas", "250 tablas", "en tablas", etc. No es lógico pensar que nuestro corpus va a contener todos y cada uno de los ejemplos posibles. En consecuencia, aumentar la cantidad de ejemplos de tablas no va a mejorar las predicciones sobre esta palabra. El efecto es que el número de coeficientes del modelo es excesivo y se ajustan tanto al conjunto de entrenamiento que las predicciones empeoran.

Como técnicas de suavizado se menciona como la más simple la eliminación de atributos que se activan menos de n veces (*cutoff*). Además, *Gaussian prior* (Chen y Rosenfeld, 2000, 1999), modelos de máxima entropía difusa (*fuzzy ME model*) (Lau, 1994), *fat restrictions* (Khudanpur, 1995; Newmann, 1977), y relajación de las restricciones de igualdad entre frecuencias esperadas y observadas (Kazama y Tsujii, 2003). Nuevamente en (Curran y Clark, 2003) se hace una comparación entre el *cutoff* y el *Gaussian prior*, aplicado en este caso al etiquetado de categorías léxicas (*POS-tagging*).

La selección de atributos persigue dos objetivos al mismo tiempo: aumentar el acierto en la clasificación y disminuir el tiempo de aprendizaje. Se puede ver como un preproceso anterior al propio entrenamiento cuyo propósito es, y valiéndose del corpus de aprendizaje, descartar la información que no aporta nada o que incluso provoca confusión en el aprendizaje. La detección de estas fuentes de información no deseadas se estudia en capítulos posteriores, donde se expondrán diversos análisis sobre este asunto, realizados por nosotros con nuestro sistema de WSD basado en ME, al tiempo que se comentarán algunos trabajos de otros investigadores.

Mikheev (1998) se basa en la idea de que un atributo complejo es mejor que dos simples que no son independientes, y generaliza y selecciona atributos a partir del conjunto de ejemplos mediante un algoritmo previo al IIS.

Zhou et al. (2003) presentan una revisión bastante completa de las técnicas de selección de atributos y proponen una mejora del al-

goritmo de selección de atributos incremental (IFS) (Berger et al., 1996).

3.8 Conclusiones

Los modelos de máxima entropía condicional son utilizados como método de aprendizaje supervisado del que se obtiene un modelo probabilístico y, por tanto, una función de clasificación. El aprendizaje es, en realidad, un proceso de optimización por el que el modelo de probabilidad obtenido es el que tiene la máxima entropía, o lo que es lo mismo, la máxima incertidumbre, ya que no se quiere asumir nada que no esté en los propios ejemplos de aprendizaje.

Una función de clasificación obtenida de esta forma incluye un conjunto de coeficientes o parámetros estimados por el procedimiento de optimización, cada uno asociado a un atributo concreto, de forma que el coeficiente determina el peso del atributo dentro de la función de clasificación.

Los ejemplos, la información que se considera indispensable, los contextos, y la clase a la que pertenecen, se codifican en forma de atributos binarios para procesarlos en el aprendizaje.

Como ventajas de los modelos de probabilidad de ME se pueden citar los buenos resultados obtenidos en ciertas tareas del PLN mediante la utilización de información relativamente simple, y que permite, «*virtualmente sin ninguna restricción, la representación del conocimiento especifico de un determinado problema en forma de atributos*» (Ratnaparkhi, 1998).

4

Sistema WSD-Máxima Entropía

Como se dijo al principio, nuestro objetivo es construir un sistema de WSD, o sea, que nuestro ordenador lea un texto y le asigne a cada palabra su sentido correspondiente. Elegimos los *modelos de máxima entropía condicional* como la herramienta con la que realizar tal asignación. El porqué éstos y no otros ya se ha expuesto en el capítulo anterior: Ratnaparkhi (1998) los aplicó, con éxito, a otras tareas del PLN, mientras que para WSD no teníamos conocimiento de su uso, al menos en forma de sistema completo y terminado. También se espera de ellos una mejor adecuación al problema de manejar palabras y propiedades sintácticas del texto, además de hacerlo de una forma relativamente simple.

El método por el que vamos a conseguir el modelo de probabilidad óptimo, el que maximiza la entropía, ha sido descrito en el capítulo anterior y muy pocas modificaciones o adaptaciones hay que hacer, si acaso la personal forma de abordar su programación en aras de una mayor eficiencia o flexibilidad. No obstante, la tarea más importante aún está por hacer: elegir la información con la que vamos a caracterizar cada uno de los ejemplos del corpus y cuyo proceso dará como resultado un clasificador.

Para dejar bien claro cuál va a ser el resultado final, la siguiente sección expone los objetivos de la implementación. A continuación se tratan los detalles que finalmente se han traducido en un programa de ordenador. Se hablará de la arquitectura general del sistema, del tratamiento de los corpus de aprendizaje y de evaluación y, como la parte fundamental de todo el proceso de desambiguación, la definición de los atributos o caracterización de los contextos.

4.1 Objetivos

El objetivo es obtener clasificadores (o más propiamente, funciones de clasificación). Para hacer la tarea más eficiente el proceso de aprendizaje se subdividirá por palabras para disponer de un clasificador por cada una. Aún se pueden especializar más los clasificadores si tenemos en cuenta las categorías sintácticas (nombres, verbos, adjetivos y adverbios). A la palabra que define las clases en las que se puede clasificar la denominaremos *palabra objetivo*.

La restricción de un clasificador por cada palabra viene dado por la propia naturaleza de la tarea. Es bien conocido que los métodos de aprendizaje automático que abordan el problema de la clasificación son muy sensibles a la cantidad de clases: cuantas más clases posibles, probablemente, más difícil será encontrar el modelo de probabilidad óptimo y peores las predicciones[1]. Por otro lado, y mencionando concretamente el método elegido por nosotros, dadas las características de los MME, cuanto más grande sea el conjunto de clases, la cantidad de funciones de atributo, y por tanto el número de coeficientes a estimar, aumenta espectacularmente y los tiempos de respuesta se incrementan igualmente.

El proceso de selección de atributos lo entendemos más como una *selección de fuentes de información* útiles, en contra de las subtareas habituales en otros modelos y métodos, por las cuales se eliminan aquellos atributos que influyen negativamente en el resultado final. Si un atributo es "compuesto está a la derecha de interés#4", la fuente de información a la que pertenece este atributo particular es "palabras a la derecha de interés#4".

4.2 Descripción general del sistema

El aquí propuesto es un método de aprendizaje supervisado a partir de un corpus de textos anotados semánticamente. Así pues, la tarea necesita una fase previa de aprendizaje antes de poder construir y almacenar un clasificador para cada palabra. En esta fase de

1 Evidentemente, no todas las tareas del PLN son iguales. Incluso dentro de WSD, no está tan claro que una palabra con un alto grado de polisemia redunde en un clasificador con resultados pobres.

aprendizaje se recogen los ejemplos del corpus y se incorporan al modelo de probabilidad para hacer la estimación de la función de clasificación.

El sistema tiene, por tanto, dos tareas fundamentales, aprender (módulos de extracción de contextos y estimación de parámetros) y desambiguar (módulo clasificador). Para ello se ha diseñado y desarrollado el conjunto de programas de ordenador que se describe a continuación[2].

Brevemente, puesto que se explicarán con detalle más adelante, el **extractor de contextos** es el módulo encargado de procesar los textos de entrada (adecuadamente etiquetados con la información necesaria) y confeccionar las cadenas de descripción de cada contexto. Esta descripción consistirá en la detección y codificación de la información relevante.

Otro de los módulos es el **estimador de parámetros**, que tiene como único propósito el cálculo de los coeficientes que dan peso a cada una de las funciones o características. El procedimiento implementado, como ya se ha dicho, es el *Generalized Iterative Scaling* (GIS). Su entrada es la lista de contextos procesados por el módulo extractor.

Finalmente, el **clasificador** es el que realiza la tarea de asignar sentidos a textos no anotados. Usa el módulo extractor para caracterizar los contextos y, posteriormente, calcula las probabilidades de que ese contexto se corresponda con cada una de las clases posibles.

Además, son necesarios otros programas como lematizadores, etiquetadores sintácticos o analizadores de la oración, que incorporan al texto plano original la información morfológica, sintáctica y semántica a procesar.

Pasamos ahora a describir más detalladamente el sistema

2 La programación se ha llevado a cabo enteramente en C++.

4.3 El sistema en detalle

4.3.1 Fuentes de información morfológicas, sintácticas y semánticas

La evidencia nos dice que WSD maneja o puede llegar a manejar muchos tipos diferentes de información útiles para resolver el sentido correcto de un contexto (McRoy, 1992; Ng y Lee, 1996). Los atributos pueden definirse usando únicamente el texto plano, pero nos tendríamos que limitar a las formas de las palabras en él escritas, y esto no suele bastar. De un analizador de oraciones podremos conseguir lemas, categorías gramaticales, información de número y persona, roles sintácticos, dependencias, palabras compuestas, etc. Habrá que ver qué datos son útiles y cuáles más que otros. En general, para este trabajo esta información se combinará mediante atributos enfocados hacia los siguientes datos:

Palabras: son las palabras que se encuentran más cerca de la palabra ambigua. En general, las más útiles para WSD se encuentran dentro de una ventana centrada en la ambigua de [-3,+3] posiciones relativas (Martínez y Agirre, 2000; Ng y Lee, 1996). Se utilizarán las propias palabras y sus lemas, y se podrá diferenciar entre si son palabras llenas (con contenido léxico; para ello necesitamos su categoría sintáctica) o no, y la influencia de ciertas composiciones de palabras en el aprendizaje de sentidos concretos.

Categorías gramaticales: en principio, saber si son nombres, verbos, adjetivos o adverbios, pero la propia etiqueta sintáctica, que suele suministrar más información, puede ser útil para definir atributos.

Análisis de la frase: hablamos de dependencias, roles sintácticos, unidades gramaticales, detección de palabras compuestas, etc. Por rol sintáctico entendemos, por ejemplo, si la palabra actúa de sujeto (o está dentro de un sintagma nominal que actúa como sujeto), si es un complemento, etc. Las dependencias se representan mediante un árbol cuya raíz es un núcleo verbal principal de

la frase, y en el que se van desarrollando los distintos sintagmas hasta llegar a las hojas terminales que ya son palabras. Que esta información sea más o menos rica depende de las capacidades del analizador sintáctico.

Información de dominio: aquí ya abandonamos las cercanías de la palabra objetivo y abarcamos una ventana más amplia, intentando aprehender el tema del discurso o el dominio. Gale et al. (1992b) proponen un principio que intuitivamente tiene mucha fuerza, y es que las palabras dentro de un mismo texto o discurso han de estar estrechamente relacionadas semánticamente y, además, sería lógico que varias apariciones de la misma palabra se refieran, en todos o casi todos los casos, a un único sentido. Es el principio de "un sentido por discurso".

En definitiva, el contexto más amplio, el documento, el párrafo[3], juega o debe jugar un papel muy importante a la hora de determinar el sentido correcto de una palabra.

Relacionado con lo dicho, ya sin restringir la búsqueda de información al contexto cercano, se puede tratar de identificar aquellas palabras que aparecen más a menudo asociadas más a menudo a un sentido concreto. Ng y Lee (1996) definen una medida de frecuencia para discriminar las palabras que considera relevantes estadísticamente, aunque nosotros simplificaremos esta búsqueda con una medida de frecuencia pura: aquellas palabras llenas que aparecen asociadas con un determinado sentido en un tanto por ciento m, siendo este valor determinado empíricamente.

Otras fuentes de información: Mención aparte merecen los propios sentidos de las "otras" palabras. Parece muy evidente, también, que si conociéramos el sentido de las palabras que acompañan a la ambigua, si no ayudan a asignarle sentido correcto, seguro que al menos podrían evitar que se le asignara uno incorrecto. La forma de incorporar esta información al sistema de forma que sea útil no está clara, ya que implica, posiblemente, un replanteamiento de la estrategia con la que se aborda la tarea,

3 Si es que los conocemos; el corpus *DSO*, por ejemplo, está compuesto por frases extraídas del corpus *Brown*, y aunque se puede llegar a saber qué frases aparecen en el mismo documento, no se dispone del propio documento completo.

posiblemente con un método incremental de asignación de sentidos a palabras, y la combinación de métodos de desambiguación basados en conocimiento y en corpus.

Herramientas

En el trabajo de implementación que se expone en este capítulo, el preproceso de los ejemplos para enriquecerlo con información sintáctica es vital puesto que de ella dependen la mayor parte de las fuentes de información utilizadas.

La siguiente enumeración menciona algunos programas analizadores que añaden a la frase la información morfo-sintáctica que ayuda al proceso de WSD, y que nuestra experimentación utiliza dependiendo del idioma del texto a tratar. Un analizador de oraciones (*parser*) puede aportar información, entre otros, sobre la categoría gramatical de la palabra (nombre, verbo, adjetivo, ...), lema, sintagmas, rol gramatical (sujeto, verbo, objeto, complemento, ...), detección de palabras compuestas, y dependencias. Dependiendo del programa utilizado, obtendremos una información más o menos completa. Por ejemplo, un *POS-tagger* ofrece información limitada a poco más que las categorías sintácticas de cada palabra.

Lo que a continuación se muestra son las propias descripciones proporcionadas por los desarrolladores de cada producto. Algunos ejemplos breves de las salidas de estos analizadores puede verse en el anejo C.

Tree-tagger: TreeTagger (Schmid, 1994, 1995) es una herramienta para la anotación del texto con información de categorías gramaticales y lema desarrollado dentro del proyecto TC del *Institute for Computational Linguistics of the University of Stuttgart*. Ha sido utilizado, con éxito, para etiquetar textos en alemán, inglés, francés, italiano, griego, y frances antiguo, y es fácilmente adaptable a otros lenguajes si se dispone de un lexicón y un corpus anotado de entrenamiento.

Connexor: (Tapanainen y Järvinen, 1997) su línea de productos *Connexor Machinese* incluye *Machinese Syntax*, que anteriormente se conocía simplemente como *Conexor* (de *Conexor Func-*

tional Dependency Grammar parser). *Machinese Syntax* para inglés es un analizador sintáctico que etiqueta con categorías gramaticales, sintagmas nominales y relaciones sintácticas (sujeto, objeto, complemento, cadenas verbales, funciones adverbiales, etc.). Actualmente, está disponible para inglés, francés, español, finés y sueco (`www.conexor.fi`).

Minipar: (Lin, 1997, 1998) MINIPAR es un analizador sintáctico del inglés, cubriendo categorías gramaticales, sintagmas nominales y verbales, y relaciones sintácticas (sujeto, objeto, etc.). Según sus autores, evaluado sobre el corpus SUSANNE(Sampson, 1995), MINIPAR consigue cerca del 88 % de precisión y el 80 % de cobertura en las relaciones de dependencia. MINIPAR es muy eficiente, analizando 300 palabras por segundo en un Pentium II 300 con 128MB de memoria principal.

Un aspecto interesante que no hemos llegado a evaluar es el impacto de la eficacia de estos distintos analizadores en la tasa de acierto de nuestro propio sistema de WSD. La tecnología en esta área, siendo muy alta su tasa de éxito en el etiquetado, no es perfecta, y sí que podemos afirmar que todos presentan algún defecto de funcionamiento, que si es detectado y corregido contribuye a mejorar la precisión en la desambiguación semántica.

4.3.2 Módulo de aprendizaje: contextos y atributos

Un aspecto importante de la implementación de los modelos de probabilidad de ME es la forma de las funciones que calculan cada atributo. Estas funciones son las que se utilizan para rellenar un vector de características para cada contexto, en el que cada componente nos dice si el atributo al que representa se encuentra o no en el contexto, si su función asociada ha devuelto cierto o falso. Estos vectores son los que se utilizarán en el aprendizaje propiamente dicho.

Funciones de atributo

Supongamos que nos enfrentamos al siguiente corpus de entrenamiento de la palabra *interest*:

«... *considering the widespread* **interest#1** *in the election ...*»
«... *to the best* **interest#5** *of both governments ...*»
«... *anonymous persons expressing* **interest#1** *in the trial ...*»

Hemos decidido que vamos a utilizar la palabra inmediatamente anterior a la que nos "interesa" para extraer dos propiedades de ella: su forma y su categoría gramatical. Podríamos definir las siguientes funciones de atributo:

atrib1: ¿es widespread anterior a interest#1?
atrib2: ¿es best anterior a interest#5?
atrib3: ¿es expressing anterior a interest#1?
atrib4: ¿es ADJ anterior a interest#1?
atrib5: ¿es ADJ anterior a interest#5?
atrib6: ¿es VERB anterior a interest#1?

Vamos a efectuar una pequeña modificación en la forma de definir los atributos, en un intento de hacerlos menos detallados, lo que nos lleva a denominar a los descritos en el ejemplo anterior como **atributos "no relajados"**, en contraposición a los **"relajados"** que definimos y justificamos a continuación.

Atributos "relajados"

WSD es una tarea donde hay muy pocas activaciones de atributos de los que hemos llamado no relajados, sobre todo para propiedades del tipo "palabra anterior" o similares, porque lo normal es que la mayoría de las palabras aparezcan muy pocas veces. Esto significa que los contextos, casi todos, están asociados a vectores de muchos ceros y pocos unos, y por tanto, la frecuencia de valores 1 de un atributo concreto es, casi siempre, muy baja.

Nos planteamos agrupar varias de estas funciones de atributo en en forma de atributos menos específicos, de manera que la propiedad

deja de hacer referencia a un valor específico para utilizar conjuntos de valores. Por ejemplo:

atrib1: ¿es widespread o expressing anterior a interest#1?
atrib2: ¿es best anterior a interest#5?
atrib3: ¿es ADJ o VERB anterior a interest#1?
atrib4: ¿es ADJ anterior a interest#5?

Aquí hemos agrupado, por clases, atributos en tipos de atributo. Es como preguntar *"esta palabra, ¿es de las que aparece alguna vez en el conjunto de aprendizaje con el sentido es y?"*, mientras que los atributos anteriores eran preguntas del tipo, *"esta palabra ¿es la palabra x y el sentido es y?"*

La consecuencia más obvia es la cantidad de funciones que se generarían en un corpus normalmente extenso. Pensando en una propiedad concreta, la forma de la palabra anterior generaría para las primeras, una función por cada valor encontrado en el corpus de $(valor, sentido)$, y para las relajadas, tan sólo una por clase. Y esto multiplicado todas las propiedades que queramos utilizar (que en este ejemplo han sido sólo dos).

El uso de funciones relajadas se convierte en una especie de generalización, donde no pesa tanto el valor concreto de la propiedad, reduciendo el número de funciones e incrementando la frecuencia de activación de los atributos.

En el capítulo 5, dedicada en gran parte a mostrar los resultados de la experimentación sobre distintos tipos de atributos aplicados al entrenamiento, se verá que su utilidad es evidente empíricamente. Es más, unas no necesariamente sustituyen a las otras sino que pueden ser complementarias, se pueden utilizar atributos relajados y no relajados al mismo tiempo en el aprendizaje de una determinada palabra[4].

4 Adelantándonos al capítulo mencionado, la complementariedad entre atributos relajados y no relajados puede deberse a que le damos más peso a un tipo de información (colocaciones de dos palabras a izquierda y derecha de la palabra objetivo, por ejemplo) por una relativa duplicación, coincidiendo con que ese tipo concreto de información es especialmente útil para la desambiguación

Definición automática de atributos

Una de las propiedades de los MME es la posibilidad de utilizar fuentes de información heterogéneas. Esto es posible gracias a la forma que tienen las funciones que definen los atributos (véase la ecuación 3.1). Dado un contexto, la evaluación del conjunto de atributos sobre este contexto dará como resultado un vector de ceros y unos. Evidentemente, dos ocurrencias de contexto pueden ser iguales desde el punto de vista de su caracterización (sus vectores de atributos son iguales).

El contexto en el que se encuentra una palabra objetivo, sea en el aprendizaje o en la clasificación, se describe mediante una serie de atributos. La caracterización del contexto puede llevarse a cabo de muchas maneras y con distintos formalismos; lo que a continuación se desarrolla y detalla es la implementación particular realizada para adecuar los modelos de probabilidad de máxima entropía al problema de la detección del sentido correcto de las palabras.

Para cierto tipo de tareas es posible hacer una definición manual previa de estas funciones y para otras, como la nuestra, la asignación del sentido a las palabras, éstas pueden definirse automáticamente a partir del corpus de aprendizaje.

Retomando el ejemplo anterior, estas funciones tienen la forma:

$$f(x,c) = \begin{cases} 1 \text{ si } c = c' \text{ y } palabra(x,p) = w \\ 0 \text{ en otro caso} \end{cases} \quad (4.1)$$

El argumento p hace referencia a la posición dentro del contexto relativa a la posición de la palabra objetivo. Supongamos que queremos caracterizar el contexto mediante la palabra inmediatamente a la izquierda de la palabra a clasificar interés, supongamos también que los contextos utilizados para aprender son el X_1, el X_2, y el X_3:

X_1 = «El Gobierno argumentó que el **legítimo** interés#2 de la Nación no es incompatible con la solidaridad con los países más pobres.»

X_2 = «El BBVA **muestra** interés#1 por la compra de acciones de la empresa WSD-Systems.»

X_2 = «Los usuarios de ING Direct tienen **mucho** interés#1 en no verse perjudicados por la banca tradicional.»

Las funciones generadas serían, suponiendo que devolverán el valor 0 en el caso de no cumplirse la condición:

$F_{12}(x, c)$ = 1 si palabra(x,-1) = legítimo y c = 2
$F_{21}(x, c)$ = 1 si palabra(x,-1) = muestra y c = 1
$F_{32}(x, c)$ = 1 si palabra(x,-1) = mucho y c = 1

Cada palabra encontrada genera una función por cada clase posible en el corpus, en este caso la 1 y la 2. Estas funciones, evaluadas sobre el conjunto de muestras serían utilizadas por el GIS para calcular 3 parámetros, uno por cada función o característica.

Si fuera el caso de una función de las que hemos llamado "relajadas", el resultado sería el siguiente:

$Fr_1(x, c)$ = 1 si palabra(x,-1) $\in$ {legítimo} y c = 1
$Fr_2(x, c)$ = 1 si palabra(x,-1) $\in$ {muestra, mucho} y c = 2

Si quisiéramos examinar composiciones de palabras, las funciones se pueden adaptar a la forma:

$$f(x,c) = \begin{cases} 1 \text{ si } c = c' \text{ y } palabra(x, p_1) = w_1 \\ \quad \text{y } palabra(x, p_2) = w_2 \\ \quad \text{y... } palabra(x, p_n) = w_n \\ 0 \text{ en otro caso} \end{cases} \tag{4.2}$$

Nótese, finalmente, que la definición de atributos no tiene porqué ser exclusivamente automática y a partir del corpus de aprendizaje. Se podrían incorporar fácilmente preferencias de selección, patrones, composiciones de palabras, etc. previamente establecidos. Dicho de otro modo, la incorporación de información externa al corpus es inmediata si fuese necesaria o conveniente.

Características implementadas

No relajados
- 0: la palabra ambigua
- l: lemas (de palabras llenas) en ±1, ±2, ±3
- s: palabras en posiciones ±1, ±2, ±3
- b: lemas de pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- c: pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- p: categorías gramaticales de palabras en ±1, ±2, ±3
- k_m: lemas de nombres que aparecen en al menos el m % de contextos de un sentido
- r: rol gramatical de la palabra ambigua
- d: la palabra de la que depende la ambigua
- m: palabra compuesta a la que pertenece la ambigua

Relajados
- L: lemas (de palabras llenas) en ±1, ±2, ±3
- W: palabras llenas en ±1, ±2, ±3
- S: palabras en ±1, ±2, ±3
- B: lemas de pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- C: pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- P: categorías gramaticales en ±1, ±2, ±3
- D: la palabra de la que depende la ambigua
- M: palabra compuesta a la que pertenece la ambigua

Figura 4.1. Lista de grupos de atributos

El conjunto de atributos definido para el entrenamiento del sistema se describe en la figura 4.1, donde el uso de minúsculas indica que el tipo es "no relajado", y el de mayúsculas que su definición es "relajada". Se basan, principalmente, en el conocimiento lingüístico del contexto cercano a la palabra ambigua: palabras y composiciones de palabras que la acompañan, categorías gramaticales, rol gramatical, dependencias, etc. Algunos de ellos son posibles dado el tipo de analizador sintáctico que se utiliza[5]. La elección de estas características se inspira en los trabajos de Ng y Lee (1996), Escudero et al. (2000a) y Martínez y Agirre (2000), añadiendo ciertas variaciones a lo allí descrito e información suministrada por el analizador morfo-sintáctico con la forma de nuevos atributos.

5 Por ejemplo, los atributos r y m (y sus versiones relajadas), se pueden utilizar si el texto ha sido analizado con Minipar (véase el ejemplo de salida del anejo C). Los atributos de tipo d se basan en el árbol de dependencias sintácticas generado por este producto: unas palabras dependen de otras, formando sintagmas y cláusulas, información que Minipar representa con la cuarta columna, con números que hacen referencia a la primera columna del análisis.

Los atributos están organizados y se aplican por grupos. La intención es enfocar el sistema hacia tipos de información y no a atributos concretos. Cada grupo es, en realidad, un conjunto de tipos de atributos. Por ejemplo, en el grupo de atributos s están todas las funciones para cada posible valor, en el corpus de aprendizaje, de (a, c, i); a es cualquier palabra que se encuentra en algún ejemplo clasificado como sentido c en una posición $i \in \{-3, -2, -1, +1, +2, +3\}$. Dicho de otra forma, cuando aprendemos de un corpus anotado mediante el grupo de atributos s estamos utilizando todas las palabras en una ventana $[-3.. + 3]$ alrededor de la palabra objetivo.

Otra clasificación de los grupos anteriores atendiendo al tipo de información que manejas es la mostrada en la figura 4.2.

palabras: 0, o, s, l, b, c, L, W, S, B, C,
*categoría gramática*l: p, P
palabras clave: k_m
análisis de la frase: r, d, m, v, D, M

Figura 4.2. Grupos de atributos por fuente de información

El módulo de aprendizaje se completa con el estimador de parámetros, una implementación del procedimiento GIS, tal y como se ha mencionado anteriormente.

Módulo de clasificación

El módulo de clasificación trabaja con contextos no anotados semánticamente, sin información de clase. El contexto se construye atendiendo a los tipos de atributo aprendidos y calcula la probabilidad condicional de cada clase aprendida, usando las funciones de atributo definidas en el aprendizaje y los coeficientes obtenidos por el GIS (según la expresión mostrada en la ecuación 3.7): la clase que obtiene el mayor valor es la clase que el clasificador asigna al contexto.

5

Evaluación

Una vez implementado el sistema, la evaluación de la bondad del mismo se va a realizar, principalmente, con el horizonte del SENSEVAL-2, sobre el que se ha hecho la comparación de la aproximación por máxima entropía.

En primer lugar, en la sección 5.1 se muestra una comparación entre nuestro método y máquinas de vectores soporte y listas de decisión. Aunque no se puede ser concluyente, sí nos atrevemos a decir que ME es tan válido como cualquier otro método supervisado. A decir verdad, nuestra intuición es que lo verdaderamente interesante es cómo afinar el método más que el método en si: preproceso, selección de fuentes de información, etc.

En la sección 5.2 se exponen los datos de evaluación del sistema siguiendo varias estrategias de selección de atributos sobre los conjuntos de contextos de la muestra léxica para español del SENSEVAL-2. Sería más apropiado hablar de selección de grupos de atributos ya que la base del estudio son las agrupaciones enlistadas en el capítulo anterior. La hipótesis de partida es que realizando un estudio previo del corpus de entrenamiento se puede llegar a un conjunto de atributos de aprendizaje más adecuado.

En primer lugar, en la sección 5.2.1 se realiza un análisis de grupos de información implementados en el sistema. De este análisis se concluye que cada palabra necesita un conjunto de características diferenciado si se pretende obtener la máxima precisión. Estos valores servirán de referencia para la evaluación, representando los sistemas ideales a los que se pretende llegar.

Tradicionalmente, la adecuación de los atributos a la tarea se ha abordado de forma global, para todas las palabras por igual. Lo que aquí se propone es un estudio preliminar palabra a palabra de los grupos de atributos. El objetivo es diferenciar el aprendizaje de cada

una con sus atributos particulares, en vez de utilizar un único conjunto para todas.

Dentro de este análisis se calcula, primero, la tasa de acierto de referencia de los sistema ideales para todas las palabras y para cada categoría gramatical (nombres, verbos y adjetivos). Entendemos por sistema ideal el que obtiene la máxima tasa de acierto *a posteriori*, conocidos los sentidos correctos del conjunto de test. Estos datos nos servirán para comprobar si las estrategias de selección de atributos por preproceso del corpus de entrenamiento son efectivamente rentables, y si es mejor realizarlas globalmente o por palabra.

Para la evaluación propiamente dicha, la selección de grupos de atributos por palabras y por categorías en el preproceso del corpus de entrenamiento se aplicará en la sección 5.2.2, y los resultados de la evaluación del aprendizaje posterior se compararán con los obtenidos por los sistemas que compitieron en SENSEVAL-2 en la tarea en español, comparación que resulta, finalmente, altamente satisfactoria.

Esperamos de nuestro sistema que tenga un comportamiento coherente, esto es, que nuestro sistema funcione mejor cuanto mayor sea la información suministrada en el aprendizaje. En la sección 5.3 se realiza una prueba incremental a partir de los mismos datos de la sección anterior, la muestra léxica en español de SENSEVAL-2, de tal forma que se comparan las precisiones obtenidas en diez pruebas consecutivas que incorporan cantidades crecientes de ejemplos de entrenamiento, utilizando en todas ellas los mismos tipos de atributo. Se verá que el sistema, cuantos más ejemplos incorpora al aprendizaje, mejor es la tasa de acierto.

Otro aspecto a considerar es el grado de polisemia de las palabras a procesar. Se espera que WSD funcione mejor cuanto menor sea la cantidad de clases. En la sección 5.4 se muestra también una pequeña prueba utilizando etiquetas de dominio (Magnini y Strapparava, 2000) en lugar de synsets de WN. Las etiquetas de dominio hacen el trabajo de agrupar varios *synsets* bajo una única denominación común, lo que conlleva una disminución de la polisemia en WN. Esto se traduce en un aumento de la precisión de nuestro método en aproximadamente un 7 % frente a una desambiguación con *synsets* como clases.

Por último, en la sección 5.5 se anota brevemente la posibilidad de cooperación entre métodos basados en corpus y basados en el conocimiento, apuntando a que esta cooperación, si se encontrara la forma adecuada, podría incrementar la precisión. Como ejemplo de esta combinación de métodos, se muestra un experimento por el que un método basado en el conocimiento etiqueta previamente los corpus de entrenamiento y prueba, utilizando esta anotación en forma de atributos en la desambiguación final mediante nuestro sistema de WSD basado en los MME. También se aportan datos sobre un sistema por votación que alcanza los mismos resultados que el mejor sistema para los nombres de la muestra léxica en español del SENSEVAL-2.

El resultado final se puede resumir en que nuestro sistema es competitivo si se compara con otros sistemas y métodos actuales, y así se expone en el desarrollo del capítulo que aquí continua.

5.1 Comparación de los MME con otros métodos supervisados

Aprovechando la invitación del proyecto Meaning a colaborar en una de las subtareas programadas, nuestro sistema ME fue incorporado en la comparación de métodos supervisados del proyecto, concretamente máquinas de vectores soporte (SVM) y listas de decisión (DL) (Màrquez et al., 2003).

El entrenamiento se realizó sobre SemCor con sentidos de WN1.6 traduciendo las clasificaciones a WN1.7 en tiempo de evaluación. Esta evaluación se hizo sobre los textos del SENSEVAL-2 *lexical sample* para inglés. Los ficheros de entrenamiento eran los mismos para todos, donde estaban definidos los atributos a usar. Los resultados publicados se muestran en el cuadro 5.1.[1]

El el cuadro 5.1 están marcados en negrita los mejores resultados obtenidos en cada categoría. También se muestran los datos del clasificador basado en el sentido más frecuente (SMF). En este cuadro

1 En el mismo documento se detalla otro experimento de comparación con, además, varias versiones de AdaBoost, pero en un 10FCV sobre Semcor: los resultados globales también daban mínimas ventajas a SVM y ME sobre los demás.

	Ejemplos	SMF	DL	ME	SVM
Nombres	1021	51,00	50,73	**53,71**	53,18
Verbos	1523	35,21	37,10	37,45	**41,17**
Adjetivos	459	48,21	50,11	48,76	**50,76**
Todos	3003	42,57	43,72	44,52	**46,72**

Cuadro 5.1. Comparación con otros métodos supervisados en el proyecto Meaning (Màrquez et al., 2003)

se puede ver un comportamiento que va a ser general en todas las evaluaciones que se van a mostrar: ME aprende y clasifica los verbos bastante peor que adjetivos y sobre todo que nombres.

Hay que tener en cuenta que se trataba de comparar los tres métodos bajo las mismas condiciones, por lo que no se puede comparar con la evaluación oficial de sistemas de SENSEVAL-2. La tasa de acierto es baja, pero no es habitual que haya más verbos para desambiguar que nombres (no sólo en cantidad de ejemplos sino en palabras diferentes), ni se pudieron elegir otros atributos que los suministrados para el experimento. Además, el corpus de entrenamiento no fue SENSEVAL-2 sino SemCor, estando etiquetados cada uno con versiones diferentes de WN. El criterio seguido fue aprender con los sentidos de la versión 1.6 (por el Semcor) y, una vez clasificados los contextos de prueba, convertirlos a WN1.7 para evaluar sobre los sentidos correctos de SENSEVAL-2, proceder éste que parece haber afectado en demasía al resultado final.

Aunque, en principio, y lejos de ser una prueba definitiva, pueda pensarse que SVM es mejor, comparados los resultados palabra a palabra, ME obtiene mejores resultados que SVM en 18 palabras mientras que SVM gana a ME en 16 palabras. El acierto global se explica porque SVM gana "con más claridad" que cuando lo hace ME. Los datos de veces (palabras) que, comparando dos métodos, uno de ellos gana o pierde o empatan se pueden consultar en el cuadro 5.2.

5.2 Evaluación sobre SENSEVAL-2 en español

En la prueba que ahora nos ocupa, hacemos uso de los conjuntos de prueba y evaluación proporcionados por SENSEVAL-2 pa-

	MFS		
DL	30-7-11	DL	
ME	35-5-8	24-15-9	ME
SVM	33-7-8	25-16-7	16-14-18

Cuadro 5.2. Ganados, empatados y perdidos en palabras con diferencias de más de 0,5 puntos en el proyecto Meaning (Màrquez et al., 2003)

ra la tarea conocida como *Spanish lexical sample*, que consiste en la desambiguación de una palabra dentro de un texto que contiene unas pocas frases (véase el ejemplo del anejo B). Estos conjuntos de ejemplos fueron analizados con *Conexor* (Tapanainen y Järvinen, 1997), descrito en la sección 4.3.1.

En este conjunto de experimentos nos centraremos en el ajuste de parámetros del sistema, que consiste en la selección de grupos de atributos que caracterizan la información de aprendizaje (véase la figura 4.1). Nosotros defendemos, y queremos corroborarlo con esta evaluación, que cada palabra es mejor aprendida, se obtiene un mejor clasificador, si se utiliza la información adecuada, hecho éste refrendado por varios autores (Mihalcea, 2002; Pedersen, 2002a). Dicho de otra forma, no todas las palabras han de procesarse de igual manera, es conveniente particularizar el entrenamiento para cada una, realizar una selección de grupos de atributos previa.

Para ayudar a la verificación de esta hipótesis, en primer lugar, se establece un conjunto de valores de referencia (*baseline*) con los que poder comparar los distintos ajustes y sistemas de clasificación que se van a exponer a continuación. Con un conjunto fijo de atributos de aprendizaje, serán los mejores resultados posibles de nuestro método, valores de referencia que deberían ser superados por los sistemas que se van a proponer.

Una vez que ya disponemos de esos valores de referencia, a continuación se propone la evaluación de varios sistemas (atendiendo a diferentes selecciones de atributos) teniendo en cuenta que palabras diferentes se desambiguan mejor con atributos diferentes. Con un preproceso del conjunto de entrenamiento se establecen los mejores conjuntos de atributos por palabra y por categoría, lo que permite sobrepasar las máximas tasas de acierto de referencia mencionadas antes.

Finalmente, comparando con los datos de los sistemas que se presentaron a la tarea, nuestro sistema quedaría en segundo lugar, lo que es un muy buen puesto dada su relativamente escasa sofisticación. Esto nos hace pensar que un preproceso más exhaustivo permitiría batir el mejor resultado del SENSEVAL-2.

5.2.1 Valores de referencia (*baseline*)

Pretendemos, en primer lugar, obtener los valores de referencia que nos permitan comparar la evaluación de varias estrategias de selección de grupos de atributos.

Para calcular estos valores, se aprendió del conjunto de entrenamiento con varias combinaciones de atributos y, probadas sobre el conjunto de test, se seleccionaron aquellas que obtenían la máxima tasa de acierto para todas las palabras (nombres, verbos y adjetivos objeto de la muestra léxica), y también para cada categoría gramatical. El resultado es el techo máximo de acierto que el sistema podría obtener, en distintas condiciones de selección de atributos ideal.

Los valores de referencia se pueden consultar en el cuadro 5.3.

	Acierto	Selección atribs.
Todos	67,1	0LWSBCk5
Nombres	68,3	LWSBCk5
Verbos	59,5	sk5
Adjetivos	78,3	LWsBCp
Todos	68,4	Combinado

Cuadro 5.3. Tasas de acierto de referencia utilizando los datos de la muestra léxica en español de SENSEVAL-2: máximos con un conjunto fijo de atributos global y por categoría

Un sistema con un único conjunto de atributos para todas las palabras conseguiría un 67,1 % de acierto si usara la combinación $0LWSBCk5$.[2]

2 Recordemos que las cadenas de caracteres representado selecciones de atributos se construyen a partir de los datos de la figura 4.1. Por ejemplo, $0LWSBCk5$ indica que se utiliza la forma de la palabra objetivo (0), los lemas y las formas de las palabras en una ventana (-3, +3) (L, W y S), composiciones de dos lemas y

No obstante, esta selección no es la mejor si la aplicamos a las palabras de una categoría gramatical concreta: si sólo tuviéramos en cuenta los nombres, la combinación $LWSBCk5$ es, de todas las probadas, la que mayor acierto obtendría (68,3 %), mientras que para verbos sería $sk5$ (59,5 %) y para adjetivos $LWsBCp$.

Finalmente, un sistema ideal que clasificara nombres, verbos y adjetivos con estos grupos de atributos diferentes (selección de atributos combinada), obtendría un 68,4 % de acierto.

El siguiente paso es comprobar si aún se podría mejorar este dato si la selección se hiciera palabra a palabra. En el cuadro 5.4 se ven los resultados de aplicar la prueba anterior a cada palabra en particular. La columna "contextos" informa de la cantidad de contextos del conjunto de test de cada palabra, y "precisión" es la tasa de aciertos con la mejor combinación de atributos.

Para este cuadro, se aprendió cada palabra con varios conjuntos de atributos y se clasificó el test correspondiente, eligiendo como valor de referencia aquel que obtuvo la máxima tasa de acierto. De esta forma, por ejemplo, para el nombre 'autoridad', la mejor selección de atributos consiguió un 58,8 % de acierto.

Repitiendo el proceso para todas las palabras, se llegó al sistema ideal de referencia: si dispusiéramos de un sistema de selección de grupos de atributos ideal, uno que acertara siempre, la tasa de acierto en la clasificación, globalmente, subiría desde el 67.1 % (del cuadro 5.3) hasta un 72,6 %. Este sistema ideal basado en la mejor selección de conjuntos de atributos obtiene, por tanto, importantes mejoras en cuanto a nombres, verbos y adjetivos.

El problema, obviamente, es encontrar el método para detectar cuáles son los mejores conjuntos de atributos para cada palabra a partir del conjunto de entrenamiento, que es la única información de la que se dispone *a priori*. La siguiente sección plantea esta selección de una forma simple, por la prueba sistemática de conjuntos de atributos en una prueba *3-fold cross-validation* (3FCV) sobre el conjunto de aprendizaje.

---

de dos palabras en una ventana (-2, +2) (B y C), y palabras clave que aparecen asociadas a un sentido más de cinco veces de cada cien contextos ($k5$). Otro conjunto de atributos que aparece en el cuadro 5.3 es p, etiquetas asignadas por el analizador sintáctico a las palabras en una ventana (-3, +3) alrededor de la palabra objetivo. Los caracteres en minúscula indican atributos "no relajados", y "relajados" en mayúscula.

palabra	pos	contextos	precisión	palabra	pos	contextos	precisión
autoridad	N	34	58,8	apuntar	V	44	63,6
bomba	N	37	81,1	clavar	V	54	59,3
canal	N	41	90,2	conducir	V	53	54,7
circuito	N	49	61,2	coronar	V	74	68,9
corazón	N	47	72,3	explotar	V	41	51,2
corona	N	40	80,0	saltar	V	37	45,9
gracia	N	61	83,6	tocar	V	74	62,2
grano	N	22	68,2	tratar	V	70	61,4
hermano	N	57	71,9	usar	V	56	83,9
masa	N	41	68,3	vencer	V	65	83,1
naturaleza	N	56	69,6	brillante	A	87	72,4
operación	N	47	57,4	ciego	A	42	78,6
órgano	N	81	84,0	claro	A	66	89,4
partido	N	57	84,2	local	A	55	90,9
pasaje	N	41	53,7	natural	A	58	58,6
programa	N	47	59,6	popular	A	204	88,2
tabla	N	41	75,6	simple	A	57	80,7
copiar	V	55	47,3	verde	A	33	72,7
actuar	V	73	74,0	vital	A	79	82,3
apoyar	V	49	65,3				
				Todos		2225	72,6
				Nombres		799	72,7
				Verbos		745	64,4
				Adjetivos		681	81,4

Cuadro 5.4. Tasas de acierto de referencia utilizando los datos de la muestra léxica en español de SENSEVAL-2: máximas precisiones con la selección de atributos ideal por palabra

5.2.2 Ajuste de los conjuntos de atributos

Nuestro propósito es detectar la mejor combinación de grupos de atributos procesando el conjunto de entrenamiento. La propuesta consiste en realizar varias pruebas nFCV sobre el conjunto de aprendizaje y el análisis de los resultados promediados para cada palabra. El resultado es un conjunto de sistemas basados, respectivamente, en diferentes definiciones de atributos para clasificadores globales (para todas las palabras), por palabras y por categorías gramaticales.

Mejores resultados globales y por categoría del 3FCV

El cuadro 5.5 muestra los cinco mejores grupos de atributos calculados a partir de un 3FCV sobre el conjunto de entrenamiento[3], mediante la prueba sistemática de varias combinaciones de atributos predefinidas.[4]

Todos		Nombres	
61,5	sbcprdk3	62,0	LWSBCk5
61,1	sbcpdk5	61,6	0LWSBCk5
61,1	LWSBCk5	60,7	sk5
60,9	0sbcprdk3	60,5	0LWSBCPDk5
60,9	sbcpk5	60,4	0sbcprdk3
Verbos		**Adjetivos**	
55,9	sbcprdk3	72,6	0spdk5
55,2	sbcprdk5	72,1	0spk5
55,0	sbcpdk5	72,0	sbcpdk5
54,8	sbcpk5	72,0	0sbcpk5
54,3	0sbcprdk3	71,9	0sbcpdk5

Cuadro 5.5. Ajuste: mejores 5 selecciones de atributos global y por categoría gramatical, calculadas por 3FCV sobre el corpus de entrenamiento

Las tasas de acierto se han calculado globalmente para cada categoría sintáctica y para todas las palabras en conjunto. El mejor resultado al aplicar un mismo grupo de atributos a todas las palabras se consigue con la combinación $sbcprdk3$ (recuérdese que cada letra define un grupo de tipos de atributos[5], como se explica en la sección 4.3.2). Esta combinación obtiene un 61,5 %.

Nuevamente observamos diferencias entre categorías, y son los nombres para los que conseguimos la información correcta, $LWSBCk5$, esto es, el análisis 3FCV (máximo acierto del 62 %) coincide con el sistema ideal del cuadro 5.3 (68,3 %).

3 Dado el tamaño del corpus de entrenamiento, se consideró más conveniente dividir los datos en 3 carpetas en vez de las usuales 10. El corpus se preprocesa para distribuir uniformemente los ejemplos, según el sentido anotado.

4 Nótese que no se establece ningún algoritmo de selección de atributos salvo que, simplemente, se ha establecido una lista bastante extensa de cadenas representando combinaciones de tipos de atributos y se han probado todas.

5 Sólo los atributos de tipo k se acompañan de un parámetro: $k3$ significa "palabras clave al 3 %".

Mejores resultados por palabra del 3FCV

Podemos desglosar los datos del 3FCV por palabra y obtener cuál es el mejor conjunto de atributos para cada una. El cuadro 5.6 muestra estos mejores resultados. El objetivo es determinar la información que es más relevante para cada palabra, en detrimento de establecer un conjunto fijo de atributos con el que aprender todas ellas.

Palabra	Atributos	Acierto	SMF	Palabra	Atributos	Acierto	SMF
autoridad,N	sbcp	58,9	50,3	clavar,V	sbcprdk3	56,1	44,9
bomba,N	0LWSBCk5	76,2	70,7	conducir,V	LWsBCPD	53,4	35,8
canal,N	sbcprdk3	57,9	30,7	copiar,V	0sbcprdk3	45,7	33,8
circuito,N	0LWSBCk5	53,6	39,2	coronar,V	sk5	69,8	32,7
corazón,N	0Sbcpk5	78,1	60,7	explotar,V	0LWSBCk5	59,3	31,8
corona,N	sbcp	72,2	48,9	saltar,V	LWsBC	40,3	13,2
gracia,N	0sk5	63,4	29,5	tocar,V	0sbcprdk3	58,3	31,3
grano,N	0LWSBCr	68,1	48,3	tratar,V	sbcpk5	52,7	20,8
hermano,N	0Sprd	73,1	60,2	usar,V	0Sprd	73,2	66,9
masa,N	LWSBCk5	75,6	45,5	vencer,V	sbcprdk3	69,6	61,8
naturaleza,N	sbcprdk3	52,7	42,4	brillante,A	sbcprdk3	75,6	51,2
operación,N	0LWSBCk5	54,3	37,7	ciego,A	0spdk5	81,2	56,5
órgano,N	0LWSBCPDk5	71,5	51,5	claro,A	0Sprd	91,9	85,4
partido,N	0LWSBCk5	83,9	52,4	local,A	0LWSBCr	79,8	75,0
pasaje,N	sk5	68,5	45,1	natural,A	sbcprdk10	47,1	26,7
programa,N	0LWSBCr	58,7	48,6	popular,A	sbcprdk10	86,5	63,2
tabla,N	sk5	66,3	48,8	simple,A	LWsBCPD	77,6	62,1
actuar,V	sk5	51,4	29,3	verde,A	LWSBCk5	60,1	31,7
apoyar,V	0sbcprdk3	73,0	63,5	vital,A	Sbcp	77,4	44,1
apuntar,V	0LWsBCPDk5	66,1	47,8				

Cuadro 5.6. Ajuste: mejores selecciones de atributos por palabra, calculadas por 3FCV sobre el corpus de entrenamiento

La columna *atributos* en esta tabla describe la combinación de tipos de información que ha obtenido la mejor precisión. Por ejemplo, autoridad obtiene su mejor precisión con $sbcp$, pero la mejor para bomba es $0LWSBCk5$. La columna *acierto* muestra los valores de precisión (el sistema responde al 100 % de los contextos de prueba) y la columna *SMF* el acierto si se hubiera clasificado utilizando el sentido más frecuente.

Tanto el cuadro 5.6 como los datos resumidos en el cuadro 5.5 revelan que la utilización de atributos relajados en nuestro sistema es útil; el uso tanto de éstos como de los no relajados contribuyen a mejorar la precisión incluso para una misma palabra, como ocurre

con el adjetivo vital ($Sbcp$), por ejemplo. Podemos concluir para este adjetivo, y de acuerdo al corpus utilizado, que la información de cada palabra cercana (grupo s) es menos relevante que ciertas palabras compuestas (grupos b y c) que sí tienen un peso estadístico importante para algunos sentidos. De ahí que se utilice el grupo S, que proporciona una información más general y más ambigüa[6].

La forma de la propia palabra objetivo (atributos 0) es útil tanto para nombres como para verbos y adjetivos aunque muchas de las palabras no hacen uso de ella. En general, este atributo parece relacionarse fuertemente con ciertos sentidos, lo que explica su selección en unos casos y no en otros. Por otro lado, la categoría (p y P) no se selecciona tantas veces. Los lemas y las formas de las palabras y composiciones cercanas (L y W, y B y C) son complementarios en la mayor parte de los casos. Hablando de información del análisis de la frase, las relaciones gramaticales y las dependencias (r, y d y D, respectivamente) parecen adecuadas sobre todo si se combinan con otro tipo de atributos. Además, las palabras clave (km) aparecen seleccionadas muchas veces, posiblemente debido a la fuente y el tamaño de los contextos de SENSEVAL-2.

El siguiente paso es aplicar este análisis de atributos a la desambiguación del conjunto de evaluación, y comprobar que tal aproximación es correcta.

5.2.3 Evaluación: aplicación del análisis de atributos

Los datos del cuadro 5.5 y del 5.6 se usaron para construir tres diferentes conjuntos de clasificadores con la intención de comparar su acierto: **MEfix** utiliza la mejor selección para todas las palabras; **MEbfs.pos** la mejor para cada categoría, y **MEbfs** la mejor para cada palabra en particular; finalmente, **vME** es un sistema de votación por mayoría que tiene como entrada las respuestas los tres anteriores.

6 Los atributos "relajados" tienen una frecuencia de activación mayor, por lo que la aparición de una palabra concreta dentro de un contexto asociado a una clase en particular tiene un menor peso en el modelo de probabilidad. Al utilizar en el aprendizaje atributos "no relajados" puede darse el caso de una palabra w_i que aparezca, en el conjunto de entrenamiento, dentro de un único ejemplo anotado como c_j: el clasificador tenderá a asignar siempre la clase c_j a cualquier contexto que contenga esta palabra. El uso de atributos relajados evita estos casos, provocando una menor diferencia entre los valores de probabilidad obtenidos para cada clase posible.

Evaluados sobre el conjunto de test, el cuadro 5.7 muestra la comparación de estos cuatro sistemas. En primer lugar, si comparamos los resultados de este sistema con las aciertos máximos del cuadro 5.3, podemos concluir que, efectivamente, una estrategia de selección de la mejor combinación de atributos garantiza una desambiguación más precisa (la cuestión sería, ahora, el desarrollar un método más eficiente que la pura prueba de una lista de selecciones prefijada).

Todos		Nombres	
67,7	MEbfs.pos	68,3	MEbfs.pos
67,6	vME	67,8	vME
66,7	MEbfs	66,1	MEbfs
65,8	MEfix	64,6	MEfix
Verbos		**Adjetivos**	
58,3	vME	77,4	vME
58,3	MEbfs.pos	77,2	MEbfs.pos
58,3	MEfix	77,1	MEbfs
58,0	MEbfs	75,6	MEfix

MEfix: *sbcprdk3* para todas las palabras
MEbfs: cada palabra con su mejor selección de atributos
MEbfs.pos: *LWSBCk5* para nombres, *sbcprdk3* para verbos, y *0spdk5* para adjetivos
vME: votación por mayoría entre MEfix, MEbfs.pos, and MEbfs

Cuadro 5.7. Evaluación de sistemas ME con diferentes estrategias de selección de grupos de atributos

En general, y ocurre con todos los sistemas de desambiguación léxica actuales, los verbos son difíciles de aprender y la precisión del método decae para esta categoría. En nuestra opinión, los verbos necesitan un tratamiento especial, más información (basada en el conocimiento, quizás), si no es que el conjunto de sentidos definido para los mismos (casi siempre, en los datos que manejamos, WordNet) no es el adecuado. Así mismo, los adjetivos, por su menor grado de polisemia, son más fáciles de aprender y clasificar que nombres y verbos.

El sistema MEfix se acerca bastante al sistema ideal mostrado en el cuadro 5.3. MEfix utiliza la combinación $sbcprdk3$ mientras que el

sistema ideal debería haber aprendido con $0LWSBCk5$. Sin embargo, la merma en precisión es de un 1,3 %. La selección de atributos de forma global garantiza un buen resultado si de aplicar un único conjunto de atributos a todas las palabras se trata.

El sistema MEbfs.pos obtiene una tasa de acierto del 67,7 %, mientras que el sistema ideal obtendría un 68,4 % (sistema combinado en el cuadro 5.3). Nuevamente la diferencia es mínima, validando la propuesta de selección de atributos. El cuadro 5.8 recopila los datos de referencia y los obtenidos en este experimento para una comparación más cómoda.

Referencia			Evaluación		
	Acierto	Selección atribs.	MEbfs.pos	MEbfs	MEfix
Todos	67,1	0LWSBCk5	67,7	66,7	65.8
Nombres	68,3	LWSBCk5	68,3	66,1	64,6
Verbos	59,5	sk5	58,3	58,0	58,3
Adjetivos	78,3	LWsBCp	77,2	77,1	75,6
Todos	68,4	Combinado			
Todos	72,6	BFS			

Cuadro 5.8. Comparación de los mejores sistemas de selección de atributos con los valores de referencia

Sin embargo, el sistema MEbfs, el que se apoya en las mejores selecciones de grupos de atributos por palabra, es claramente inferior al resultado que se podría haber obtenido en el caso de haber coincidido el análisis 3FCV con el sistema ideal (de un 72,6 % esperado, sólo conseguimos un 66,7 %).

Finalmente, el sistema de votación entre estos tres sistemas (vME) no aporta nada, posiblemente porque los tres sistemas se basan, fundamentalmente, en el mismo método y hay demasiado acuerdo entre ellos. Más interesante sería que los métodos de desambiguación usados en la votación fueran más dispares que por la mera diferencia en la selección de atributos. Más adelante, en la sección 5.5, se podrá comprobar que la incorporación de un sistema totalmente diferente sí que mejora la precisión final.

El menos efectivo es MEfix, tal y como se esperaba. Sin embargo, la mejor selección por palabra (MEbfs) no es mejor que la selección

por categoría (MEbfs.pos). En el cuadro 5.9 podemos comparar estos dos sistemas. El primer grupo de datos, *ESPERADOS*, son los valores de referencia, los aciertos que hubiéramos obtenido con los verdaderamente mejores grupos de atributos para cada palabra. Los datos de evaluación, el segundo y tercer grupo, se refieren a la disminución en los sistemas MEbfs y MEbfs.pos respecto de esos valores ideales "esperados".

Las selecciones de atributos marcadas en negrita son las que coinciden con las esperadas, así como los aciertos en negrita en la columnas BFS indican cuando MEbfs gana o empata con MEbfs.pos.

Aunque la diferencia entre verbos y adjetivos es mínima (MEbfs.pos obtiene dos y un acierto más, respectivamente, que MEbfs) lo que llama la atención es que la selección MEbfs no obtenga una tasa de acierto mayor, lo que se agrava con la clara diferencia negativa para los nombres.

El análisis 3FCV del corpus de entrenamiento sólo "acierta" en cuatro palabras, y aventaja a MEbfs.pos en nueve. El problema reside en que, en la mayor parte del resto de las palabras, MEbfs obtiene más errores de clasificación que MEbfs.pos.

Interpretamos que el éxito del sistema MEbfs.pos sobre el MEbfs se debe a que las equivocaciones en la selección de atributos de algunas palabras se compensan y se diluyen en los promedios de la categoría (se prueba cada selección de atributos con todas las palabras de la categoría). Por contra, en la desambiguación individualizada su peso es excesivo en el resultado final. Digamos que la preselección de atributos basada en resultados promedio por categoría, MEbfs.pos, compensa el aprendizaje y no genera resultados tan dispares si comparamos unas clasificaciones con las otras.

La selección por categoría tiene a su favor el método de cálculo. Al promediar entre los resultados obtenidos, estamos adoptando un criterio más conservador. Así, estamos eligiendo unos atributos que funcionan bien para todas las palabras en general (en el análisis del corpus de entrenamiento, al menos), en detrimento de unos máximos de precisión dudosos, en teoría los de la selección MEbfs, cuya confirmación depende del conjunto de evaluación.

Por un lado, si con MEbfs acertamos para una palabra concreta, con MEbfs.pos no prevemos que el resultado sea mucho peor. Por el

palabra	pos	ESPERADO atributos	aciertos	BFS atributos	decre.	BFS.pos atributos	decre.
autoridad	N	0LWSBCr	20	sbcp	-5	LWSBCK5	-4
bomba	N	0LB	30	0LWSBCk5	-1	LWSBCk5	-1
canal	N	**LWSBCk5**	37	sbcprdk3	-10	**LWSBCk5**	0
circuito	N	sbcprdk10	30	0LWSBCk5	-4	LWSBCk5	-3
corazón	N	0sbcprdk5	34	0Sbcpk5	-8	LWSBCk5	-3
corona	N	0LB	32	sbcp	-5	LWSBCk5	-1
gracia	N	LWsBCp	51	0sk5	**-1**	LWSBCk5	-6
grano	N	0sbcp	15	0LWSBCr	**-4**	LWSBCk5	-6
hermano	N	0LWSBCk5	41	0Sprd	-3	LWSBCk5	0
masa	N	0LWSBCk5	28	LWSBCk5	-1	LWSBCk5	-1
naturaleza	N	LWSBC	39	sbcprdk3	-3	LWSBCk5	0
operación	N	Sk5	27	0LWSBCk5	-2	LWSBCk5	-2
órgano	N	Sk5	68	0LWSBCPDk5	-4	LWSBCk5	-2
partido	N	0LWSBCPDk5	48	0LWSBCk5	-1	LWSBCk5	-1
pasaje	N	0sbcpk5	22	sk5	**0**	LWSBCk5	-2
programa	N	LWSBC	28	0LWSBCr	-1	LWSBCk5	0
tabla	N	**sk5**	31	**sk5**	**0**	LWSBCk5	-3
copiar	V	0LB	26	0sbcprdk3	-5	sbcprdk3	0
actuar	V	**sbcprdk3**	54	sk5	-1	**sbcprdk3**	0
apoyar	V	sbcp	32	0sbcprdk3	-4	sbcprdk3	-3
apuntar	V	sbcprdk10	28	0LWsBCPDk5	-3	sbcprdk3	-3
clavar	V	LB	32	sbcprdk3	-8	sbcprdk3	-4
conducir	V	sk5	29	LWsBCPD	-10	sbcprdk3	-9
coronar	V	0sbcpk5	51	sk5	-4	sbcprdk3	-2
explotar	V	0sbcpk5	21	0LWSBCk5	**-1**	sbcprdk3	-4
saltar	V	LWSBCPD	17	LWsBC	-3	sbcprdk3	0
tocar	V	0sbcpk5	46	0sbcprdk3	**-2**	sbcprdk3	-7
tratar	V	**sbcpdk5**	43	**sbcpk5**	**0**	sbcprdk3	-7
usar	V	0sbcprdk3	47	0Sprd	-3	sbcprdk3	-3
vencer	V	LWSBCk5	54	sbcprdk3	-4	sbcprdk3	-4
brillante	A	0sbcprdk5	63	sbcprdk3	-3	0spdk5	-3
ciego	A	0LWSBCPDk5	33	0spdk5	**0**	0spdk5	0
claro	A	LB	59	0Sprd	-3	0spdk5	-5
local	A	0LWSBCk5	50	0LWSBCr	**-2**	0spdk5	-6
natural	A	0sp	34	sbcprdk10	**-3**	0spdk5	-6
popular	A	LWsBCp	180	sbcprdk10	-5	0spdk5	-3
simple	A	0sbcprdk5	46	LWsBCPD	-5	0spdk5	-2
verde	A	LB	24	LWSBCk5	-2	0spdk5	-2
vital	A	0sprd	65	Sbcp	-6	0spdk5	-1
Todos			1615		-130		-109
Nombres			581		-53		-35
Verbos			480		-48		-46
Adjetivos			554		-29		-28

Cuadro 5.9. Comparando las pérdidas de precisión de MEbfs y MEbfs.pos respecto de los clasificadores "ideales"

contrario, si nos equivocamos, lo más probable es que la selección del segundo no produzca una caída de la precisión tan grande como la del primero.

Así, llegamos a la conclusión de que la prueba sistemática de conjuntos de atributos es válida si se quiere acercar el sistema final a los sistemas ideales que no diferencian más allá de la categoría de las palabras, pero no es suficiente si pretendemos llegar a detallar un conjunto diferente para cada palabra. Es probable que, además de necesitar un método de selección más sofisticado, este comportamiento se deba al tamaño del corpus de aprendizaje, que es obviamente mucho mayor para una categoría que para una única palabra, lo que hace más fiable la prueba nFCV.

Comparación con los sistemas de SENSEVAL-2

El cuadro 5.10 compara todos estos sistemas con los resultados oficiales de la tarea lexical sample para español del SENSEVAL-2. Si nuestros sistemas hubieran competido entonces, hubiéramos obtenido unos excelentes resultados, sobre todo en nombres y adjetivos.

Todas		Nombres		Verbos		Adjetivos	
71,3	jhu(R)	70,2	jhu(R)	64,3	jhu(R)	80,2	jhu(R)
67,7	**MEbfs,pos**	68,3	**MEbfs,pos**	59,5	css244	77,4	**vME**
67,6	**vME**	67,8	**vME**	58,4	umd-sst	77,2	**MEbfs,pos**
67,0	css244	66,1	**MEbfs**	58,3	**vME**	77,2	css244
66,7	**MEbfs**	65,2	css244	58,3	**MEbfs,pos**	77,1	**MEbfs**
65,8	**MEfix**	64,6	**MEfix**	58,3	**MEfix**	75,6	**MEfix**
62,7	umd-sst	62,1	duluth 8	58,0	**MEbfs**	72,5	duluth 8
61,7	duluth 8	61,2	duluth Z	51,5	duluth 10	71,2	duluth 10
61,0	duluth 10	61,1	duluth 10	51,3	duluth 8	70,6	duluth 7
59,5	duluth Z	60,3	umd-sst	51,1	ua	70,3	umd-sst
59,5	duluth 7	59,2	duluth 6	49,8	duluth 7	68,9	duluth 6
58,2	duluth 6	59,0	duluth 7	49,0	duluth Z	68,9	duluth Z
57,8	duluth X	58,6	duluth X	47,8	duluth X	68,7	ua
56,0	duluth 9	55,7	duluth 9	47,7	duluth 9	67,8	duluth X
54,8	ua	51,4	duluth Y	47,4	duluth 6	65,5	duluth 9
52,4	duluth Y	46,4	ua	43,1	duluth Y	63,7	duluth Y

Sistemas: JHU(R) de la Johns Hopkins University;
CSS244 de la Stanford University;
UMD-SST de la University of Maryland;
Duluth de la University of Minnesota;
UA de la Universidad de Alicante.

Cuadro 5.10. Comparando con los sistemas de SENSEVAL-2 español

Teniendo en cuenta que nuestros sistemas basados en ME son relativamente simples puesto que no realizan un preproceso del corpus complejo, ni se combinan con otros métodos, el resultado se puede considerar altamente satisfactorio. El problema se reduce a encontrar un proceso de selección de fuentes de información más eficiente que la mera prueba unas cuantas combinaciones posibles. Es, por otro lado, una estrategia dependiente en demasía del corpus de entrenamiento, lo que puede influenciar negativamente en el aprendizaje y clasificación posterior pero, puesto que todos los sistemas se evalúan bajo las mismas condiciones, no parece descabellado afirmar que nuestro sistema puede competir al más alto nivel de precisión con los demás.

Así mismo, la influencia de los analizadores elegidos para el etiquetado sintáctico previo no está bien estudiada, pero sí que hemos comprobado que una revisión y corrección de ciertos "vicios" de estas herramientas incrementa la precisión de los clasificadores. Un aspecto poco explotado de este análisis son las relaciones entre sintagmas dentro de la oración (sujeto-verbo, sujeto-verbo-objeto, ...), tan sólo se han probado unos pocos atributos simples.

A esto hemos de añadir, en un futuro próximo, información sobre entidades (personas, lugares, expresiones temporales), anafórica, e incluso el sentido de las palabras del contexto, posiblemente en un proceso de etiquetado incremental, siempre y cuando podamos estar seguros de que las anotaciones son correctas en un alto porcentaje.

Por nuestros resultados, y por los sistemas presentados en SENSEVAL-2, parece clara la ventaja de utilizar varios sistemas de desambiguación en paralelo y su posterior combinación en una única respuesta. Esta idea es la base de todo el desarrollo posterior que se comenta en la sección 5.5. Estos datos también parecen indicar un límite para los sistemas basados en corpus y aprendizaje automático estadístico, como ya varios autores están anunciando: los métodos actuales consiguen un resultado que difícilmente podrán mejorar a no ser que un nuevo enfoque permita traspasar esta frontera.

5.3 Prueba incremental

Como comprobación de que el método de MME implementado es adecuado para la tarea, y como refrendo de que la incorporación de más ejemplos contribuyen a un mejor aprendizaje, el cuadro 5.11 muestra la evaluación sobre los mismos datos de forma incremental.

En este caso se han dividido los corpus de aprendizaje en 10 partes, y en cada prueba (hasta 10) el corpus de entrenamiento del sistema añadía una parte más a la anterior. De esta forma, la prueba '1' sólo utiliza una parte del corpus de aprendizaje, la '2' dos, y así hasta la décima prueba que utiliza todo el corpus original.

atributos	POS	1	2	3	4	5	6	7	8	9	10
LWSBCk5	N	65,2	54,1	59,4	59,6	59,8	61,6	61,6	62,0	63,7	68,5
sbcprdk3	V	42,6	45,0	47,0	48,3	48,7	49,3	49,5	50,6	50,7	56,0
0spdk5	A	64,6	65,8	71,4	72,0	74,3	74,4	75,3	75,5	74,3	76,2
	ALL	57,4	54,6	58,9	59,6	60,5	61,4	61,8	62,3	62,6	66,7

Cuadro 5.11. Prueba incremental

Se comprueba que, para todas las categorías de palabras, el sistema consigue incrementar su tasa de acierto a medida que la cantidad de ejemplos anotados es mayor.

5.4 *WordNet Domains* como conjunto de clases

Podemos cambiar el conjunto de clases y utilizar etiquetas de dominio en lugar de sentidos de WN; por diferenciar, podemos llamar a las clasificaciones resultantes como desambiguación del dominio de las palabras (*word domain disambiguation*, WDD) (Magnini y Strapparava, 2000; Magnini et al., 2001).

Por un lado, etiquetar con información de esta naturaleza provoca agrupaciones de synsets y la reducción del grado de polisemia, lo que debe redundar en un aumento del acierto respecto de WSD. Por otro, varios investigadores defienden que tareas tales como recuperación

de información y question answering podrían mejorar sus resultados con desambiguación de dominios antes que por synsets.

Magnini y Cavaglia (2000) propusieron un enriquecimiento de WN1.6, *WordNet Domains* (WND), utilizando *subject fields codes*, conjuntos de palabras relevantes en un dominio concreto, basándose en las propias etiquetas de los diccionarios (medicina, arquitectura), y que fue la base de los sistemas ITC-irst en el SENSEVAL-2. Otra propuesta de enriquecimiento de WN con información de dominios se puede ver en (Montoyo et al., 2001) que extrae las etiquetas del sistema de catalogación IPTC.

En este subapartado se van a mostrar los resultados obtenidos al evaluar los nombres del corpus DSO con WND y comparándolos con los de WSD mostrados anteriormente. En este caso hemos optado por el corpus DSO por dos razones: es un corpus más extenso, y contiene ejemplos en inglés cuyas etiquetas originales teníamos convertidas a sentidos de WN1.6, lo que facilita la tarea de transformar sentidos en dominios.

120 Nombres
872 Ejemplos(*) por nombre
Dominios: 11 nombres monosémicos
Synsets: 1 nombre monosémico
3,5 dominios(*) por nombre
4,8 synsets(*) por nombre
(*) valores promedio

Cuadro 5.12. Promedios de synsets y dominios por palabra, de los nombres del DSO

El corpus DSO, después de convertir los sentidos anotados a WN1.6, y del preproceso explicado anteriormente que divide los ficheros de cada palabra en 10 partes, tiene 120 nombres y 938 sentidos: 100 nombres reducen su polisemia hasta 629 etiquetas de dominio, con una disminución promedio de 4,8 synsets por palabra hasta 3,5 dominios (véase el cuadro 5.12). Algunos nombres, 11, se convierten en monosémicos, con lo que la clasificación obtiene, obviamente, un 100 % de éxito (se han mantenido estos datos puesto que se quiere mostrar la ganancia en aciertos al comparar WDD con WSD). El preproceso para preparar el 10FCV, por el que se elimi-

nan los sentidos con menos de 10 ejemplos[7], provoca que el nombre college sea monosémico tanto con *synsets* como con dominios.

Como muestra, el cuadro 5.13 contiene los mejores resultados para un subconjunto de los nombres en una evaluación 10FCV. El objetivo es comparar WDD (parte izquierda del cuadro) con WSD (parte derecha). Así, las columnas *Doms* (por dominios) y *Sens* (por sentidos) muestran el número de clases de cada palabra (polisemia). Para ambos tipos de desambiguación, *Atributos* es la selección de grupos de atributos con el mejor resultado, Acierto la tasa de acierto, y *DMF* y *SMF* es el acierto que se obtendría seleccionando el dominio y el sentido más frecuente, respectivamente.

	Doms	Atributos	Acierto	DMF	Sens	Atributos	Acierto	SMF
action,N	4	sprdm	59,4	46,8	5	0sprdmk10	52,7	46,8
activity,N	2	0sBCprdmk10	87,0	85,7	3	0sprdm	71,3	68,8
art,N	2	Más frecuente	97,5	97,5	4	0sprdm	65,2	48,0
body,N	2	0LSsBCprdm	86,3	77,9	4	0LSsBCprdm	68,6	60,5
book,N	3	0sbcprdmk10	84,4	80,6	4	0sprdmk10	70,1	65,0
business,N	6	0sbcprdmk10	65,0	50,3	7	0sBCprdmk10	64,2	50,3
case,N	3	0sbcprdmk10	74,6	66,8	9	0sbcprdmk10	56,8	32,5
center,N	3	0LSsBCprdm	80,9	58,3	6	0sbcprdmk10	72,4	58,3
church,N	2	0sprdm	70,5	67,1	3	0sprdmk10	67,1	62,1
condition,N	2	0sbcprdmk10	87,9	84,6	3	0LSsBCprdm	83,4	79,6
course,N	4	0sBCprdmk10	78,9	49,4	5	0sBCprdmk10	72,1	42,3
interest,N	5	0sprdmk10	71,8	45,9	6	0sprdmk10	70,9	45,9
line,N	14	0LSsBCprdm	65,3	42,5	22	0sprdmk10	56,0	22,7
work,N	3	0LSBCprdm	80,6	71,7	6	0sprdmk10	54,6	32,8
Total	55		74,7	61,8	87		63,6	46,6

Cuadro 5.13. Ejemplo de la mejor selección de atributos para WDD y WSD

El cuadro 5.14 muestra los resultados de evaluación de los 120 nombres del corpus DSO cuando los conjuntos de clases están formados por etiquetas de dominio en vez de sentidos de WN. La primera consecuencia es la disminución de clases posibles para algunas de las palabras y el aumento en la tasa de acierto del método. Obviamente, aquellas palabras que no reducen su conjunto de clases no contribuyen a este aumento.

7 este detalle no afecta al experimento puesto que lo que pretendemos es mostrar que un conjunto de clases más reducido ayuda a obtener una mejor precisión

Atributos	Dominios	Synsets	Dif
SMF	68,7	58,7	
LB	73,5	64,6	+8,94
SP	74,8	66,6	+8,20
0LB	75,4	67,1	+8,34
0SP	75,7	67,8	+7,96
sp	77,2	69,5	+7,70
0sp	77,7	70,2	+7,52
sprdm	78,1	70,6	+7,48
0sprdm	78,4	71,0	+7,37
0LSsBCprdm	78,6	71,0	+7,58
0sprdmk10	78,7	71,4	+7,26
0sBCprdmk10	78,7	71,4	+7,27
0sbcprdmk10	78,7	71,4	+7,33

Cuadro 5.14. Resultados de WDD y WSD

Siguiendo el método 10FCV se han realizado varias pruebas con distintas selecciones de atributos. La columna *Dominios* muestra la tasa de acierto promedio para cada selección de atributos cuando las clases son *WN Domains*, y la columna *Synsets* cuando son los synsets de WN. La columna *Dif* muestra la diferencia entre uno y otro valor[8]. La primera fila de resultados, la etiquetada con *SMF*, son los valores obtenidos al clasificar cada contexto con el significado más frecuente en el corpus.

El mejor resultado para WDD se obtiene con la selección de atributos más compleja ($0sbcprdmk10$) y es de, aproximadamente, un 7 % más de acierto que la obtenida para WSD. La prueba *t* de Student sobre el 10FCV (Dietterich, 1998) $t_{9,0,975} = 1{,}833$ muestra que la mejor selección de atributos ($0sbcprdmk10$) no es significativamente mejor que las siguientes cuatro selecciones (con $sprdm$ ya es significativo). A medida que los clasificadores están más informados, utilizan más atributos, las diferencias entre uno y otro conjunto de clases se hacen más pequeñas pero en valores casi despreciables.

8 Las anomalías aparentes en los valores de la columna *Dif* son efecto del redondeo hasta un decimal.

5.5 Sobre una posible cooperación con otros métodos: Marcas de Especificidad

Según Pedersen (2002b), en general, se pueden establecer unas proporciones generales para los ejemplos utilizados en la desambiguación según la dificultad de su clasificación. Estos valores serían *50/25/25* que corresponden a la hipótesis de que el 50 % de los contextos son fáciles de desambiguar, un 25 % presentan una dificultad mayor y el otro 25 % una dificultad extrema. Pedersen llegó a esta conclusión comparando, dos a dos, las respuestas de los sistemas participantes en las tareas de muestra léxica de español e inglés de SENSEVAL-2: promediando todas las comparaciones, en un 50 % de los contextos de prueba los dos sistemas desambiguaron correctamente, y en un 25 %, alguno de los dos dio la respuesta correcta.

En un experimento paralelo en nuestro grupo de investigación (Montoyo et al., 2002), utilizando Semcor y, simultáneamente, marcas de especificidad (SM, de *Specification Marks*) y ME, se clasificaron 267 ejemplos elegidos de la colección como conjunto de prueba. Los resultados parecen confirmar la afirmación de Pedersen. Obviamente, los valores concretos no son tan importantes como la clasificación de las instancias según su dificultad. En el cuadro 5.15 se muestran las coincidencias en la clasificación de ambos clasificadores y cuándo aciertan ambos o alguno de los dos.

Casos comparados	267
Método	Acierto
ME	61,49 %
SM	32,96 %
Coinciden	30,71 %
Al menos uno acierta	71,91 %
Aciertan los dos	22,47 %
Fallan los dos	28,09 %

Cuadro 5.15. Comparando ME y MES sobre un subconjunto de Semcor

Ambos sistemas coincidieron en la clase asignada en un 31 % de los contextos, siendo un 8 % las respuestas coincidentes y erróneas. En cualquier caso, alguno de ellos (o los dos) da la respuesta correcta

en un 72 % de los casos. Esto quiere decir que el clasificador perfecto que combine estos dos métodos, aquel que es capaz de elegir la respuesta correcta de las dos propuestas para un ejemplo, obtendría un 72 % de acierto, lo que significa una mejora del 10 % sobre el método más preciso. Parecidos resultados se obtuvieron añadiendo un tercer clasificador basado en densidad conceptual (Agirre y Rigau, 1996).

La conclusión obvia es que los métodos basados en el conocimiento sufren una baja precisión pero pueden ser útiles como ayuda a los métodos basados en corpus.

Los dos experimentos que se comentan a continuación corroboran empíricamente la afirmación anterior. Tanto si se utiliza MES para incorporar información adicional en forma de atributos en ME como si se añade a un sistema de votación el incremento en la tasa de acierto justifica la exploración de esta vía de cooperación entre métodos.

5.5.1 WordNet Domains como nuevos atributos

Este experimento pretende evaluar un nuevo tipo de atributo que utiliza etiquetas de *WN Domains* en el contexto cercano. Para ello vamos a utilizar los datos de la muestra léxica de SENSEVAL-2 para inglés. Concretamente en una ventana de 2 palabras a izquierda y derecha de la palabra objetivo (no se tiene en cuenta la etiqueta de esta última). Simplificando, el proceso consiste en dos fases: la primera es el preproceso del conjunto de entrenamiento con SM, que anota los nombres de los contextos con dominios, y la segunda es el propio aprendizaje de ME con esos datos como atributos adicionales.

Necesitamos predesambiguar el texto cercano utilizando *WN Domains* como clases. Con la heurística de dominio de SM (Montoyo et al., 2003), podemos etiquetar el conjunto de entrenamiento con información de dominio. Posteriormente, esta información se incorpora en forma de atributos a ME.

El cuadro 5.16 muestra la comparación entre la desambiguación del conjunto de evaluación aprendiendo sin y con información de dominios. En total, la incorporación de estos atributos supone una mejora de 2 puntos, de nuevo con grandes diferencias entre palabras.

Nombres	No Doms.	Sí Doms,	Mejora	Nombres	No Doms.	Sí Doms,	Mejora
art	68,3	68,3	0,0	grip	15,8	15,8	0,0
authority	53,8	56,3	2,5	hearth	79,3	79,3	0,0
bar	51,9	51,0	-1,0	holiday	100	100	0,0
bum	86,5	91,9	5,4	lady	87,5	90,0	2,5
chair	89,8	89,8	0,0	material	36,2	51,7	15,5
channel	12,5	18,8	6,3	mouth	56,9	58,8	2,0
child	61,0	62,7	1,7	nation	72,0	72,0	0,0
church	60,0	60,0	0,0	nature	43,2	46,0	2,7
circuit	24,5	38,8	14,3	post	51,2	51,2	0,0
day	64,0	64,7	0,7	restraint	48,4	51,6	3,2
detention	90,9	86,4	-4,6	sense	43,2	48,7	5,4
dyke	80,0	80,0	0,0	spade	82,4	88,2	5,9
facility	71,4	64,3	-7,1	stress	46,0	40,5	-5,4
fatigue	86,8	86,8	0,0	yew	79,2	79,2	0,0
feeling	60,4	66,7	6,3				
Total	59,6	61,8	2,2				

Cuadro 5.16. Resultados de aplicar la heurística de dominio de marcas de especificidad

Resumiendo estos datos, de 29 nombres, 4 empeoraron al utilizar la información adicional pero 14 mejoraron. La razón de que la mejora total sea de sólo un 2 % está principalmente en que gran parte de los nombres no incrementan su acierto. Tampoco se debe olvidar que el etiquetado por parte de SM no es correcto al 100 %.

Aún así, hace falta un estudio más profundo del impacto de estas etiquetas de dominio en el proceso de desambiguación. No se ha tenido en cuenta si el dominio factotum (etiqueta de *WNDomains* que indica que cierto nombre resulta inclasificable) debe eliminarse. Tampoco se ha probado una ventana de palabras mayor en torno al nombre objetivo. Además, sería interesante ver la mejora que se pueda producir en verbos y adjetivos.

5.5.2 Votación por mayoría

En otro experimento de combinación de métodos, el cuadro 5.17 incluye el enriquecimiento del sistema vME: **vME+SM**. Este sistema de votación incluye, además de los tres ya mencionados de ME (véase la sección 5.2.2), otro clasificador basado en **marcas de especificidad**.

Vuelve a ser una comparación a posteriori de nuestros desarrollos frente a los resultados obtenidos en SENSEVAL-2 para español.

Todas		Nombres	
0,713	jhu(R)	0,702	jhu(R)
0,684	**vME+SM**	0,702	**vME+SM**
0,677	MEbfs,pos	0,683	MEbfs,pos
0,676	vME	0,678	vME
0,670	css244	0,661	MEbfs
0,667	MEbfs	0,652	css244
0,658	MEfix	0,646	MEfix
0,627	umd-sst	0,621	duluth 8
0,617	duluth 8	0,612	duluth Z
0,610	duluth 10	0,611	duluth 10
0,595	duluth Z	0,603	umd-sst
0,595	duluth 7	0,592	duluth 6
0,582	duluth 6	0,590	duluth 7
0,578	duluth X	0,586	duluth X
0,560	duluth 9	0,557	duluth 9
0,548	ua	0,514	duluth Y
0,524	duluth Y	0,464	ua

Cuadro 5.17. Combinando con Marcas de Especificidad en SENSEVAL-2 español

Puesto que es un método basado en el conocimiento de aplicación, hasta ahora, exclusivamente en nombres, el cuadro sólo muestra los resultados para esa categoría, y los resultados teniendo en cuenta todas las categorías. El resultado del sistema vME+SM es óptimo, puesto que iguala, para los nombres, al mejor sistema.

5.6 Conclusiones

Los modelos de probabilidad condicional de máxima entropía son adecuados para la tarea de desambiguación léxica automática. Comparado con otros métodos es competitivo.

El hecho crucial del éxito en la clasificación es la elección de las fuentes de información apropiadas para el aprendizaje. Cada palabra es mejor clasificada si se ha aprendido con un conjunto particular de atributos, y esta característica no es exclusiva de ME. Obviamente, estamos hablando de corpus concretos, no estamos en condiciones de evaluar la certeza de esta hipótesis en condiciones reales ya que no disponemos de medios para evaluarla contra un texto cualquiera.

La selección de fuentes de información que se ha mostrado aquí es demasiado simple, la prueba sistemática de combinaciones

de tipos de atributos sobre el corpus de entrenamiento buscando la mejor. Se ha demostrado que, efectivamente, la búsqueda de esta mejor selección de atributos ayuda a incrementar la precisión, aunque para llegar a diferenciar el aprendizaje palabra por palabra se necesitaría un método más sofisticado.

También se ha probado empíricamente que la reducción de clases que supone *WordNet Domains*, o simplemente la disminución de la polisemia por el sistema que sea, influye positivamente en la eficacia de los sistemas de WSD.

Así, mismo, se ha explorado la combinación de métodos supervisados y no supervisados con claros indicios de que esa cooperación es productiva.

Se ha de profundizar en todos estos puntos, así como en el preproceso de los conjuntos de aprendizaje y evaluación, con el objetivo de conseguir un sistema de WSD completo y eficiente.

6

Alta precisión en WSD: método incremental

Hasta ahora, la desambiguación léxica se ha llevado a cabo en dos fases: aprendizaje y clasificación. Dicho de forma imprecisa, el aprendizaje estándar se realiza una sola vez, con un conjunto previamente anotado de ejemplos en el que se va a basar la posterior clasificación.

A menudo nos quejamos de la insuficiencia del corpus, bien sea por cantidad de ejemplos bien por inadecuación de los ejemplos a la realidad. El problema básico continúa siendo la disponibilidad de tales corpus puesto que ahora mismo son pocos, pequeños, y exclusivos de algunos idiomas (aún cuando se está dedicando mucho esfuerzo a subsanar esta carencia).

Pero cierto es que fuentes de texto sin anotar las hay, y muchas, de fácil acceso y virtualmente inagotables. Cómo aprovechar tales fuentes para la generación automática de corpus anotados, y en particular para WSD, es la tarea por resolver y que, parece, estamos lejos de solucionar de forma satisfactoria.

En realidad estamos refiriéndonos al cuello de botella que sufren los métodos basados en corpus, a la recolección y anotación de conjuntos de ejemplos lo suficientemente extensos como para ser efectivos. Si somos capaces de obtener esos recursos de forma automática esperamos un mayor acierto en la tarea que lo utilice.

Evidentemente, tal y como están las cosas, un corpus generado automáticamente no garantiza la ausencia de errores pero si que el esfuerzo de anotación ha sido mínimo (o mucho menor, al menos). El problema es especialmente grave en WSD ya que el esfuerzo anotador es mayor que para otras tareas. Para conseguir un sistema supervisado de WSD de amplia cobertura y alta precisión, según Ng (1997), se necesitan ejemplos para al menos 3200 palabras, lo que

supone 16 personas-año de trabajo de anotación, asumiendo una media de 1000 ejemplos anotados por palabra;Mihalcea (2003), citando al propio Ng (1997), eleva el cálculo a 80 personas-año.

6.1 Objetivos

Hagamos una ligera variación en lo que hemos llamado aprendizaje, intentemos aprovechar el propio proceso de desambiguación para mejorar lo aprendido. La idea es muy simple, si de un ejemplo recién clasificado estamos muy seguros, lo añadimos al conjunto anotado y volvemos a aprender y clasificar, incrementando progresivamente el tamaño del corpus anotado inicial.

Este aprendizaje progresivo, y conservador, es lo que hemos llamado **reentrenamiento**, el proceso por el cual, a partir de dos conjuntos de ejemplos, uno anotado semánticamente y otro no, se realizan sucesivos aprendizajes incrementando el conjunto inicial con clasificaciones del conjunto no anotado. El problema es obvio y difícil de abordar en estos momentos: cómo estar seguros de la clase asignada a un contexto para no añadir errores al siguiente aprendizaje.

En principio, los modelos de ME son adecuados para esta tarea si recordamos que éstos nos proporcionan las probabilidades de que un contexto pertenezca a cada clase posible. Si establecemos un criterio cualitativo sobre estas probabilidades, y ese criterio se demuestra válido, ya tenemos la forma de asegurar la máxima precisión.

Lo que a continuación se detalla es la propuesta de un método incremental general, aplicado a la resolución del sentido de las palabras, que no es exclusivo de los modelos de máxima entropía sino que se puede utilizar con cualquier método de aprendizaje automático. Se definirá la propuesta y los parámetros que afectan a su funcionamiento, y se mostrarán los resultados obtenidos.

El propósito final de tal propuesta (o una de sus posibles aplicaciones) es asegurar la confianza necesaria en la anotación. Si conseguimos una alta precisión, aún si es a costa de una cobertura mayor, podríamos pensar en la aplicación de WSD a varios objetivos, entre otros:

- la generación automática de corpus anotados, que incluye el incremento de los actuales conjuntos de entrenamiento de los sistemas de WSD.
- la realimentación de los sistemas de WSD añadiendo información de sentidos como nuevos atributos, o como apoyo a sistemas basados en el conocimiento
- la generación de otros recursos como puedan ser preferencias de selección, patrones semánticos, ontologías, etc.
- la expansión de las preguntas de un sistema de búsqueda de respuestas basado en los sentidos de los que sí se está seguro.

Por eso, y como se explicará en la sección 6.5.1, nuestra prioridad no es la cobertura (estándar o absoluta) sino la precisión. El cómo se utilicen estas clasificaciones con alto grado de confianza no es objeto de este trabajo, tan sólo se apuntan, a través de los experimentos que se desarrollarán más adelante, posibles marcos de aplicación del reentrenamiento que incluyen, también, SENSEVAL-2.

6.2 Antecedentes

El método que se va a proponer en este capítulo se inscribe dentro del amplio abanico de técnicas incrementales o de semilla (*bootstrapping*) y, más concretamente, el punto de partida es el trabajo de Blum y Mitchell (1998), donde se propone un nuevo método de clasificación iterativo, el *co-training* (a partir de ahora **coentrenamiento**).

Sin embargo, uno de los primeros métodos incrementales que se citan específicamente para WSD es el de Yarowsky (1995). Se trata de un método no supervisado que se basa en dos restricciones: que una palabra tiende a tener un único sentido dentro de un mismo discurso, y también dentro de una misma "colocación" (*one sense per discourse, one sense per collocation*). El método se evaluó sobre un pequeño conjunto de palabras con dos posibles sentidos cada una. Partiendo de las definiciones de un diccionario, se construyó una semilla con colocaciones representativas de cada sentido y se utilizó como entrada para un algoritmo de listas de decisión. El incremento del

corpus anotado se hacía con aquellas clasificaciones que superaban un cierto umbral de probabilidad.

Este trabajo tuvo (y tiene) un gran impacto en la comunidad científica especializada en WSD por la alta precisión conseguida (95 % aproximadamente), aunque es evidente que las condiciones del experimento están un poco alejadas de la realidad. Blum y Mitchell (1998) afirmaron que este algoritmo es un caso particular de su propuesta.

En el coentrenamiento, dos clasificadores sencillos (*weak learners*) pueden ayudarse el uno al otro a mejorar su acierto siempre y cuando concurran ciertas condiciones. Los dos clasificadores se entrenan a partir de un pequeño **conjunto anotado** (CA), la *semilla*, y clasifican un **conjunto no anotado** (CNA). De estas dos clasificaciones, cada uno elige los ejemplos que considera más fiables y los incorpora al conjunto anotado para volver a entrenar y clasificar en un proceso iterativo que termina según unos criterios preestablecidos.

A medida que se ejecutan las iteraciones, el CA se va haciendo mayor con las contribuciones de cada clasificador. Así el clasificador que llamaremos h_1 utiliza en la siguiente iteración los ejemplos que ha clasificado el segundo clasificador, h_2, y viceversa. De esta forma se espera que se reduzca el error cometido por cada clasificador en una tasa significativa.

Los clasificadores son diferentes porque utilizan dos vistas distintas de los mismos datos para aprender. El término 'vista' podemos asimilarlo a una selección de atributos, es decir cada clasificador entrena con conjuntos distintos de atributos pero sobre los mismos ejemplos. El problema que Blum y Mitchell mostraban era la clasificación de páginas web en dos posibles clases, páginas personales del personal docente y las que no lo son. Establecen, pues, una partición binaria de los datos de entrenamiento, dos únicas clases, ejemplos 'positivos' y 'negativos'. Un clasificador aprendía a partir del conjunto de palabras que formaban el texto de la página y el otro del conjunto de palabras que se encontraban dentro de los enlaces de esa página hacia otras.

El coentrenamiento puede aplicarse a problemas de clasificación que cumplan las siguientes condiciones:

1. cada vista de los datos debe ser suficiente por si misma para realizar la tarea
2. los ejemplos anotados por coentrenamiento obtienen esa misma clase con cualquiera de las dos vistas
3. las vistas son condicionalmente independientes dada la clase

El algoritmo propuesto por Blum y Mitchell es el siguiente[1]:

```
CA: es el conjunto anotado (semilla)
    con dos únicas clases: positivo y negativo
CNA: es el conjunto no anotado
p: es la cantidad de anotaciones positivas por iteración y
    clasificador
n: es la de negativas

Crear un subconjunto CNA' de ejemplos seleccionados
    aleatoriamente del CNA

Para k iteraciones
    Entrenar un clasificador h1 con CA (vista 1)
    Entrenar un clasificador h2 con CA (vista 2)
    Etiquetar p ejemplos positivos y n negativos
        del CNA' con h1
    Etiquetar p ejemplos positivos y n negativos
        del CNA' con h2
    Añadir estos ejemplos al CA
    Rellenar CNA' con 2p + 2n ejemplos
        elegidos aleatoriamente de CNA
```

Por ser foco de atención y esperanza de muchos investigadores en métodos supervisados, ávidos de aprovechar una ingente cantidad de información no anotada, esta primera propuesta ha tenido bastantes revisiones y matizaciones (véase la subsección 6.2.1). Además de ser un método con unas restricciones muy fuertes, el problema del coentrenamiento radica en la rápida degradación del acierto cuando se acumulan los errores de clasificación.

Después de una breve muestra de publicaciones posteriores a la aparición del artículo de Blum y Mitchell, se propondrá un método de aprendizaje iniciado en una semilla que recoge varias de las ideas del coentrenamiento y que aspira a minimizar los errores. La evaluación

1 La decisión de no utilizar el CNA al completo es simplemente una cuestión de eficiencia, suponiendo que el tamaño del CNA es demasiado grande y ralentizaría en demasía la clasificación en cada iteración. Así, eligen aleatoriamente un subconjunto que se rellena en cada iteración manteniendo un tamaño constante. Se supone que no supone una merma del resultado final, aunque no aclaran el impacto de la selección aleatoria de la semilla.

del mismo desde varios puntos de vista, y con varios corpus distintos, cierra el capítulo, incluyendo un estudio del comportamiento del reentrenamiento en el corpus DSO.

6.2.1 Aplicaciones y mejoras del coentrenamiento

El sistema *CoBoost* (Collins y Singer, 1999) se basa en un algoritmo de *boosting* que incorpora a la función de optimización las ideas del coentrenamiento, aplicado en este caso al reconocimiento de entidades.

Pierce y Cardie (2001) defienden una revisión manual del proceso en una iteración intermedia (*corrected co-training*), demostrando que la corrección en ese momento de todas o algunas de las etiquetas erróneamente asignadas por coentrenamiento aumenta significativamente el acierto en la última iteración.

Ghani (2001), aplicado a la clasificación de textos, plantea que los problemas de clasificación con muchas clases pueden ser divididos, con éxito, en muchos problemas de binarios. Él combina *Error Correcting Output Coding* (Dietterich y Bakiri, 1995) y coentrenamiento.

Abney (2002) considera la independencia condicional de las vistas demasiado fuerte ya que no se encuentra con facilidad en casos reales, por lo que propone un algoritmo que pretende demostrar que la relajación de este presupuesto es posible y satisfactoria. Para ello propone una modificación del algoritmo a la que da el nombre de *Greedy Agreement*. También considera, en contra de Blum y Mitchell (1998), que el método de Yarowsky (1995) no es un caso especial de coentrenamiento.

Nigam y Ghani (2000a,b) describieron las implicaciones del uso del coentrenamiento comparado y combinado con EM (*expectation-maximization*). Arguyen que, efectivamente, cuando los datos pueden ser vistos dualmente de forma natural, métodos que hagan uso de de esa división de atributos obtienen mejores resultados que aquellos que no la usan. Incluso si esa doble visión no es natural sino forzada por el coentrenamiento los beneficios son cuantificables.

Sin embargo, Ng y Cardie (2003) muestran resultados en los que, para esas tareas sin una división clara de atributos, una variación de

EM (FS-EM) y un algoritmo débilmente supervisado monovista, denominado por ellos autoentrenamiento[2] (*self-training*), obtenían mejores resultados que el coentrenamiento.

Steedman et al. (2003) combinan dos aproximaciones a la adquisición automática de ejemplos, la *selección de ejemplos* y el coentrenamiento. La selección de ejemplos (*sample selection*) (Thompson et al., 1999; Hwa, 2000; Tang et al., 2002) es una variante del aprendizaje activo (*active learning*) (Cohn et al., 1994) y consiste en la búsqueda de ejemplos no anotados con una alta utilidad de aprendizaje (que más probablemente mejorarán el clasificador). El coentrenamiento busca los ejemplos más fiablemente anotados. La aplicación de esta técnica es la construcción de analizadores sintácticos basados en la estadística de corpus, como desarrollo a partir del trabajo de Sarkar (2001).

6.3 Un nuevo método incremental

Vamos a suponer una palabra w con tres sentidos, $s1$, $s2$ y $s3$. Supongamos, así mismo, un corpus anotado (CA) que servirá como conjunto de ejemplos para el aprendizaje, y un corpus no anotado (CNA) que se ha de clasificar.

Un clasificador "normal" sería entrenado, vendría definido por un aprendizaje desde el CA y, posteriormente, sería utilizado para clasificar los contextos presentes en el CNA en las tres clases posibles, en cualquiera de los sentidos de la palabra w. Digámoslo así, se aprende todo de una vez y se obtiene el clasificador que clasifica todo el CNA en una única ejecución.

Podríamos plantearnos otra forma de trabajar, que en vez de un único clasificador, sucesivos clasificadores aprovecharan la información suministrada por el anterior, un proceso iterativo que fuera mejorando el clasificador poco a poco. Supongamos que aprendemos del CA pero sólo clasificamos una pequeña porción del CNA, no por completo. La intención de clasificar tan sólo unos cuantos contextos

2 También utilizan varios clasificadores que han de ponerse de acuerdo para asignar una etiqueta a un ejemplo concreto; nuestro reentrenamiento hace lo contrario, sólo uno de ellos debe proponer la etiqueta.

es que, por el criterio que sea, aseguremos que la precisión es muy alta, aunque la cobertura sea mínima.

Pero esta pequeña porción de contextos clasificados en el primer aprendizaje son añadidos al CA previo, y pasan a ser ejemplos de un corpus de aprendizaje mayor que será el material de un segundo entrenamiento. A este nuevo corpus, el CA más los pocos ejemplos obtenidos del CNA lo llamaremos CA2, y al CNA del que eliminamos justo esos contextos ya etiquetados en la primera clasificación lo llamaremos CNA-2.

Tras el segundo aprendizaje, el clasificador que obtenemos está "más informado", tiene más ejemplos de los que aprender, el CA2. Si procesamos el CNA-2 con este segundo clasificador y seleccionamos unas cuantas clasificaciones más, podemos obtener de la misma forma que antes un CA3 y un CNA-3, y construir un tercer clasificador aún más informado.

Y así, hasta llegar a CAn y CNA-n, donde incluso puede que el CNA esté vacío porque todos los ejemplos han sido clasificados y han pasado al CA.

Esta forma de trabajar se basa en la esperanza de que los sucesivos clasificadores son capaces de detectar los contextos cuyas etiquetas son muy fiables, y que gracias a que un clasificador es mejor que el anterior, cada vez va a ser más fácil etiquetar los contextos que van quedando en los CNA, así hasta clasificarlos todos.

El problema, ya lo hemos dicho, es como filtrar qué etiquetas son fiables y cuáles no. Debemos encontrar un proceso que automáticamente rechace las clasificaciones de las que no se puede asegurar su corrección. Esta es la motivación de nuestro método, que pasamos a describir a continuación.

6.3.1 Un ejemplo

Sea la misma palabra w, con sus tres sentidos $s1$, $s2$ y $s3$, y el CA y el CNA que le son propios. Dividamos el aprendizaje en sentidos, esto es, construyamos tres clasificadores, cada uno correspondiente a un sentido: $c1$, $c2$ y $c3$. El objeto de cada uno de estos clasificadores es decidir si un contexto pertenece o no a su clase, es decir, $c1$

clasificará los contextos del CNA en "sí $s1$" y "no $s1$", al igual que los otros dos con sus propios sentidos.

Este proceder necesita de un paso previo: reetiquetar el CA para cada clasificador. El corpus de aprendizaje del clasificador $c1$ a construir ha de contener ejemplos positivos y negativos del sentido $s1$. Fácilmente, todos los ejemplos anotados como $s1$ son positivos y los etiquetados como $s2$ o $s3$ son negativos. En realidad, "triplicamos" el CA, puesto que los otros dos sentidos tendrán sus propios conjuntos de entrenamiento con ejemplos positivos y negativos.

Corpus originales

CA	CNA
e_1.s1	x_1
e_2.s2	x_2
e_3.s3	x_3

Nuevos corpus

CA.s1	CA.s2	CA.s3	CNA
e_1.SÍ	e_1.NO	e_1.NO	x_1
e_2.NO	e_2.SÍ	e_2.NO	x_2
e_3.NO	e_3.NO	e_3.SÍ	x_3

Supongamos que $e_1.s1$, $e_2.s2$ y $e_3.s3$ son los tres ejemplos del CA, cada uno etiquetado con un sentido, y que tres contextos del CNA, x_1, x_2 y x_3, tres contextos del CNA, son la entrada a los tres clasificadores y que el resultado obtenido es el siguiente:

Consenso

	c1	c2	c3
x_1	SÍ	NO	SÍ
x_2	NO	NO	NO
x_3	**SÍ**	NO	NO

El contexto x_1 ha sido clasificado como positivo por el clasificador del sentido $s1$ y por el del $s3$, el contexto x_2 no ha obtenido clasificaciones positivas en ninguno de los tres, y el tercero, x_3 sólo pertenece al sentido $s1$ por que sólo $c1$ lo ha etiquetado como positivo.

Un criterio de fiabilidad va a ser que sólo aquellos contextos de los que estamos seguros de su etiqueta son susceptibles de pasar del CNA al CA. En este caso, x_1 es ambiguo puesto que dos sentidos lo reclaman como "suyo", y x_2 no pertenece a ninguno. Los tres clasificadores sí están de acuerdo en que x_3 es del sentido $s1$, hay consenso entre ellos.

El siguiente paso sería trasvasar los contextos consensuados del CNA al CA, pero tenemos tres CA distintos. La operación es obvia, x_3 pasa como positivo al corpus del sentido $s3$, y como negativo a los otros, eliminándolo del CNA. Los corpus de entrenamiento con un ejemplo más, y el no anotado con uno menos. El proceso se repite hasta que no se consigue clasificar positivamente ningún contexto más del CNA.

Trasvase

CA.s1	CA.s2	CA.s3	CNA
e_1.SÍ	e_1.NO	e_1.NO	x_1
e_2.NO	e_2.SÍ	e_2.NO	x_2
e_3.NO	e_3.NO	e_3.SÍ	
x_3.**SÍ**	x_3.**NO**	x_3.**NO**	

Compliquemos un poco más el trabajo de los clasificadores. Hasta ahora no hemos dicho nada de la información utilizada en los aprendizajes binarios. Se entiende que para todos se ha elegido un conjunto de atributos con los que caracterizar los ejemplos. Supongamos que establecemos, no uno, sino dos clasificadores por sentido: ($c11$, $c12$), ($c21$, $c22$) y ($c31$, $c32$), de tal forma que los clasificadores $cx1$ son entrenados con el conjunto de atributos A_1, y los $cx2$ con el conjunto A_2.

Por aclarar, y como ejemplo, supongamos que $A_1 = LWS$, las palabras y lemas alrededor de w, y $A_2 = BC$, las composiciones de 2 palabras y lemas alrededor, también, de w.

La clasificación de los tres contextos se hace ahora en dos pasos: primero deben estar de acuerdo los dos clasificadores de cada sentido, y luego debe haber consenso entre los tres sentidos. El siguiente ejemplo comienza, otra vez, desde los corpus originales.

Consenso de sentido

	c11	c12	c21	c22	c31	c32
x_1	SÍ	NO	SÍ	NO	**SÍ**	**SÍ**
x_2	NO	NO	SÍ	NO	NO	NO
x_3	**SÍ**	**SÍ**	NO	SÍ	NO	NO

Ahora cada sentido debe proponer sus positivos por acuerdo entre los dos clasificadores que trabajan para él. De esta forma, el contexto x_1 es positivo para el sentido $s3$ porque $c31$ y $c32$ están de acuerdo en ello, mientras que las otras parejas de clasificadores no lo han hecho. El resultado es que cada sentido propone sus positivos tal y como se muestra a continuación:

Consenso entre sentidos

	c1	c2	c3
x_1	NO	NO	**SÍ**
x_2	NO	NO	NO
x_3	**SÍ**	NO	NO

Ahora tenemos dos contextos clasificados, x_1 para $s3$, que pasará al corpus de aprendizaje como positivo para este sentido y como negativo para los otros, y x_3 para $s1$, que de igual forma será positivo en este sentido y negativo en los otros.

Trasvase

CA.s1	CA.s2	CA.s3	CNA
CA.s1	CA.s2	CA.s3	CNA
e_1.SÍ	e_1.NO	e_1.NO	
e_2.NO	e_2.SÍ	e_2.NO	x_2
e_3.NO	e_3.NO	e_3.SÍ	
x_1.NO	x_1.NO	x_1.SÍ	
x_3.SÍ	x_3.NO	x_3.NO	

Este proceso se repite, nuevamente, hasta que no se consiguen nuevos ejemplos positivos para ningún sentido, o por cualquier otro criterio de parada. En realidad, el método que estamos proponiendo es más complejo puesto que también se establecen valores de

probabilidad mínimos para dar una respuesta positiva a un contexto, pero esto lo contamos ya en la siguiente sección.

El objetivo siempre es dificultar la clasificación errónea, y se espera de este sistema de acuerdos a dos niveles el asegurar la corrección de las etiquetas.

6.4 Método propuesto

Partiendo de estas ideas se va a proponer un método de semilla cuyo objetivo es obtener ejemplos de corpus no anotados con la máxima fiabilidad de etiquetado posible. A este método lo llamamos **reentrenamiento**. Todo lo que a continuación se expone se entiende que se refiere a una palabra concreta.

Se parte de un corpus anotado (CA) y otro no anotado (CNA), ambos de una palabra con c clases (en nuestro caso, sentidos de WN).

El reentrenamiento se ha planteado como un proceso iterativo de incorporación de nuevos ejemplos al CA. La elección de estos nuevos ejemplos se hace aprendiendo del CA y clasificando el CNA.

El aprendizaje se divide en tantos aprendizajes binarios como clases tenga la palabra. Del CA se obtienen c conjuntos de entrenamiento, de forma que cada uno contiene ejemplos positivos y negativos, siendo positivos los anotados como pertenecientes a una clase y negativos los que son positivos para las otras clases.

De cada conjunto de entrenamiento, se obtienen 2 clasificadores binarios para cada clase utilizando conjuntos de atributos distintos (en total, tendremos $2 \times c$ clasificadores binarios) y se clasifica el CNA con ambos. De cada par de clasificaciones, se produce una **propuesta parcial**, una clasificación única por clase. Tendremos, pues, c propuestas parciales.

El siguiente paso consiste en comparar las propuestas binarias parciales para obtener una única **propuesta común** de nuevos ejemplos positivos. Se propone un consenso excluyente de tal forma que los ejemplos considerados positivos por más de una clase no se tienen en cuenta. La propuesta común consiste en todos aquellos ejemplos que sólo son positivos para una única clase.

Los ejemplos de la propuesta común se incorporan a su conjunto de entrenamiento correspondiente y se reconstruyen los clasificadores binarios para volver a clasificar el CNA. Se repite el proceso hasta un número de iteraciones suficiente o predeterminado.

Dicho de otra forma, el reentrenamiento de una palabra se basa en varios entrenamientos parciales binarios, varias clasificaciones binarias y la combinación de éstas en una propuesta única que alimenta a la siguiente iteración, siguiendo el siguiente esquema procedural:

Generación de semillas: para cada clase de la palabra a procesar, se genera una semilla con ejemplos positivos y negativos, en principio utilizando todos los ejemplos disponibles en el CA. Estas semillas son los conjuntos de entrenamiento particulares de cada clase, y a los que se irán incorporando nuevos ejemplos positivos y negativos.

Entrenamientos binarios (propuestas parciales): para cada clase de la palabra, se activan dos entrenadores (programas de aprendizaje basados en máxima entropía y diferentes conjuntos de atributos) y se obtienen dos clasificaciones del CNA. Estos clasificadores, dos por sentido, trabajan con 2 clases únicamente, positivo y negativo (es sentido x, o no lo es).

A partir de estas dos clasificaciones, se realiza una propuesta parcial por consenso de incorporación de ejemplos para cada uno de los sentidos: para un sentido x, si sus dos clasificadores están de acuerdo en que un contexto es positivo éste se propone como ejemplo candidato. Cada sentido tendrá su propia lista de candidatos, su propuesta parcial.

Consenso entre sentidos (propuesta combinada): la elección de los ejemplos que se añadirán a los conjuntos de entrenamiento pasa por la verificación conjunta de todas las propuestas. Para cada candidato se comprueba que sólo una de las propuestas lo haya anotado como positivo; en caso contrario se entiende que el contexto es ambiguo.

Regeneración de los conjuntos de entrenamiento: a partir de la propuesta combinada, cada clase incorpora sus

ejemplos positivos a su conjunto de entrenamiento particular y el resto como negativos.

La siguiente iteración comienza en "entrenamientos binarios", y el proceso finaliza en un número de iteraciones establecido por parámetro, o cuando ya no se consiguen nuevos ejemplos desde el CNA.

Una representación gráfica de este proceso puede verse en la figura 6.1.

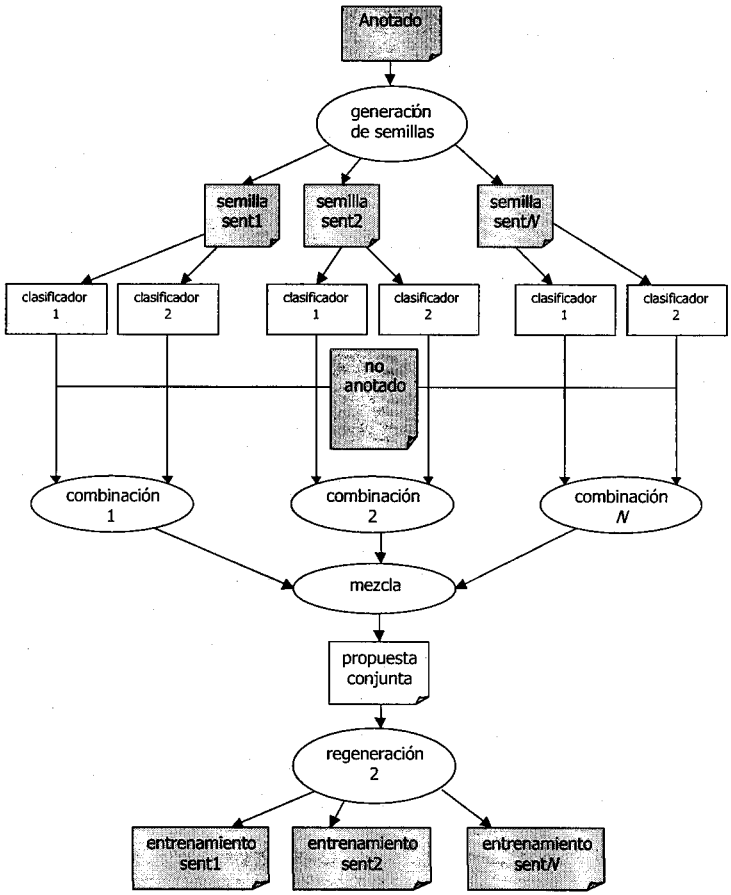

Figura 6.1. Esquema de reentrenamiento

6.5 Criterios de calidad de la clasificación

Se trata de establecer qué ejemplos del CNA se incorporan a los conjuntos de entrenamiento como positivos. Se definirán detalladamente los filtros apuntados anteriormente, los que se pretende que aseguren la correcta clasificación de algunos ejemplos,los que van a ser incorporados al posterior aprendizaje.

Para las propuestas parciales: en la fase de confección de las propuestas parciales, las que resultan del proceso de cada sentido, el primer criterio es que los dos clasificadores entrenados para un sentido en particular den como positivo un ejemplo. Si alguno de los dos, o los dos, lo clasifican como negativo, no será propuesto.

Además, podemos establecer un **umbral** de confianza basado en la *diferencia entre la probabilidad de que sea positivo* y la probabilidad de que sea negativo. Supongamos:

i= $\{1|2\}$, índice de clasificador binario
e: es un ejemplo no anotado
u: es el umbral
p_i : la probabilidad de que e sea positivo en el clasificador i
n_i: la probabilidad de que e sea negativo en el clasificador i
c: es la clase de los dos clasificadores binarios

e es positivo para la clase c si $p_1 - n_1 > u$ y $p_2 - n_2 > u$

Para la propuesta común: después de que para cada sentido se propongan los candidatos a ser etiquetados, el primer paso para la confección de la propuesta combinada es comprobar los ejemplos que sólo son positivos en una de las propuestas parciales, los que sólo son "reclamados" por un único sentido. Los demás, aquellos ejemplos que son propuestos como pertenecientes a más de un sentido, son rechazados.

Una vez que se ha hecho esta combinación de las propuestas parciales, se eligen los ejemplos que obtienen el mayor valor de probabilidad dentro de su clase. Es como si, para cada clase, ordenáramos todos "sus" ejemplos,

de mayor a menor probabilidad y sólo escogiéramos el primero (o los primeros si hay varios que obtienen la máxima probabilidad de su clase). Puesto que se han utilizado dos clasificadores para cada propuesta parcial, la suma de las probabilidades ($p_1 + p_2$) es el criterio de ordenación.

6.5.1 Aumento de la cobertura absoluta

Reinterpretación de la medida

En primer lugar, defenderemos que los datos de cobertura o cobertura absoluta no son prioritarios en este capítulo, o al menos su interpretación ha de ser otra forzosamente, nuestra meta es la precisión. Si asumimos que el CNA es infinito (o como poco, inmenso, lo que no es difícil de imaginar si establecemos *internet* como CNA), la cobertura y la cobertura absoluta tienden a cero, clasifiquemos lo que clasifiquemos.

En realidad, los datos que se exponen en secciones posteriores se basan en un CNA finito, y por eso se ofrece este dato, pero lo que estamos midiendo realmente es la rapidez con que el método incorpora nuevos ejemplos al CA, no la virtud de dar respuesta a la mayor cantidad posible de intentos de clasificación. En nuestro reentrenamiento, damos prioridad a la seguridad de que lo clasificado lo está correctamente y, dadas estas condiciones, una cobertura alta simplemente indicaría que podemos conseguir un corpus sensiblemente mayor que el CA inicial.

Pensemos que si de 1.000 ejemplos no etiquetados conseguimos 10 anotaciones, la cobertura sería de un 1 %, pero un centavo de millón son 10.000 nuevas muestras. El problema ya no es tanto cubrir todos los ejemplos de los que disponemos sino asegurarnos de que se obtiene una cantidad razonable o suficiente de ejemplos de bastantes sentidos, y no de sólo uno.

Es obvio que si la cobertura es extremadamente baja, el número de ejemplos es mínimo, y la posibilidad de que sean de sentidos diferentes es menor que si la cobertura fuera alta (que tampoco nos asegura esa distribución deseable). Por eso, a los datos de precisión y cobertura se añade la **cobertura de sentidos**, esto es, para cuántos sentidos diferentes se han obtenido nuevos ejemplos.

Método de incremento de la cobertura absoluta

El utilizar un umbral como se detalla en la sección anterior puede incluso no conseguir clasificaciones. Una forma de aumentar esa cobertura es disminuir el umbral cuando el método ya no consigue clasificar más ejemplos. Podemos empezar por 0.9 y, a medida que se agota el proceso ir bajando décima a décima el **umbral de inicio** hasta llegar a 0.0, o a un valor establecido por parámetro, el **límite inferior**. Esta disminución puede ser fija para todos los sentidos o variable, en cuyo caso penalizaremos los sentidos con más "éxito" frente a los otros. Quiere esto decir que si un sentido ha conseguido más ejemplos nuevos que los otros, la disminución sólo se produce en éstos en un intento de facilitarles la obtención de nuevos ejemplos. El algoritmo expuesto en la figura 6.2 describe cómo es este proceso de disminución de umbrales.

```
si no hay ejemplos nuevos
   x = sentido con más ejemplos nuevos
   para todo sentido y <> x
        si y.umbral > límite_inferior
             y.umbral -= 0.1
   si ninguno disminuyó su umbral
        si x.umbral > límite_inferior
            x.umbral -= 0.1
            para todo sentido y
                y.umbral = x.umbral
    si ninguno disminuyó su umbral
        TERMINAR REENTRENAMIENTO
    si no
        continuar
```

Figura 6.2. Algoritmo de modificación de umbrales

6.5.2 Diferencias entre reentrenamiento y coentrenamiento

Nuestro reentrenamiento es en realidad un conjunto de coentrenamientos si atendemos a que utilizamos dos clasificadores binarios por cada uno de los sentidos de la palabra. No obstante, hay diferencias en estos aprendizajes parciales que los alejan de la propuesta inicial de coentrenamiento.

En el coentrenamiento definido por Blum y Mitchell (1998) se mantiene la proporción entre positivos y negativos, respetando las frecuencias de la propia semilla. Nuestro reentrenamiento no tiene en cuenta esta característica, incorpora todos los ejemplos que obtienen la máxima probabilidad si no hay colisiones entre clases (propuesta común).

Lo cierto es que un ejemplo que no sea elegido en una iteración por este límite, por lo general, acaba siendo elegido en las siguientes. De hecho, el aspecto de la proporción de negativos y positivos no se ha tenido en cuenta dado que, por el filtro de la propuesta combinada, ciertos sentidos tienen dificultades para ganar positivos y, simplemente, no se puede satisfacer la proporción. Así, el coentrenamiento está definido para un número finito de iteraciones, mientras que nuestro reentrenamiento puede detenerse antes de la última iteración porque no es capaz de clasificar ningún contexto con la suficiente seguridad.

Tampoco se procesan los negativos como tales, simplemente se eligen de entre los positivos de otras clases, lo que puede dar lugar a menos negativos de los esperados (por ejemplo, que sólo se incorpore un positivo a la siguiente iteración y que una clase espere dos o más negativos). En el coentrenamiento se eligen tanto positivos como negativos en la proporción definida. En nuestro reentrenamiento, la elección de negativos es más complicada debido a la concurrencia de varias propuestas de positivos y negativos en la misma iteración.

El coentrenamiento parte de una semilla cuyos miembros son elegidos aleatoriamente, mientras nosotros utilizamos todo el CA. En las pruebas que realizamos, previas al estudio que aquí se va a desarrollar, la elección aleatoria nos daba resultados muy dispares en diferentes ejecuciones, por lo que la descartamos.

La única razón por la que se nos ocurre que un ejemplo no deba ser parte de la semilla tiene que ver con el aprendizaje activo y es que ese ejemplo en particular sea dañino por si mismo, que no sea adecuado para el entrenamiento. No hemos realizado estudio alguno respecto a este asunto.

Otro aspecto no suficientemente aclarado por Blum y Mitchell (1998), ya que dan a entender que sus dos clasificadores podrían trabajar en paralelo, es qué ocurre cuando ambos clasificadores pro-

ponen el mismo ejemplo como positivo, aspecto éste que está asumido en nuestro reentrenamiento.

Acercando el método al coentrenamiento

Existe una posibilidad obvia aplicable a nuestro reentrenamiento que es no realizar dos aprendizajes para cada sentido sino entrenamientos consecutivos pero alternando los conjuntos de atributos. De esta forma, nos acercamos más aún al coentrenamiento definido por Blum y Mitchell (1998).

Por ejemplo, si suponemos dos conjuntos de atributos, la primera iteración haría un único aprendizaje para cada sentido con uno de ellos y clasificaría, incorporándose los positivos a los conjuntos de entrenamiento. La siguiente iteración aprendería con los nuevos conjuntos de entrenamiento y con el otro grupo de atributos y volvería a clasificar, y así sucesivamente.

Se hará un pequeña prueba para comprobar que siguiendo este esquema de proceso se obtienen peores resultados que con el reentrenamiento.

6.5.3 Otros posibles parámetros

Se pueden establecer varias estrategias basadas en matizaciones referidas principalmente al tratamiento global de la palabra o particularizado para cada sentido.

Se pueden establecer diferencias entre los sentidos asignándoles diferentes umbrales de inicio y límite inferior a cada uno, en la proporción de positivos y negativos, e incluso en los conjuntos de atributos de aprendizaje. El problema reside en encontrar una pauta y disponer de la información suficiente como para establecer estas diferencias.

Por ejemplo, la intuición nos dice que los sentidos menos frecuentes deberían incorporar más ejemplos negativos que los más frecuentes pero, generalmente, también son los sentidos más difíciles de detectar: el proceso iterativo tendería a incrementar el peso de los ejemplos negativos y dificultaría aún más la clasificación de sus positivos. Esta tendencia se podría contrarrestar con un umbral más bajo

para los menos frecuentes pero corremos el riesgo de que se generen errores debido a que no haya competencia y sólo ellos propongan positivos.

No hablemos ya de distintos conjuntos de atributos, diferentes para cada sentido de la misma palabra. Aunque pueda parecer lo contrario, hay una fuerte interacción entre los sentidos en el proceso ya que constantemente compiten entre ellos por clasificar los ejemplos como suyos, y esta competencia es precisamente la base del método.

En realidad, en este trabajo no se han abordado tales cuestiones y hemos pospuesto la investigación de su impacto por la complejidad que añade al propio sistema.

6.5.4 Terminología

Con el fin de evitar confusiones, y también como resumen de las secciones anteriores, se recuerdan y establecen aquí las siguientes definiciones:

sentido: cualquiera de los sentidos del corpus.

clase: cualquiera de las categorías en que un contexto puede ser anotado por un clasificador; en nuestro caso las clases coinciden con los sentidos.

positivo: ejemplo anotado como positivo para un sentido concreto, que podría etiquetarse con ese sentido.

negativo: ejemplo anotado como negativo para un sentido concreto, que su etiqueta podría no ser ese sentido.

precisión: cantidad de ejemplos correctamente clasificados dividido por la cantidad de ejemplos anotados.

cobertura: o **cobertura absoluta**, cantidad de ejemplos clasificados dividido por la cantidad de ejemplos que tiene el corpus. Salvo que se diga expresamente lo contrario, por 'cobertura', en este capítulo, vamos a referirnos a 'cobertura absoluta'.[3]

3 Estas medidas fueron definidas, de forma general, en la sección 2.2.

promedios de precisión y cobertura: los valores promediados que se van a mostrar en las sucesivas descripciones de experimentos se han calculado sumando todos los aciertos, todas las respuestas y todos los ejemplos de todas las palabras, dividiendo las dos primeras sumas por la tercera.

cobertura de sentidos: es la *ratio* de sentidos cubiertos (cantidad de sentidos que han conseguido algún ejemplo de la clasificación del CNA dividido por la cantidad de sentidos en el corpus). Por ejemplo, si 'art' tiene 4 sentidos y todo los ejemplos que el sistema ha clasificado lo han sido como sentido 1, el valor obtenido es del 25 %. Se entiende como óptimo un 100 %, ya que el método ha sido capaz de clasificar al menos un ejemplo de cada sentido. Cuando esta medida se refiere a un conjunto de palabras, el cálculo se realiza dividiendo la suma de los sentidos reconocidos por la suma de sentidos en el corpus.

atributos: o grupos de atributos, hace referencia a la especial agrupación de atributos utilizados en el aprendizaje, tal y como se explicó en la subsección 4.3.2.

umbral de inicio: es el valor del umbral que se pasa por parámetro al proceso, tal y como se definió en la subsección 6.5.1. Como son valores de probabilidad, sus posibles valores están entre 0,0 y 1,0.

variación de umbral: viene determinado por el límite de umbral ya mencionado, y es el decremento máximo que se puede aplicar al umbral de inicio para el aumento de cobertura, como también se definió en la subsección 6.5.1.

umbral: en lo sucesivo, al dar valores de umbral nos estaremos refiriendo a la combinación de los dos anteriores en forma de parámetro. Por ejemplo, 0.9-0.4, indica un umbral de inicio de 0,9 que puede disminuir hasta 0,5 si las iteraciones no consiguen ningún ejemplo nuevo.

6.6 Introducción a las pruebas experimentales sobre reentrenamiento

Varios son los experimentos que hemos definido para la evaluación del reentrenamiento. Las diferencias entre unos y otros van desde la hipótesis que se quiere demostrar empíricamente hasta los conjuntos de entrenamiento utilizados.

En primer lugar, la sección 6.7 pretende comprobar si la combinación de información anotada y no anotada contribuye a mejorar los resultados de nuestro sistema WSD basado en los MME. Para ello, utilizando el conjunto de aprendizaje de la muestra léxica en inglés de SENSEVAL-2 como CA, y un CNA extraído de la colección en inglés del TREC 2002 (Voorhees y Buckland, 2002), se aplica el reentrenamiento para conseguir más ejemplos de entrenamiento. Finalmente, este aprendizaje se evalúa contra el conjunto de test de la misma tarea de SENSEVAL-2.

El problema de este experimento es que no tendremos datos de si el reentrenamiento ha anotado bien o mal, tan sólo veremos indicios al evaluar sobre el conjunto de test, si la precisión aumenta o disminuye al compararla con la obtenida de un aprendizaje estándar con el corpus de entrenamiento original.

Por eso, en la sección 6.8 el objetivo es determinar la validez del método, probando primero con el corpus *Interest* si la anotación es correcta y, posteriormente, con el corpus DSO.

Con la evaluación sobre el DSO se determinará si los filtros establecidos por el reentrenamiento son efectivos, esto es, dos clasificadores binarios diferentes por sentido, la configuración de umbrales, y la mezcla o consenso entre las propuestas parciales de cada sentido. También se compara con una adaptación del coentrenamiento para comprobar si nuestro algoritmo es más ventajoso, más preciso.

Una vez que tengamos estos datos, en la sección 6.9 se propondrán varias estrategias de selección de grupos de atributos para la configuración del reentrenamiento. Estas estrategias se basan en un estudio previo del corpus de entrenamiento con el objeto de calcular los atributos más precisos globalmente y por palabra, incluso con información dependiente de los sentidos. En esta ocasión se usarán

los datos de entrenamiento y test de la muestra léxica en español de SENSEVAL-2.

En todos los casos se busca una mejora en la precisión aunque sea a costa de la cobertura, pero también intentando asegurar una cobertura de sentidos mínima, es decir, que el sistema detecte ejemplos para la mayoría de los sentidos presentes en el corpus.

6.7 Prueba inicial de WSD con corpus aumentados por reentrenamiento

Los primeros datos sobre reentrenamiento los damos en esta prueba. El experimento consiste en comparar la desambiguación de algunas palabras con y sin reentrenamiento. Los parámetros del experimento se muestran en el cuadro 6.1.

corpus	
CA:	SENSEVAL-2 *English lexical sample*
CNA:	TREC (2002): *Associated Press*
TEST:	SENSEVAL-2 *English lexical sample*
	17 nombres seleccionados
reentrenamiento	
atributos:	LWS - BCDr
iteraciones:	1000
umbral:	0.9-0.9
entrenamiento y clasificación	
	LWS
	BCDr
	LBDMr
	LBWCDM
	sk10

Cuadro 6.1. Datos de la prueba de desambiguación sobre SENSEVAL-2 con y sin reentrenamiento

Se han utilizado los ejemplos de entrenamiento y prueba de la muestra léxica en inglés del SENSEVAL-2. En particular se han escogido palabras cuyos resultados en una clasificación "normal" son discretos con nuestro sistema ME (tasas de acierto del 60 % o menos), para las que se espera que, efectivamente, un corpus más pobla-

do ayude a aumentar la precisión. Las palabras elegidas son todas nombres.

El CNA está compuesto por algunos artículos no etiquetados del TREC 2002 (del conjunto de la *Associated Press*). Así, para estas palabras y mediante el proceso de reentrenamiento antes descrito, al CA se le añaden tantos ejemplos como se pueda extraídos de esta colección.

Todos los datos que se van a mostrar aquí se han calculado a partir de las ejecuciones cuyos resultados se muestran en el cuadro 6.2.

Cada 100 iteraciones se ha generado un nuevo corpus que ha sido utilizado para aprender y ser evaluado contra el conjunto de test. Así, disponemos de 11 resultados medidos por F1[4]. El primero, *BASE*, es el resultado de entrenar con el CA original de SENSEVAL-2. A continuación, los 10 valores siguientes se corresponden a los aprendizajes con cada uno de los aumentos de ese corpus inicial mediante reentrenamiento. Ciertos nombres han visto detenido el proceso antes de la iteración 1000, por lo que no tienen valores desde la columna correspondiente.

Al final, tenemos un corpus de entrenamiento aumentado, es decir, al conjunto de aprendizaje original de SENSEVAL-2 le hemos añadido nuevos ejemplos obtenidos automáticamente por reentrenamiento desde un corpus no anotado.

El objetivo es comparar si el incremento de ejemplos por este método aumenta el éxito de la desambiguación. Se han definido 5 conjuntos de atributos para el entrenamiento sobre el nuevo corpus (los que se especifican en el cuadro 6.1: "entrenamiento y clasificación"). Se intenta comprobar si se mejora la precisión para todos ellos.

Los nuevos corpus de entrenamiento provocan resultados dispares según la palabra y el clasificador final elegido. Por ejemplo, para un aprendizaje con los atributos $BCDr$, child obtendría un 50,8 % de F1 con el corpus original (columna *BASE*), mientras que con el corpus incrementado por reentrenamiento hasta 1000 iteraciones

4 En esta prueba sí tiene sentido la cobertura ya que se intenta cubrir el máximo de un número finito de contextos con la máxima precisión; no estamos comprobando directamente si los ejemplos conseguidos por reentrenamiento están correctamente etiquetados o no.

6.7 Prueba inicial de WSD con corpus aumentados por reentrenamiento

palabra	atributos	BASE	100	200	300	400	500	600	700	800	900	1000
art	BCDr	51,1	32,4	32,4	32,4	29,6	33,8	31,0	31,0	31,0	33,8	33,6
	LBDMr	56,3	46,5	46,2	40,6	40,6	39,2	39,2	37,8	35,0	33,6	33,6
	LBWCDM	50,0	45,9	47,2	47,2	41,2	39,4	37,9	37,9	36,4	34,8	34,1
	LWS	58,8	53,2	48,9	45,7	39,4	36,4	36,4	36,4	36,4	35,0	32,2
	sk10	58,3	55,6	50,0	47,2	37,5	33,3	29,2	30,6	30,6	31,9	30,6
bar	BCDr	55,9	55,7	54,8	54,8	53,9	53,7					
	LBDMr	55,2	55,2	53,4	52,6	54,3	53,2					
	LBWCDM	53,5	**54,6**	51,4	**54,3**	**55,0**	**56,3**					
	LWS	56,3	54,5	50,9	52,4	**57,5**	**56,7**					
	sk10	61,5	59,8	59,8	60,7	59,0	58,1					
channel	BCDr	46,6	46,6	**51,1**	45,1	45,1						
	LBDMr	55,2	50,7	52,2	49,3	50,7						
	LBWCDM	58,1	53,8	52,0	52,0	52,8						
	LWS	60,0	55,0	50,0	47,0	45,1						
	sk10	67,2	62,7	53,7	53,7	53,7						
child	BCDr	50,8	49,2	50,8	**52,5**	**54,1**	**54,1**	**54,1**	**55,7**	**57,4**	**55,7**	**54,1**
	LBDMr	60,7	57,4	53,7	53,7	56,9	55,3	55,3	55,3	55,3	55,3	55,3
	LBWCDM	49,0	**49,1**	**50,0**	**51,4**	**49,5**	47,7	50,5	46,4	**49,6**	**49,1**	47,4
	LWS	71,1	65,0	58,5	60,2	61,8	61,8	61,8	56,9	56,9	53,7	53,7
	sk10	64,5	61,3	61,3	61,3	58,1	56,5	56,5	56,5	56,5	54,8	54,8
church	BCDr	53,7	52,5	53,7	**55,7**	53,7	**56,9**	**55,3**	**60,2**	**55,3**	53,7	**55,3**
	LBDMr	55,7	**59,0**	**58,5**	**65,0**	**58,5**	**58,5**	**58,5**	**58,5**	**56,9**	56,9	55,3
	LBWCDM	43,8	**45,3**	**48,6**	**52,3**	**51,8**	**51,3**	**49,6**	**49,6**	**49,1**	**49,1**	**49,1**
	LWS	57,9	57,9	50,8	**62,3**	50,0	55,7	46,8	52,5	46,8	50,4	47,2
	sk10	69,4	58,1	58,1	56,5	53,2	58,1	56,5	51,6	53,2	50,0	48,4
circuit	BCDr	19,6	**35,0**	**35,0**								
	LBDMr	40,4	**44,2**	**44,2**								
	LBWCDM	28,2	**32,2**	**32,2**								
	LWS	44,7	**49,0**	**49,0**								
	sk10	73,1	67,3	71,2								
facility	BCDr	48,1	**51,4**	**50,9**	**49,1**	**50,9**	**49,1**	**49,1**	**49,1**	**49,1**	**49,1**	**49,1**
	LBDMr	49,5	49,5	**52,7**	**54,5**	**52,7**	**50,9**	**50,9**	**50,9**	**50,9**	**50,9**	**50,9**
	LBWCDM	40,4	**41,7**	**44,9**	**49,0**	**49,0**	**46,9**	**46,9**	**46,0**	**46,5**	**48,0**	**46,0**
	LWS	44,4	43,6	41,8	**47,3**	**52,7**	**52,7**	**56,4**	**52,7**	**52,7**	**52,7**	**52,7**
	sk10	62,5	57,1	55,4	55,4	55,4	53,6	51,8	53,6	51,8	51,8	51,8
feeling	BCDr	61,9	59,8	59,8	59,8	55,7	55,7	57,7	55,7	55,7	55,7	55,7
	LBDMr	61,9	57,7	53,6	55,1	53,1	51,0	53,1	53,1	53,1	53,1	53,1
	LBWCDM	53,9	52,7	52,2	49,5	**55,3**	**57,4**	**58,9**	**54,7**	**56,3**	**59,8**	**59,8**
	LWS	52,6	**58,3**	**56,3**	52,1	**55,7**	**53,6**	**53,6**	**55,7**	**53,6**	**57,1**	**57,1**
	sk10	59,2	53,1	55,1	53,1	53,1	53,1	53,1	55,1	55,1	55,1	55,1
grip	BCDr	54,5	**56,6**	54,5								
	LBDMr	54,5	**56,6**	54,5								
	LBWCDM	58,7	**61,7**	57,4								
	LWS	55,9	**62,4**	**63,8**								
	sk10	76,0	72,0	70,0								
material	BCDr	43,5	39,3	39,3	35,9	35,9	35,9	35,9	34,2	34,2	27,4	27,4
	LBDMr	45,2	39,3	37,6	35,9	33,9	33,9	35,6	35,6	35,6	30,5	28,8
	LBWCDM	38,3	29,2	32,3	29,7	28,6	26,7	25,9	25,9	25,9	18,5	18,5
	LWS	49,6	31,6	31,6	27,8	25,9	24,1	24,1	20,7	22,4	22,4	22,4
	sk10	57,6	45,8	40,7	39,0	35,6	35,6	35,6	33,9	33,9	32,2	32,2
mouth	BCDr	47,3	45,5	43,6	47,3	47,3						
	LBDMr	55,4	50,0	50,0	43,9	42,1						
	LBWCDM	51,0	**54,7**	50,5	**53,2**	49,1						
	LWS	51,3	**52,6**	49,1	45,6	45,6						
	sk10	57,9	50,9	45,6	49,1	49,1						
nature	BCDr	46,5	**48,8**	**48,8**	**53,5**	**53,5**	**53,5**	**53,5**	**53,5**	**53,5**	**53,5**	**53,5**
	LBDMr	48,8	**51,2**	**51,2**	**53,5**	**53,5**	**53,5**	**53,5**	**53,5**	**53,5**	**53,5**	**55,8**
	LBWCDM	37,3	**41,0**	**42,5**	**45,0**	**45,0**	**45,0**	**46,9**	**48,8**	**51,2**	**51,2**	**51,2**
	LWS	55,8	52,9	55,2	55,2	52,9	52,9	50,6	50,6	52,9	52,9	52,9
	sk10	45,5	**52,3**	**50,0**	**50,0**	**47,7**	45,5	45,5	45,5	45,5	45,5	45,5
restraint	BCDr	45,2	35,7	38,1	38,1	38,1						
	LBDMr	44,7	42,4	47,1	40,0	40,0						
	LBWCDM	39,4	**43,2**	**50,0**	**51,3**	**53,2**						
	LWS	45,8	**47,1**	**51,2**	**48,3**	**50,6**						
	sk10	65,9	54,5	43,2	47,7	50,0						
sense	BCDr	51,9	**57,1**	**54,5**	**57,1**	**57,1**	**59,7**	**67,5**	**59,7**	44,2	44,2	51,9
	LBDMr	67,5	57,1	57,1	59,7	59,7	62,3	67,5	64,1	48,7	48,7	53,8
	LBWCDM	42,9	36,6	31,0	36,6	39,4	**47,2**	**47,2**	**49,3**	**49,3**	46,6	**46,6**
	LWS	61,5	**64,1**	53,8	51,3	56,4	56,4	53,8	56,4	53,8	53,8	53,8
	sk10	61,5	59,0	51,3	53,8	**64,1**	**64,1**	**64,1**	**64,1**	59,0	59,0	61,5
stress	BCDr	43,8	43,8	41,1	43,8	41,1						
	LBDMr	45,9	**48,6**	45,9	43,2	43,2						
	LBWCDM	40,0	40,0	**41,2**	**43,5**	**48,6**						
	LWS	33,8	**36,1**	**37,8**	**42,7**	**48,0**						
	sk10	44,7	42,1	44,7	**50,0**	**50,0**						

Cuadro 6.2. Resultados de F1 de la evaluación del reentrenamiento

aumentaría ese valor a 54,1 %. Sin embargo, este nuevo corpus no es el mejor que se podría haber obtenido: si el reentrenamiento previo se detiene en la iteración 800, el nuevo corpus permite subir la clasificación final hasta el 57,4 %. La interpretación de estos valores es que hasta la iteración 800 el reentrenamiento ha ido incorporando más ejemplos correctos que incorrectos, pero parece que a partir de este momento los errores adquieren más peso o, simplemente, los ejemplos obtenidos son demasiado distintos del corpus de test.

Para la misma palabra ocurre algo parecido si el aprendizaje se realiza con la selección $LBWCDM$. Sin embargo, los otros aprendizajes obtienen mejor resultado con el corpus original que con el incrementado.

Si examinamos nombre a nombre la máxima F1 obtenida en la evaluación *BASE* por cada uno, vemos que los nuevos corpus obtenidos en la última iteración por reentrenamiento no consiguen mejorar en ningún caso ese valor máximo. Tan sólo con stress se observa ese aumento.

El cuadro 6.3 muestra los resultados finales de evaluar de esta forma las 17 palabras. La columna *iteraciones* informa de cuándo se ha terminado el proceso de reentrenamiento porque no ha podido clasificar más ejemplos[5]. Los otros valores de cada palabra muestran en cuantas de esas pruebas el corpus ampliado ha mejorado los resultados del corpus original (luego el valor máximo es 5).

El cuadro se divide en dos partes, *precisión* y *F1*. En ambos subcuadros nos encontramos con valores *absolutos* calculados con el corpus generado en la última iteración. *Tendencia* es un cálculo por regresión lineal de cuál podría ser el valor de precisión o F1 si el proceso de reentrenamiento no se hubiera detenido hasta las 5000 iteraciones.

Por ejemplo, los datos del nombre art deben interpretarse como sigue: el reentrenamiento se ha detenido en la iteración 1000; el entrenamiento con el nuevo corpus no ha conseguido, en ninguna de las 5 evaluaciones, superar la precisión obtenida con el entrenamiento a partir del corpus original; la tendencia es que aún aumentando el número de iteraciones, esa precisión no va a mejorar; lo mismo se puede decir de F1.

5 Un valor de 400, por ejemplo, indica que el proceso se ha detenido entre la iteración 300 y la 400.

Palabras	Iteraciones	Precisión		F1	
		Absoluto	Tendencia	Absoluto	Tendencia
art	1000	0	0	0	0
authority	1000	0	0	1	0
bar	500	0	2	2	2
channel	400	0	0	0	0
child	1000	1	1	1	1
church	1000	2	3	2	3
circuit	200	4	4	4	4
facility	1000	4	4	4	4
feeling	1000	2	2	2	2
grip	200	3	1	3	1
material	1000	0	0	0	0
mouth	400	1	1	1	1
nature	1000	4	4	4	4
post	1000	4	4	4	4
restraint	400	2	2	2	2
sense	1000	2	2	2	2
stress	400	2	2	3	0

Cuadro 6.3. Palabras: mejoras y tendencias al engrosar el corpus de aprendizaje con reentrenamiento

Por contra, nature, que también ha agotado las 1000 iteraciones del reentrenamiento, supera o iguala en 4 de las pruebas la precisión y el F1 conseguidos con el corpus original y, además, la tendencia es que mantengan esa ventaja.

El cuadro 6.4 es similar al anterior pero esta vez desde el punto de vista de las selecciones de atributos que marcan las 5 pruebas. Aquí se mide la cantidad de palabras que con esa selección ven aumentada su precisión o su F1 o ambas con el nuevo corpus de entrenamiento.

Atributos	Precisión		F1	
	Absoluto	Tendencia	Absoluto	Tendencia
BCDr	9	8	9	8
LBDMr	5	5	5	5
LBWCDM	7	8	10	9
LWS	7	8	8	8
sk10	3	3	3	3

Cuadro 6.4. Selecciones de atributos: mejoras y tendencias al engrosar el corpus de aprendizaje con reentrenamiento

La selección $BCDr$ (una de las utilizadas para el reentrenamiento), consigue que 9 de las 17 palabras elegidas mejoren su precisión,

pero sólo 8 tienen una tendencia ascendente. Lo mismo ocurre si examinamos su F1.

Tanto LWS como $BCDr$ son dos buenas elecciones, posiblemente por ser las mismas utilizadas para el reentrenamiento, con una ligera ventaja para la primera. Nótese que LWS utiliza las palabras y los lemas del contexto cercano, mientras que $BCDr$ utiliza información más sofisticada (combinaciones de 2 lemas, dependencias y rol sintáctico).

Si atendemos a F1, la mejor selección es, posiblemente por ser la más completa, $LBWCDM$ que contiene a las dos anteriores (a excepción de los atributos r).

Los datos en los que se ha basado esta evaluación se encuentran desglosados a continuación.

Desglose de los resultados del reentrenamiento

Las distribuciones de sentidos de los corpus originales y de la anotación por reentrenamiento se pueden ver en el cuadro 6.5. Éste muestra para cada sentido la cantidad de ejemplos en cada uno de los corpus. La numeración de los sentidos no concuerda con la de WN, es una ordenación teniendo en cuenta que, de las elegidas, la palabra que más sentidos tiene, tiene diez. Hemos llamado *BASE* al CA, el corpus de entrenamiento original de SENSEVAL-2, y *TEST* al de evaluación. Las filas etiquetadas como *REENT* ofrecen datos del corpus aumentado, en la última iteración, por reentrenamiento a partir del *BASE*, teniendo en cuenta que estos valores incluyen los del propio *BASE*.

Por ejemplo, el nombre art tiene 66 ejemplos del sentido 1 en el conjunto *BASE*, y 41 contextos a clasificar en el conjunto *TEST* (conjuntos de entrenamiento y test, respectivamente, de SENSEVAL). El nuevo conjunto de aprendizaje, aumentando el *BASE* por reentrenamiento, *REENT*, tiene 108 ejemplos del sentido 1, incluidos los del *BASE*, es decir, ha generado 42 nuevos ejemplos de este sentido.

Los porcentajes derivados de este cuadro se muestran el cuadro 6.6. La columna *SMF* muestra el sentido más frecuente en el corpus.

palabra	corpus	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	total
art	BASE	66	38	29								133
	REENT	108	1144	58								1310
	TEST	41	20	10								71
bar	BASE	111	9	21	24	12	4	5	3	1	17	207
	REENT	556	9	21	35	12	4	6	3	1	136	783
	TEST	58	6	14	17	8	3	2			9	117
channel	BASE	22	9	5	31	17	3	44				131
	REENT	23	9	5	407	19	3	70				536
	TEST	10	7	1	22	6	2	19				67
child	BASE	77	38	1	1							117
	REENT	1191	66	1	1							1259
	TEST	35	27									62
church	BASE	61	57	2								120
	REENT	206	1068	4								1278
	TEST	35	25	2								62
circuit	BASE	46	6	36	8	7						103
	REENT	115	6	92	8	7						228
	TEST	23	2	15	4	8						52
facility	BASE	20	2	61	28							111
	REENT	35	2	1237	65							1339
	TEST	14		28	14							56
feeling	BASE	65	25	7	3							100
	REENT	1150	52	7	3							1212
	TEST	27	18	2	1							48
grip	BASE	22	6	14	55	1						98
	REENT	22	6	14	305	1						348
	TEST	8	4	9	28							49
material	BASE	52	44	15	7	3						121
	REENT	86	1169	41	7	9						1312
	TEST	27	13	11	8							59
mouth	BASE	54	49	4	4	2	2					115
	REENT	69	499	4	4	2	2					580
	TEST	25	23	3	5							56
nature	BASE	42	9	14	9	15						89
	REENT	1328	66	69	9	22						1494
	TEST	18	3	10	7	6						44
restraint	BASE	15	25	4	31	1	9					85
	REENT	15	333	4	62	1	89					504
	TEST	8	15	3	12	2	4					44
sense	BASE	31	38	7	20	4						100
	REENT	984	76	7	357	4						1428
	TEST	19	13	5	2							39
stress	BASE	3	35	6	7	22						73
	REENT	3	355	127	65	36						586
	TEST	1	18	4	2	13						38

Cuadro 6.5. Distribución de ejemplos de las palabras escogidas de la muestra léxica en inglés del SENSEVAL-2 en los conjuntos de entrenamiento antes y después del *reentrenamiento*, y del test.

No siempre los sentidos más frecuentes en el CA obtienen muchos más ejemplos nuevos. Varios nombres ven modificada su distribución de sentidos. Son notorios los casos de channel y restraint para las que el cambio del sentido más frecuente coincide con el del test[6].

6 Los datos del SENSEVAL-2 fueron procesados con *Minipar* y de ellos se eliminaron las instancias compuestas (art_gallery, por ejemplo) y las no anotadas, por lo que no necesariamente coinciden estas distribuciones de sentidos con las originales

palabra	corpus	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	SMF
art	BASE	49,6	28,6	21,8								s1
	REENT	8,2	87,3	4,4								s2
	TEST	57,7	28,2	14,1								s1
bar	BASE	53,6	4,3	10,1	11,6	5,8	1,9	2,4	1,4	0,5	8,2	s1
	REENT	71,0	1,1	2,7	4,5	1,5	0,5	0,8	0,4	0,1	17,4	s1
	TEST	49,6	5,1	12,0	14,5	6,8	2,6	1,7			7,7	s1
channel	BASE	16,8	6,9	3,8	23,7	13,0	2,3	33,6				s7
	REENT	4,3	1,7	0,9	75,9	3,5	0,6	13,1				s4
	TEST	14,9	10,4	1,5	32,8	9,0	3,0	28,4				s4
child	BASE	65,8	32,5	0,9	0,9							s1
	REENT	94,6	5,2	0,1	0,1							s1
	TEST	56,5	43,5									s1
church	BASE	50,8	47,5	1,7								s1
	REENT	16,1	83,6	0,3								s2
	TEST	56,5	40,3	3,2								s1
circuit	BASE	44,7	5,8	35,0	7,8	6,8						s1
	REENT	50,4	2,6	40,4	3,5	3,1						s1
	TEST	44,2	3,8	28,8	7,7	15,4						s1
facility	BASE	18,0	1,8	55,0	25,2							s3
	REENT	2,6	0,1	92,4	4,9							s3
	TEST	25,0		50,0	25,0							s3
feeling	BASE	65,0	25,0	7,0	3,0							s1
	REENT	94,9	4,3	0,6	0,2							s1
	TEST	56,3	37,5	4,2	2,1							s1
grip	BASE	22,4	6,1	14,3	56,1	1,0						s4
	REENT	6,3	1,7	4,0	87,6	0,3						s4
	TEST	16,3	8,2	18,4	57,1							s4
material	BASE	43,0	36,4	12,4	5,8	2,5						s1
	REENT	6,6	89,1	3,1	0,5	0,7						s2
	TEST	45,8	22,0	18,6	13,6							s1
mouth	BASE	47,0	42,6	3,5	3,5	1,7	1,7					s1
	REENT	11,9	86,0	0,7	0,7	0,3	0,3					s2
	TEST	44,6	41,1	5,4	8,9							s1
nature	BASE	47,2	10,1	15,7	10,1	16,9						s1
	REENT	88,9	4,4	4,6	0,6	1,5						s1
	TEST	40,9	6,8	22,7	15,9	13,6						s1
restraint	BASE	17,6	29,4	4,7	36,5	1,2	10,6					s4
	REENT	3,0	66,1	0,8	12,3	0,2	17,7					s2
	TEST	18,2	34,1	6,8	27,3	4,5	9,1					s2
sense	BASE	31,0	38,0	7,0	20,0	4,0						s2
	REENT	68,9	5,3	0,5	25,0	0,3						s1
	TEST	48,7	33,3	12,8	5,1							s1
stress	BASE	4,1	47,9	8,2	9,6	30,1						s2
	REENT	0,5	60,6	21,7	11,1	6,1						s2
	TEST	2,6	47,4	10,5	5,3	34,2						s2

Cuadro 6.6. Distribución (en porcentaje) de ejemplos de las palabras escogidas de la muestra léxica en inglés del SENSEVAL-2

Se dan también casos de un sentido con un "éxito desmesurado", como pasa con el segundo sentido de art.

Esto puede ser debido a diversas causas, la primera de ellas el propio CNA, cuya distribución de sentidos no tiene por qué coincidir con la de SENSEVAL. Además, WSD supervisado depende mucho del origen de los conjuntos de entrenamiento y test (Escudero et al., 2000c; Agirre y Martínez, 2001). También el método de umbrales tiende a favorecer en intervalos de iteraciones a unos sentidos en detrimento de otros.

Por otro lado, el acierto no está asegurado y la única forma que tenemos de, indirectamente, comprobar lo bueno que ha sido el reentrenamiento es si observamos alguna mejora en la evaluación del corpus de test. También influye el tamaño del CA, escaso en la mayor parte de los casos.

El cuadro 6.7 muestra el promedio de F1 con cada nuevo conjunto de entrenamiento. Se tiene en cuenta que aquellas palabras que no llegan a las 1000 iteraciones, tienen el mismo corpus de entrenamiento para el resto de valores. El único conjunto de atributos que ve incrementado su acierto a medida que se agrandan los corpus es $LBWCDM$. Sin embargo, $sk10$ obtiene la mejor F1 con los conjuntos originales y ningún reentrenamiento consigue mejorarlo.

	BASE	100	200	300	400	500	600	700	800	900	1000
BCDr	48,0	47,3	47,2	47,6	47,0	47,6	47,9	47,6	46,3	45,8	46,4
LBDMr	53,1	51,0	50,5	49,7	49,2	48,8	49,4	49,1	47,8	47,4	47,6
LBWCDM	45,6	45,4	45,6	**47,0**	**47,2**	**47,4**	**47,6**	**47,2**	**47,6**	**47,1**	**46,8**
LWS	53,3	52,2	49,9	50,0	50,3	50,2	49,5	49,4	49,0	49,1	48,7
sk10	61,7	56,8	54,0	54,6	53,8	53,5	52,9	52,9	52,5	52,2	52,1

Cuadro 6.7. Promedios de F1 de evaluación del reentrenamiento por selección de atributos

Discusión

Los resultados no parecen nada buenos pero hay que tener en cuenta que algunos nombres, 'art' entre ellos, han perjudicado en gran medida la precisión final.

- No sabemos hasta qué punto la elección del conjunto no anotado ha influido en el resultado. Tal vez otra fuente de ejemplos dé mejores resultados. [7]
- Se ha realizado el reentrenamiento con un mismo par de selecciones de atributos para todas las palabras. Por un lado, pueden no ser adecuados para algunas de ellas y, por otro, los filtros pueden

7 Haría falta una prueba similar pero haciendo del BNC (*British National Corpus*) la fuente de ejemplos no anotados, puesto que de ahí se extrajeron los conjuntos de entrenamiento y test de SENSEVAL-2; en el momento de la redacción de esta Tesis Doctoral aún no disponíamos de este recurso.

llegar a ser tan restrictivos que no se puedan agotar todas las iteraciones. Esta misma selección puede condicionar la evaluación si se eligen posteriormente atributos distintos.

- Las 5 selecciones de atributos utilizadas para evaluar cada nuevo corpus pueden no ser las más precisas. En las próximas secciones se intentará determinar si es posible determinar ajustes de reentrenamiento a partir del CA inicial.
- La variación de umbral (el reentrenamiento puede llegar a asignar 0,0) ha provocado una descompensación grande en las cantidades de ejemplos de unos sentidos frente a otros.

Por todo ello:

- La consecución de corpus mucho más grandes no garantiza una mayor precisión sobre el test del SENSEVAL dado el escaso número de instancias a resolver; las diferencias de F1 pueden deberse a la clasificación correcta o incorrecta de un único contexto. Tampoco estamos seguros de si el reentrenamiento ha clasificado bien o mal.
- No está clara la influencia de la alteración, en ciertos casos, de la distribución de sentidos.

  Podría limitarse la incorporación de más ejemplos a un sentido concreto, pero si el objetivo es exclusivamente el SENSEVAL. Sería necesario estudiar cuántos de esos nuevos ejemplos han sido mal anotados y si los sentidos más frecuentes en la realidad son los mismos que los de la competición.
- No existe una relación clara entre cantidad de iteraciones (y por ende, de nuevos ejemplos) y precisión. Parece depender más bien de cada palabra, lo que obligaría a individualizar el proceso, como ya se defendió en secciones anteriores.
- Tampoco hay relación entre cantidad de iteraciones y alteración de la distribución de sentidos, ni entre precisiones y tamaños de los conjuntos de entrenamiento y evaluación.
- La elección de la pareja de combinaciones de atributos puede no ser compatible con la mejor selección de atributos para el conjunto de evaluación. En el cuadro 6.7 se intuye que este asunto puede ser más importante de lo que parecía a priori.

- ¿Porqué la selección $sk10$ obtiene resultados tan malos? Creemos que se debe a que el cálculo se ha hecho sobre los corpus de la última iteración, cuando se ve que hay máximos intermedios para ciertos nombres. También está claro que no todos los nombres "reaccionan" igual a los atributos definidos para el reentrenamiento.

Todo parece llevarnos hacia un ajuste de parámetros dependiendo de cuál sea la palabra que queramos enriquecer, y de cuál sea el objetivo de la desambiguación, en qué entorno se va a llevar a cabo. Se comprueba una vez más, por otra parte, la dependencia de los métodos supervisados del dominio de los conjuntos de entrenamiento y evaluación (Escudero et al., 2000c; Martínez y Agirre, 2000).

Las siguientes secciones tratan de arrojar luz sobre estos asuntos.

6.8 Evaluación de la anotación por reentrenamiento

Este experimento tiene por objeto comprobar sobre un corpus anotado si el reentrenamiento acierta al clasificar. Se han utilizado dos corpus distintos, el *Interest Corpus* en primer lugar, por su extensión, y el DSO en segundo, por cubrir más palabras y tener una extensión media. En todo caso, las palabras escogidas son nombres[8].

Tras la evaluación con el corpus Interest, se demostrará empíricamente, la validez del método, concretamente la ganancia frente a estrategias más cercanas al coentrenamiento, y la comparación con los resultados que se obtendrían con esquemas de desambiguación más simples, todo ello, ahora, con el corpus DSO.

6.8.1 Pruebas con el *interest corpus*

En esta primera prueba, se ha dividido el corpus Interest en 5 partes, lo que permite definir 5 parejas de CA y CNA, correspondientes cada una a un reentrenamiento que comienza con una semilla diferente. En todos los casos los atributos han sido ($0WSCMv$: $LSBPDr$).

8 La descripción de estos corpus se encuentra en la sección 2.3.2 y anejo B

Los cuadros 6.8, 6.9 y 6.10 muestran los resultados de estas pruebas en precisión, cobertura de sentidos y cobertura absoluta, respectivamente. Cada 25 iteraciones se ha comprobado cuántos contextos han sido correctamente clasificados y cuántos sentidos han conseguido ejemplos. Los datos que se manejan a partir de ahora son precisión, cobertura y cobertura de sentidos (aunque, al ser una única palabra, se muestra exactamente la cantidad de sentidos reconocidos, no la *ratio* descrita en la sección 6.5.4). Cuando en alguna columna no hay valor es porque el reentrenamiento se ha detenido en una iteración anterior.

			atributos: 0WSCMv - LSBPDr					
palabra	umbral	semilla	25	50	75	100	125	150
interest	0.8-0.0	1	100	100	100			
		2	100	100	100	100	100	100
		3	100					
		4	100					
		5	100					
	0.8-0.4	1	100	100	99,2	98,8	99,0	98,6
		2	100	100	100	100	100	100
		3	98,6	98,4	98,2	98,5	98,6	98,8
		4	97,8	98,7	98,3	97,8	98,1	98,3
		5	100	100	99,4	99,5	99,1	99,2
	0.6-0.4	1	98,9	99,2	99,3	99,4	99,5	99,6
		2	100	100	100	100	100	100
		3	99,0	99,3	99,0	98,7	98,5	98,3
		4	98,5	99,0	99,4	99,4	99,5	99,6
		5	100	100	100	100	100	99,6

Cuadro 6.8. Corpus Interest: precisiones

Como primer hecho relevante, el reentrenamiento tiene un buen comportamiento tanto en precisión como en cobertura de ejemplos y de sentidos. También es cierto que el aprendizaje con cada una de las semillas obtiene buenos resultados sin reentrenamiento, como se puede ver en el cuadro 6.12 que se muestra como referencia. En este cuadro puede verse los datos de una clasificación "normal": se aprende con la semilla y se clasifica el CNA. La precisión es equivalente a la cobertura y F1 puesto que se han clasificados todos los contextos de los correspondientes CNA. Comparadas con las del reentrenamiento, aunque sin clasificar la totalidad de los CNA, las precisiones conseguidas por la clasificación estándar son mucho más bajas.

			atributos: 0WSCMv - LSBPDr					
palabra	umbral	semilla	25	50	75	100	125	150
interest	0.8-0.0	1	3	3	3			
		2	2	2	2	2	2	2
		3	3					
		4	4					
		5	4					
	0.8-0.4	1	3	3	4	4	4	4
		2	2	2	2	2	2	2
		3	4	5	5	5	5	5
		4	4	4	4	5	5	5
		5	4	5	5	5	5	5
	0.6-0.4	1	4	4	4	4	4	4
		2	3	3	3	3	3	3
		3	4	4	5	5	5	5
		4	4	4	4	4	4	4
		5	4	4	4	5	5	5

* sentidos = cantidad absoluta de sentidos

Cuadro 6.9. Corpus Interest: cobertura de sentidos

			atributos: 0WSCMv - LSBPDr					
palabra	umbral	semilla	25	50	75	100	125	150
interest	0.8-0.0	1	3,7	5,2	5,6			
		2	3,4	4,9	6,5	7,9	9,3	10,6
		3	0,9					
		4	1,1					
		5	1,0					
	0.8-0.4	1	3,7	5,2	6,8	8,8	10,1	11,4
		2	3,4	4,9	6,5	7,9	9,3	10,6
		3	3,8	6,6	9,0	10,3	11,7	13,1
		4	2,4	4,1	6,3	9,5	11,2	12,5
		5	2,1	6,1	8,7	10,1	11,6	12,9
	0.6-0.4	1	4,8	6,4	7,8	9,4	10,8	12,1
		2	4,4	7,5	10,2	11,6	13,0	14,3
		3	5,2	7,9	10,4	12,4	14,1	15,6
		4	3,6	5,5	8,2	9,5	10,9	12,9
		5	5,3	7,3	8,7	11,1	12,4	14,7

Cuadro 6.10. Corpus Interest: coberturas

No hay grandes diferencias entre unas semillas y otras, salvo en la segunda que cubre tan sólo dos sentidos de los seis anotados en el corpus, al menos hasta la iteración 150. Este hecho puede constatarse en el cuadro 6.11, que resume las ejecuciones expuestas, promediando los resultados entre las cinco semillas.

Todas las pruebas han conseguido una precisión por encima del 98 %. El umbral 0.8-0.0 obtiene las mejores precisiones pero el pro-

	Atributos: 0WSCMv - LSBPDr					
	Iteraciones					
Umbral	25	50	75	100	125	150
	Precisión (%)					
0.8–0.0	100	100	100	100	100	100
0.8–0.4	99,3	99,4	99,0	98,9	99,0	99,0
0.6–0.4	99,3	99,5	99,5	99,5	99,5	99,4
	Cobertura de sentidos (cantidad)					
0.8–0.0	3,2	2,5	2,5	2	2	2
0.8–0.4	3,4	3,8	4	4,2	4,2	4,2
0.6–0.4	3,8	3,8	4	4,2	4,2	4,2
	Cobertura absoluta(%)					
0.8–0.0	2,0	5,0	6,0	7,9	9,3	10,6
0.8–0.4	3,1	5,4	7,5	9,3	10,8	12,1
0.6–0.4	4,7	6,9	9,1	10,8	12,2	13,9

Cuadro 6.11. Corpus Interest: resultados promedio para las cinco semillas

Atributos	Semilla	CNA	Precisión
LSBPDr	1	1897	79,1
	2	1897	79,4
	3	1897	76,5
	4	1897	82,8
	5	1880	81,0
	promedio		79,8
0WSCMv	1	1897	80,2
	2	1897	80,3
	3	1897	77,1
	4	1897	82,6
	5	1880	81,8
	promedio		80,4

Cuadro 6.12. Corpus Interest: valores de referencia con nuestro sistema WSD–ME

ceso se detiene demasiado pronto (lo que también significa que se detectan menos sentidos y se clasifican menos contextos). De hecho, este umbral es demasiado restrictivo, y sólo con la segunda semilla consigue agotar todas las iteraciones. Es por esto que, aunque las precisiones son del 100 %, no consideramos que sea la mejor ejecución posible.

La relajación de este parámetro (con umbrales 0.8-0.4 y 0.6-0.4) permite alcanzar la última iteración, lo que se traduce en más sentidos y contextos. Concretamente, cuatro de los seis sentidos posibles han sido clasificados al menos en un contexto, y se ha cubierto alrededor del 13 % del CNA.

Aunque el número de iteraciones puede considerarse bajo, los valores de precisión y cobertura de sentidos son tan altos que no se prevé una degradación excesiva si el proceso continuara. El siguiente cuadro 6.13 muestra el comportamiento del reentrenamiento sobre el mismo corpus hasta 500 iteraciones para el umbral 0.8-0.4, promediando entre las 5 semillas.

	100	200	300	400	500
Precisión	98,9	98,9	98,7	97,8	96,2
Cobertura absoluta	9,3	18,4	26,8	35,9	46,9
Sentidos detectados	4,2	5,0	5,0	5,0	5,0

Cuadro 6.13. Corpus Interest: resultados promedio de iterar 500 veces con umbral 0.8-0.4

6.8.2 Pruebas con el corpus DSO: validez del método

Se van a mostrar varias pruebas que demostrarán que el método planteado es ventajoso frente a otras estrategias de clasificación.

A partir de este momento, los datos que se muestren se refieren a palabras escogidas del corpus DSO analizado con *Minipar*.

Para cada palabra se define un CA y un CNA (dividido cada conjunto en 10 partes, con una distribución uniforme de ejemplos y eliminados aquellos sentidos con menos de 10 ejemplos, las proporciones son 2/8, para el CA y el CNA respectivamente). La cantidad total de ejemplos de cada sentido en el CNA es la que se muestra en el cuadro 6.14.

¿Sirve de algo el reentrenamiento o basta con la clasificación estándar?

Se ha planteado un esquema iterativo relativamente complicado con la esperanza de obtener mayores precisiones aunque sea a costa de la cobertura absoluta y la cobertura de sentidos; ¿no se podría

nombres	S1	S2	S3	S4	S5	S6	S7	S8
age	276	82	85					
air	40	148	180	17	17	37		
art	171	136	39	10				
body	187	71	48	10				
century	73	290						
church	184	96	18					
class	205	52	82					
course	122	126	9	23	26			
death	203	198	12	26				
difference	225	129	90					
door	58	351						
example	107	51	75	20	10			
experience	271	144	32					
girl	246	82	32					
ground	20	123	127	35	56	13		
land	31	122	49	54	38	30	18	36
light	97	15	77	71	25	20	13	73
mind	111	82	104	28	65			
moment	235	135						
need	181	200						
picture	54	20	14	86	98	73	12	
process	288	11	17	10				
public	248	34						
purpose	87	182	77					
section	139	32	38	128				
sense	171	9	24	10	106	118		
society	84	241	20					
stage	146	72	36	42				
surface	180	20	22					
table	135	132	45					
town	117	67	204					
voice	36	173	57	42	35			

Cuadro 6.14. Distribución de sentidos en el CNA de nombres seleccionados del DSO

haber conseguido simplemente con definir ese umbral de probabilidad para una clasificación estándar? ¿Es necesario definir dos clasificadores para cada sentido, no basta con uno? Responder a estas preguntas supone demostrar la efectividad de los filtros de precisión: propuestas parciales y propuesta común, y umbrales.

De las propuestas parciales. Se efectúan varias pruebas con umbrales 0.8-0.0 y 0.8-0.4. Para verificar el filtro de las propuestas parciales definimos dos conjuntos de tipos de atributos, A_1 y A_2, y ejecutamos tres veces variando la configuración de los reentrenamientos: $(A_1 : A_2)$, (A_1), y (A_2).

La primera ejecución es la que corresponde al método propuesto, el reentrenamiento basado en dos clasificadores parciales para cada sentido. Las otras dos ejecuciones consisten en utilizar un único clasificador por sentido. De esta forma podemos comparar la precisión que obtiene cada selección por separado con la obtenida al utilizar los dos a la vez.

En realidad, cuando sólo utilizamos un clasificador estamos retirando el filtro de consenso entre clasificadores binarios parciales, pero se mantiene el de la construcción de la propuesta combinada y el umbral de confianza: un ejemplo es clasificado como positivo y perteneciente a un determinado sentido si la probabilidad obtenida en el clasificador parcial supera el umbral y si ningún otro sentido lo reclama como suyo.

En el cuadro 6.15 se muestran los valores de precisión, cobertura de sentidos, cobertura absoluta y F1, respectivamente, obtenidos con varias configuraciones de atributos y cada 25 iteraciones hasta 150. Éste es un resumen de los cuadros D.1, D.2, D.3 y D.4 del anejo D.

		0.8-0.0						0.8–0.4					
		25	50	75	100	125	150	25	50	75	100	125	150
2cc	prec	95,5	95,0	93,4	92,6	92,8	92,6	81,3	79,9	78,2	76,5	74,9	73,9
	sents	13,7	14,1	14,1	14,1	14,1	14,1	39,3	39,8	39,8	40,0	40,0	40,0
	cabs	1,1	1,8	2,2	2,6	2,9	3,2	7,1	12,9	17,7	22,8	26,9	30,3
	F1	2,0	3,4	4,0	4,8	5,3	5,8	10,7	18,1	23,3	28,1	31,2	33,8
1c	prec	92,9	92,9	92,0	91,5	91,4	91,8	73,7	71,4	69,4	67,7	66,9	66,1
	sents	19,7	19,8	19,8	19,8	19,8	18,1	46,8	48,3	48,6	48,7	49,0	49,1
	cabs	2,8	4,4	5,3	6,0	6,8	6,2	11,4	21,2	29,7	36,4	41,9	46,6
	F1	4,8	7,6	9,0	10,1	11,4	10,7	14,8	24,5	31,3	35,6	39,0	41,5

Selecciones:

0mCDdv	McBbrW
0mMCBDW	0mMCcBDbdrW
LSBPDr	0WSCMv
mcBSp	0MCbls

Cuadro 6.15. Promedios entre las distintas pruebas de la validez del filtro de propuestas parciales

La parte superior del cuadro, etiquetada como *2cc*, muestra precisión (*prec*), cobertura de sentidos (*sents*), cobertura absoluta (*cabs*) y F1 de los reentrenamientos que utilizan dos clasificadores para confeccionar las propuestas parciales de cada iteración. La parte inferior,

es el promedio de las mismas medidas de las ejecuciones de las mismas selecciones de atributos pero con un único clasificador por sentido.

Por ejemplo, se realizó un reentrenamiento con la configuración $0mCDdv - McBbrW$, otro con únicamente $0mCDdv$, y un tercero con $McBbrW$. De la misma forma se procedió con las otras tres selecciones.

Se observa que, efectivamente, los reentrenamientos obtienen un incremento de precisión respecto de los entrenamientos con un único clasificador por sentido, aunque las diferencias son mínimas cuando nos fijamos en los datos referentes al umbral 0.8-0.0. No obstante, hay que decir que puesto que hemos restringido tanto la respuesta de los sistemas, en la mayoría de las palabras estamos respondiendo a un único sentido y con muy pocos ejemplos.

Es en el umbral 0.8-0.4 donde se ve claramente la bondad de nuestra aproximación. En este caso la cobertura absoluta y de sentidos sube sensiblemente y se observa una mejora considerable utilizando el sistema binario.

Parece que el umbral 0.8 sea más beneficioso para nuestro objetivo final, pero su cobertura de sentidos está claramente por debajo de umbrales más permisivos. No olvidemos que hay palabras que empiezan a clasificar a umbrales más bajos: podríamos combinar los umbrales, no aplicar el mismo a todas.

Refiriéndonos a la selección de atributos propuesta, todas las pruebas son similares, incluso muy posiblemente redundantes, ya que se han elegido los atributos buscando, principalmente, la complementariedad entre atributos relajados y no relajados y cubriendo, más o menos, todo el conjunto de posibles propiedades a observar. Esto es así salvo para la segunda prueba donde uno de los clasificadores es un subconjunto del otro (en cuanto a los atributos definidos). También cumple que recoge la casi totalidad de los atributos definidos para nuestro sistema de WSD. No obtiene los mejores resultados, lo que viene a confirmar, nuevamente, que no por acumular atributos en el aprendizaje vamos a obtener un mejor resultado.

De la propuesta común. En este caso se retira, también, el filtro de la propuesta combinada y tan sólo se mantiene el umbral de probabilidad.

Así, de un aprendizaje-clasificación por cada sentido, pasamos al proceso normal de un aprendizaje-clasificación por palabra, esto es, el clasificador ya no devuelve *positivo* o *negativo* sino el sentido directamente.

Puesto que ahora la clasificación de un ejemplo incluye a todas las posibles clases, el concepto de umbral es ligeramente diferente: un contexto es clasificado como de una determinada clase si la diferencia entre la probabilidad más alta y la segunda más alta supera ese umbral.

El cuadro 6.16 muestra los resultados de esta forma de clasificar y se ve, claramente, que las precisiones están lejos de las obtenidas mediante reentrenamiento. Si los comparamos con el reentrenamiento en el cuadro 6.15 (donde la precisión más baja es de un 73,9 %), la definición de un umbral de probabilidad no basta por si solo para conseguir altas precisiones.

	precisión				cobertura abs.				F1			
atributos	0.8	0.7	0.6	0.0	0.8	0.7	0.6	0.0	0.8	0.7	0.6	0.0
0mCDdv	63,4	64,1	64,6	55,6	23,4	28,3	34,0	99,3	24,0	28,3	32,8	55,4
0mMCBDW	64,2	65,7	66,8	57,7	17,5	18,6	20,2	99,2	19,1	20,6	22,4	57,5
0mMCcBDbdrW	59,7	61,1	63,7	55,0	18,9	20,1	22,0	99,8	19,0	20,5	22,9	55,0
0WSCMv	60,7	62,9	64,7	57,0	19,7	21,3	23,6	99,7	19,9	22,1	24,7	56,9
LSBPDr	61,6	63,7	66,1	56,0	19,1	21,2	25,2	100,0	19,8	22,3	26,6	56,0
McBbrW	59,6	60,3	60,8	53,3	17,0	17,6	18,3	92,3	17,3	18,1	18,8	51,2
mcBSp	62,1	64,8	66,8	55,3	18,1	20,7	25,4	99,9	19,0	22,2	27,1	55,2
Promedios	61,6	63,2	64,8	55,7	19,1	21,1	24,1	98,6	19,7	22,0	25,1	55,3

Cuadro 6.16. Validez del método: sólo umbral

Coentrenamiento y reentrenamiento

En secciones anteriores mencionamos el coentrenamiento como germen del reentrenamiento. Como ya se dijo, el primero varía con respecto del segundo en que sólo se activa un clasificador por sentido, y aunque se sigue manteniendo la configuración de dos conjuntos

de atributos para el aprendizaje, estos se van alternando al pasar de una iteración a otra (el primer conjunto se utiliza en las iteraciones impares y el segundo en las pares).

El cuadro 6.17 es una comparación de las ejecuciones con coentrenamiento y los obtenidos con reentrenamiento (resumidos en el cuadro 6.15). Se muestran los resultados de aplicar el coentrenamiento a varias configuraciones de atributos, con umbrales 0.8-0.0 y 0.8-0.4, para las iteraciones 25 y 150. En los cuadros D.5 (precisión), D.6 (cobertura de sentidos), D.7 (cobertura absoluta), y D.8 (resumen comparativo), en el anejoanexotablas se pueden consultar los datos de estas ejecuciones con mayor detalle.

	Precisión				F1			
	0.8-0.0		0.8-0.4		0.8-0.0		0.8-0.4	
	25	150	25	150	25	150	25	150
COentr	91,3	88,6	72,2	63,0	6,1	14,1	13,7	42,7
REentr	95,5	92,1	81,3	73,9	2,0	4,5	10,7	33,8
	Cobertura sentidos				Cobertura absoluta			
COentr	22,9	22,9	48,0	52,7	3,6	9,0	10,5	52,5
REentr	13,7	13,7	39,3	40,0	1,1	2,5	7,1	30,3

Cuadro 6.17. Coentrenamiento y reentrenamiento: resumen

El coentrenamiento obtiene, en general, una F1 y coberturas (de sentidos y absoluta) mayores que el reentrenamiento, pero las precisiones son claramente menores. Uno de los problemas del coentrenamiento es la degradación de la precisión a medida que se va procesando una mayor cantidad de CNA, y saber en qué momento detener el proceso.

El reentrenamiento mantiene mejor el nivel de precisión, es más restrictivo a la hora de proponer una clasificación, al tiempo que sus filtros permiten la detención de la ejecución cuando prevé que los niveles de precisión van a deteriorarse.

Si nos fijamos en el experimento con umbral 0.8-0.4, el menos restrictivo de los dos, en la iteración 150 la cobertura absoluta del coentrenamiento llega al 52,5 % de cobertura promedio mientras el reentrenamiento tan sólo consigue el 30 %.

Sin embargo, la diferencia de precisión entre uno y otro es de casi el 11 % (63 % y 73,9 %) a favor del reentrenamiento. Si el sistema sigue iterando, probablemente reentrenamiento aumentaría la cobertura al tiempo que la precisión podría decrecer un poco, pero siempre por encima del coentrenamiento.

El coentrenamiento tiene la desventaja de tener que afinar mucho más los conjuntos de atributos. Recordemos que este método elimina el primer filtro, lo que significa un mayor número de ejemplos positivos en esta primera fase a un menor grado de certeza o seguridad.

En general, hemos observado que las palabras que funcionan mejor con coentrenamiento, no obtienen una precisión mucho mayor que con reentrenamiento. Por contra, las palabras que obtienen mejor resultado con reentrenamiento, lo hacen con una diferencia clara, lo que indica que el filtrado adicional respecto al coentrenamiento es crucial para nuestro objetivo.

6.8.3 Discusión

La aportación que buscamos con nuestra propuesta, el reentrenamiento, es aumentar la confianza en la corrección de las asignaciones de sentidos a los contextos. Para ello se han establecido varias configuraciones generales de atributos de aprendizaje en dos umbrales, uno muy restrictivo y otro más relajado.

Podemos concluir que el reentrenamiento consigue altas precisiones con la definición apropiada de parámetros. A estas prestaciones contribuyen los filtros definidos en el método, tanto de consenso entre clasificadores parciales como de umbral.

Es claramente ventajoso, en cuanto a precisiones obtenidas, si lo comparamos con una adaptación del algoritmo de coentrenamiento de Blum y Mitchell (1998). También se ha comprobado que un aprendizaje-clasificación estándar, pero con condiciones de clasificación más exigentes, no consigue los niveles de precisión que sí alcanza el reentrenamiento.

A continuación se van a explorar diversas estrategias para establecer la configuración apropiada para el reentrenamiento, centrándonos en la selección de las parejas de conjuntos de atributos principalmente.

6.9 Evaluación sobre el corpus del SENSEVAL-2 en español

Vamos a intentar comprender cómo funciona el reentrenamiento desde lo más general a lo más particular, desde conjuntos de palabras hasta sentidos. Para ello se estudia la relevancia de los tipos de atributos en el *proceso* de reentrenamiento.

A partir de un análisis empírico de los grupos de atributos definidos para el sistema general de desambiguación, y aplicados al reentrenamiento, se muestran las características que mejor se adecúan al corpus utilizado.

Evidentemente, el éxito del método propuesto y denominado como reentrenamiento se basa en el éxito del propio sistema de WSD descrito en el capítulo 4. Se puede obtener una mejora en precisión pero el éxito de la clasificación sigue basándose en los valores de probabilidad obtenidos por el modelo de máxima entropía. Si para una palabra concreta el método general funciona mal, es muy posible que el reentrenamiento no alcance los niveles de precisión esperados (por encima de un 80 o 90 %). Interesante es, pues, estudiar los sentidos particularmente además de las palabras globalmente. La evaluación indica que el éxito de ciertos atributos está fuertemente relacionado con sentidos más que con palabras.

Del proceso en si, se probará que la aproximación de selección de clases limitada por umbrales es efectiva. Nuevamente, la diferencia entre unas palabras y otras y entre sentidos es notoria.

A continuación, partiendo de la evaluación individual de cada grupo de atributos definido en la figura 6.3 (es una extensión de las previamente mostradas en 4.1), se van a establecer varias estrategias de selección de atributos y comprobar si los datos del corpus están correlacionados con ellas.

Las estrategias de selección de atributos, que al mismo tiempo utilizaremos para comprobar el comportamiento del *reentrenamiento* en el corpus para español del SENSEVAL-2, son:

Selección global de atributos: ordenando de forma descendente los atributos por precisión promediada sobre todas las palabras, y eligiendo los mejores, se establecen

No relajados
- 0: la palabra ambigua
- l: lemas (de palabras llenas) en ±1, ±2, ±3
- s: palabras en posiciones ±1, ±2, ±3
- b: lemas de pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- c: pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- p: categorías gramaticales de palabras en ±1, ±2, ±3
- k_m: lemas de nombres que aparecen en al menos el $m\,\%$ de contextos de un sentido
- r: rol gramatical de la palabra ambigua
- d: la palabra de la que depende la ambigua
- m: palabra compuesta a la que pertenece la ambigua
- o: palabras en posiciones ±1, ±2
- l: lemas (de palabras llenas) en ±1, ±2, ±3
- v: el verbo del que depende la palabra ambigua

Relajados
- L: lemas (de palabras llenas) en ±1, ±2, ±3
- W: palabras llenas en ±1, ±2, ±3
- S: palabras en ±1, ±2, ±3
- B: lemas de pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- C: pares de palabras en $(-2,-1)$, $(-1,+1)$, $(+1,+2)$
- P: categorías gramaticales en ±1, ±2, ±3
- D: la palabra de la que depende la ambigua
- M: palabra compuesta a la que pertenece la ambigua

Figura 6.3. Lista completa de grupos de atributos

conjuntos de atributos individualizados para el reentrenamiento.

Selección basada en sentidos: añadimos un grado más de complejidad y hacemos un estudio no por palabra exclusivamente sino que se tienen en cuenta los resultados de cada sentido. Ahora la ordenación de grupos de atributos se hace por medidas basadas en los datos de todos los sentidos de una misma palabra.

Selección global secuencial: de nuevo el orden descendente de precisión promediada de los grupos de atributos, calculada sobre el total de palabras, se utiliza para una nueva estrategia de reentrenamiento. Este proceso consiste en varios reentrenamientos sucesivos, cada uno con atributos diferentes.

6.9.1 Estudio previo del corpus de entrenamiento

El estudio de los ejemplos de entrenamiento consiste en la prueba 3FCV de todos y cada uno de los grupos de atributos definidos en la figura 6.3[9] y de forma individual en varias ejecuciones de reentrenamiento.

Esta evaluación individual por grupos de atributos se muestra en el cuadro 6.18, que expone los resultados obtenidos sobre los nombres y verbos de los conjuntos de ejemplos y test del SENSEVAL-2 en español.

	TOTAL				NOMBRES				VERBOS			
	P	C	F1	Cabs	P	C	F1	Cabs	P	C	F1	Cabs
C	93,4	9,7	17,5	10,4	97,4	9,7	17,6	10,0	89,5	9,7	17,4	10,8
c	93,1	9,7	17,5	10,4	97,4	9,7	17,6	10,0	88,9	9,7	17,4	10,9
b	85,2	11,7	20,6	13,8	89,5	11,1	19,8	12,4	81,4	12,4	21,6	15,3
B	84,7	11,7	20,5	13,8	89,1	11,2	19,9	12,6	80,8	12,2	21,2	15,1
W	81,2	16,7	27,8	20,6	87,5	14,6	25,1	16,7	76,6	19,0	30,5	24,8
L	67,6	21,1	32,1	31,2	73,7	19,5	30,9	26,5	62,8	22,8	33,4	36,3
I	66,7	21,3	32,3	32,0	73,3	19,3	30,6	26,4	61,6	23,5	34,0	38,1
D	66,2	7,1	12,8	10,7	69,8	8,6	15,3	12,3	60,8	5,4	9,9	8,9
d	66,1	6,8	12,3	10,2	69,4	8,1	14,6	11,7	61,2	5,3	9,7	8,6
0	59,1	30,3	40,1	51,3	62,7	34,3	44,3	54,7	54,7	26,0	35,2	47,5
v	58,9	6,8	12,2	11,6	62,6	8,3	14,6	13,2	53,6	5,3	9,6	9,8
s	57,2	46,2	51,1	80,8	58,3	49,8	53,7	85,4	55,8	42,4	48,2	75,9
r	56,8	19,1	28,6	33,6	57,9	24,4	34,3	42,2	54,8	13,3	21,5	24,3
S	56,7	43,1	49,0	76,0	56,7	46,8	51,3	82,6	56,7	39,0	46,2	68,8
o	56,1	45,5	50,2	81,0	58,8	50,2	54,2	85,4	52,8	40,2	45,7	76,2
P	54,7	31,0	39,6	56,7	54,7	37,2	44,3	68,0	54,6	24,3	33,6	44,4
p	49,3	34,0	40,3	69,0	48,7	39,0	43,3	80,0	50,2	28,7	36,5	57,1

Cuadro 6.18. 3fcv corpus entrenamiento SENSEVAL-2 español: datos globales por grupos de atributos

Los datos consignados en el cuadro 6.18, y el desglose por categoría, muestran la precisión (columna *P*), la cobertura (columna *C*), la medida F1 y la cobertura absoluta (columna *Cabs*). El cuadro 6.19, a partir de esos mismos datos, muestra las diferentes ordenaciones de atributos por precisión descendente tanto para todas las palabras como para nombres y verbos únicamente.

9 En realidad, no se han estudiado todos: los grupos m y M no son aplicables porque Conexor (para español) no detecta colocaciones, y el grupo km tampoco se ha utilizado por simplificar el proceso.

Los resultados así mostrados se han conseguido desactivando uno de los clasificadores binarios por sentido, lo que genera un único entrenamiento parcial por cada sentido y el paso directo a la fase de confección de la propuesta común. En otras palabras, el primer filtro no tiene efecto. El umbral es 0.9-0.9 (el umbral puede bajar durante la ejecución hasta 0.0) y las iteraciones 100, suficientes dado el tamaño de los conjuntos de ejemplos para cada palabra.

Los datos reflejados en el cuadro 6.18 indican que las colocaciones (grupos c, C, b y B) son los más precisos. La cobertura absoluta ronda el 10 %[10].

El siguiente grupo con una precisión por encima del 80 % es W, que observa las formas de las palabras en una ventana (-3, +3) alrededor de la palabra objetivo subiendo la cantidad de contextos cubiertos al 20 %.

El resto de grupos de atributos ya obtiene precisiones mucho más bajas, al tiempo que aumentan las coberturas absolutas. Los grupos D (palabra de la que depende la palabra objetivo en el árbol sintáctico de la oración) y v (primer verbo del que depende la palabra objetivo) obtienen una precisión discreta al tiempo que cubren pocos contextos (sus F1 no llegan al 13 %) mientras que otros atributos clasifican la mayor parte del CNA pero con precisiones entre el 50 % y el 60 %. Los atributos menos precisos son los asociados a las etiquetas gramaticales (grupos P y p).

En general, tienen mayor precisión las formas de las palabras frente a los lemas (precisiones de C y B, respectivamente, o de W y L).

Las pequeñas diferencias entre grupos relajados y no relajados, confirman que, a nivel de grupos de atributos, y por cuestiones de eficiencia, el uso de los primeros es suficiente y recomendable.

Por último, si observamos el cuadro 6.19, no hay diferencias remarcables entre nombres y verbos, siendo las ordenaciones casi idénticas.

Los valores concretos por palabra se pueden consultar en los cuadros D.9, D.10, D.11 y D.12 (precisión, cobertura, F1 y cobertura absoluta), en el anejo D.

10 Entiéndase que estamos haciendo referencia a valores promediados entre las 3 pruebas, es decir, el 10 % como promedio de contextos cubiertos de las partes (*folds*) usados como "no anotados" en cada prueba.

TOTAL	NOMBRES	VERBOS
C	c	C
c	C	c
b	b	b
B	B	B
W	W	W
L	L	L
I	I	I
D	D	d
d	d	D
0	0	S
v	v	s
s	o	r
r	s	0
S	r	P
o	S	v
P	P	o
p	p	p

Cuadro 6.19. 3fcv corpus entrenamiento SENSEVAL-2 español: grupos de atributos ordenados por precisión

¿Para qué sirven estos datos? Es información acerca del corpus, indicios de cómo abordar el reentrenamiento, y la base de un par de propuestas que pretenden aprovechar la idea de que cada palabra es diferente porque lo que estamos clasificando son, finalmente, sentidos. El enfoque de este análisis es que en condiciones reales tendremos el suficiente corpus como para extraer los datos que se van a mostrar en las siguientes secciones.

6.9.2 Evaluación sobre el corpus de test

Como se mencionaba anteriormente, se van a evaluar tres propuestas de selección de atributos y de ejecución del reentrenamiento. Estas propuestas se basan en los datos calculados en la sección anterior, el estudio por 3FCV del corpus de entrenamiento.

Selección global de atributos

La selección global de atributos consiste en definir los clasificadores parciales a partir de la ordenación por precisión descendente de los grupos tal y como muestra el cuadro 6.19 en su columna 'TOTAL',

la que hace referencia a las precisiones calculadas sobre todos los nombres y verbos.

La configuración se hace alternando los 6 primeros grupos de la lista ($CcbBWL$) con lo que se obtiene una configuración (CbW - cBL). Los datos de la ejecución se pueden consultar en el cuadro 6.20

atributos1	CbW
atributos2	cBL
umbral	0.8-0.2
iteraciones	175

Cuadro 6.20. Datos de la evaluación del reentrenamiento sobre el corpus de test de SENSEVAL-2 español

Salvo la definición de atributos, que cambiará a lo largo de los siguientes experimentos, el umbral y el número de iteraciones se van a mantener.

Se ha hecho una pequeña modificación en el programa para que las palabras que no consigan clasificar nada en las primeras iteración a tan alto umbral rebajen automáticamente este valor. En definitiva, y dado que el umbral no es el filtro de más peso dentro del reentrenamiento, se ha preferido relajarlo en aras de alcanzar una cobertura mayor.

Así mismo, el tamaño del corpus de test permite establecer tan sólo 175 iteraciones, excesivas para la mayor parte de las palabras y suficiente para todas.

Los resultados de la ejecución se muestran en el cuadro 6.21, donde se exponen precisión (columna *pre*), cobertura (*cob*), F1 (*F1*), cobertura absoluta (*cabs*), cantidad de sentidos distintos en el corpus (*sents*) y sentidos detectados en la clasificación (*cubiertos*).

El sistema obtiene una precisión del 82 % cubriendo algo menos de un tercio del total de contextos a clasificar. Así también, la cobertura de sentidos es buena, obteniendo clasificaciones en algo más de dos tercios de los presentes en el corpus.

Por categorías, tanto nombres como verbos obtienen precisiones altas. Llaman la atención los nombres masa y operación, y el verbo

pos	palabra	prec	cob	f1	cabs	sents	cubiertos
N	autoridad	75,0	8,8	15,8	11,8	5	2
	bomba	93,8	40,5	56,6	43,2	2	2
	canal	88,9	39,0	54,2	43,9	5	3
	circuito	100,0	20,4	33,9	20,4	4	3
	corazón	80,0	25,5	38,7	31,9	4	3
	corona	94,4	42,5	58,6	45,0	3	3
	gracia	100,0	41,0	58,1	41,0	4	2
	grano	60,0	13,6	22,2	22,7	3	3
	hermano	92,9	22,8	36,6	24,6	3	3
	masa	57,1	9,8	16,7	17,1	4	3
	naturaleza	80,0	14,3	24,2	17,9	6	5
	operación	38,5	10,6	16,7	27,7	4	4
	órgano	91,4	39,5	55,2	43,2	3	3
	partido	88,9	42,1	57,1	47,4	2	2
	pasaje	62,5	12,2	20,4	19,5	4	3
	programa	87,5	14,9	25,5	17,0	4	3
	tabla	75,0	22,0	34,0	29,3	3	2
V	actuar	36,4	7,3	12,1	20,0	6	4
	apoyar	84,0	28,8	42,9	34,2	4	4
	apuntar	100,0	14,3	25,0	14,3	6	1
	clavar	76,5	29,5	42,6	38,6	7	5
	conducir	78,6	20,4	32,4	25,9	7	5
	copiar	81,8	17,0	28,1	20,8	5	3
	coronar	76,9	27,0	40,0	35,1	4	3
	explotar	72,7	19,5	30,8	26,8	5	3
	saltar	90,0	24,3	38,3	27,0	11	3
	tocar	80,0	27,0	40,4	33,8	10	5
	usar	86,7	23,2	36,6	26,8	3	2
	vencer	80,0	30,8	44,4	38,5	6	4
	todos	82,1	24,6	37,9	30,0	137	91
	nombres	84,9	26,0	39,8	30,7	63	49
	verbos	78,7	23,0	35,6	29,2	74	42

Cuadro 6.21. Evaluación configuraciones reentrenamiento por selección de los grupos de mayor precisión global sobre *corpus* test SENSEVAL-2 español

actuar por los pobres resultados, pero viene a confirmar el distinto comportamiento del método cuando procesa diferentes palabras.

Dentro de ese intentar afinar el método para cada palabra en particular, las siguientes propuestas de configuración de los reentrenamientos se calculan palabra a palabra, como se verá en la siguiente sección. Se espera, además, mantener la precisión pero aumentando la cobertura, ya que cada palabra se procesará con los supuestos mejores atributos para ella.

Finalmente, vamos a intentar plasmar todas estas intuiciones mediante una serie de medidas cuantitativas que, se espera, recojan todo o parte de la discusión anterior en valores que se puedan orde-

nar. Estas medidas son el promedio de aciertos menos el promedio de errores, la precisión promediada, la medida F1, y el promedio de errores, y todas se refieren a un grupo de atributos. El cálculo de estas medidas se basa en los promedios del 3FCV sobre los conjuntos de entrenamiento mostrado anteriormente, y se describen a continuación.

Sea w una palabra, y ga un grupo de atributos (de los definidos en la figura 6.3), entonces:

precisión (PRE): el dato que se utiliza para ordenar descendentemente los grupos de atributos es la precisión para cada palabra. Se utilizan los datos del cuadro D.9.

precisión promediada entre sentidos (PP): Aquí se utilizan las precisiones por clase, ponderando los valores según la cantidad de ejemplos de cada sentido y hallando el promedio. Sea pre_i la precisión del sentido i, S_w el número de sentidos de la palabra, o_i la cantidad de ejemplos del sentido i, y T_w es el total de ejemplos de la palabra:

$$PP_{ga} = \frac{\sum_{i=1}^{S_w}(pre_i * o_i)}{T_w * S_w} \tag{6.1}$$

promedio de errores (PE): Ahora sólo se tienen en cuenta los errores por clase y se hace un promedio absoluto, sin ponderar según la distribución de sentidos. Sea e_i la cantidad de errores del sentido i de la palabra.

$$PE_{ga} = \frac{\sum_{i=1}^{S_w} e_i}{S_w} \tag{6.2}$$

Con estas medidas se obtienen clasificaciones de atributos que servirán para generar los conjuntos de características que definirán las parejas de clasificadores para cada sentido. Todas las ordenaciones son ascendentes excepto para PE.

El detalle de los grupos de atributos ordenados por palabra (sólo los seis primeros) se puede ver en el cuadro 6.22. Este cuadro se ha calculado a partir del análisis 3FCV de la sección anterior.

criterio	palabra	pos	1	2	3	4	5	6	palabra	pos	1	2	3	4	5	6
PRE	autoridad	N	I	L	W	o	S	L	programa	N	c	C	o	S	b	B
PE			I	L	W	0	v	r			c	C	b	W	B	L
PP			S	s	o	r	P	0			L	c	C	I	W	s
PRE	bomba	N	b	B	b	B	W	I	tabla	N	b	B	o	r	S	P
PE			b	B	c	C	I	L			b	B	c	C	v	W
PP			I	L	0	b	B	c			v	d	D	I	L	0
PRE	canal	N	c	C	c	C	W	r	actuar	V	W	c	C	s	0	S
PE			c	C	b	B	W	d			d	D	v	W	0	L
PP			c	C	W	b	B	L			s	S	o	p	0	W
PRE	circuito	N	c	C	W	0	c	C	apoyar	V	I	L	W	0	v	b
PE			c	C	b	B	W	v			c	C	b	B	d	D
PP			I	L	S	o	W	c			I	L	o	r	s	P
PRE	corazón	N	b	B	v	W	0	L	apuntar	V	c	C	b	B	W	I
PE			b	B	c	C	W	d			c	C	v	W	b	B
PP			W	b	B	L	I	c			W	s	S	I	o	L
PRE	corona	N	0	r	W	0	v	r	clavar	V	c	C	W	b	B	I
PE			W	r	0	o	s	p			c	C	d	D	b	B
PP			0	r	o	s	S	p			d	D	c	C	W	b
PRE	gracia	N	b	B	b	B	v	d	conducir	V	b	B	B	L	I	s
PE			b	B	c	C	d	D			b	B	c	C	d	D
PP			d	D	b	B	o	s			I	b	B	c	C	L
PRE	grano	N	b	B	c	C	W	I	copiar	V	b	B	P	S	p	L
PE			b	B	c	C	W	I			b	B	W	r	0	L
PP			s	0	o	r	S	P			W	0	o	S	b	B
PRE	hermano	N	d	D	W	I	L	P	coronar	V	c	C	b	B	W	v
PE			d	D	m	0	S	p			c	C	b	B	v	d
PP			r	p	S	P	s	d			b	B	W	c	C	S
PRE	masa	N	b	B	W	b	B	v	explotar	V	b	B	W	r	0	L
PE			b	B	c	C	W	d			b	B	c	C	d	D
PP			d	D	I	L	o	W			b	B	W	S	o	L
PRE	naturaleza	N	b	B	W	P	L	p	saltar	V	W	b	W	c	C	S
PE			b	B	W	I	L	P			b	B	W	P	L	p
PP			o	s	S	r	p	P			W	L	I	s	S	o
PRE	operación	N	c	C	b	B	L	I	tocar	V	b	C	S	r	p	P
PE			c	C	b	B	W	I			B	c	m	D	d	W
PP			I	L	W	d	D	b			W	C	c	I	s	o
PRE	órgano	N	c	C	o	p	0	W	usar	V	W	0	v	b	B	s
PE			c	C	b	B	W	D			v	W	b	B	L	I
PP			W	c	C	I	L	b			r	s	I	c	C	o
PRE	partido	N	c	C	o	r	P	0	vencer	V	c	C	I	s	S	o
PE			c	C	W	b	B	v			B	W	I	L	r	0
PP			I	L	v	b	B	s			B	I	L	W	S	s
PRE	pasaje	N	v	d	S	o	W	c								
PE			d	D	v	0	S	r								
PP			S	s	o	P	p	r								

Cuadro 6.22. 3fcv corpus entrenamiento SENSEVAL-2 español: ordenación de los 6 primeros grupos de atributos por palabra según los criterios PRE, PE y PP

Selección basada en sentidos

Tomando como ejemplo el nombre autoridad, vamos a ver las distintas ordenaciones de grupos de atributos y las configuraciones del reentrenamiento.

Posición	atrib	PRE	atrib	PE	atrib	PP
1	l	100	l	0,0	S	12,1
2	L	100	L	0,0	s	11,6
3	W	100	W	0,0	o	11,3
4	0	90,1	0	0,2	r	10,4
5	v	75,0	v	0,2	P	10,1
6	b	66,7	r	1,0	0	10,1
7	B	66,7	P	1,8	l	9,6
8	c	66,7	S	1,9	L	9,6
9	C	66,7	s	2,1	W	9,6
10	r	64,8	p	2,2	p	9,3
11	S	61,5	o	2,3	v	8,6
12	o	58,6	b	33,3	b	3,0
13	s	58,3	B	33,3	B	3,0
14	P	56,7	c	33,3	c	3,0
15	d	50,0	C	33,3	C	3,0
16	D	50,0	d	33,4	d	2,0
17	p	48,3	D	33,4	D	2,0

Cuadro 6.23. Ordenación de grupos de atributos para el nombre autoridad según los criterios PRE, PE y PP

Se ha configurado cada reentrenamiento a partir de los 6 grupos que encabezan la lista de cada criterio, según los datos del cuadro 6.23. De forma alternada se van definiendo los conjuntos de atributos con los de las posiciones impares para el primer clasificador (1, 3, y 5) y los pares en el segundo clasificador (2, 4, y 6). De esta forma se obtienen las siguientes selecciones de atributos para los pares de clasificadores binarios:

PRE: lWv - $L0b$

PE: lWv - $L0r$

PP: SoP - $sr0$

El sistema de configuración es el mismo para todas las palabras, y de la ejecución del reentrenamiento obtenemos los datos que se muestran en el cuadro 6.24

			precisión			cobertura			f1			cob. absoluta			cob. sentidos			
pos	palabra	test	PRE	PP	PE	PRE	PP	PE	PRE	PP	PE	PRE	PP	PE	sents	PRE	PP	PE
N	autoridad	34	71,4	73,7	100,0	14,7	41,2	14,7	24,4	52,8	25,6	20,6	55,9	14,7	5	2	4	1
	bomba	37	94,1	94,4	94,1	43,2	45,9	43,2	59,3	61,8	59,3	45,9	48,6	45,9	2	2	2	2
	canal	41	89,5	89,5	88,2	41,5	41,5	36,6	56,7	56,7	51,7	46,3	46,3	41,5	5	4	4	4
	circuito	49	92,3	88,9	100,0	24,5	32,7	16,3	38,7	47,8	28,1	26,5	36,7	16,3	4	4	4	4
	corazón	47	83,3	68,2	83,3	21,3	31,9	21,3	33,9	43,5	33,9	25,5	46,8	25,5	4	3	3	3
	corona	40	77,8	75,0	77,8	52,5	52,5	52,5	62,7	61,8	62,7	67,5	70,0	67,5	3	3	3	3
	gracia	61	95,2	92,0	95,2	32,8	75,4	32,8	48,8	82,9	48,8	34,4	82,0	34,4	4	3	3	3
	grano	22	100,0	71,4	100,0	13,6	45,5	18,2	24,0	55,6	30,8	13,6	63,6	18,2	3	2	3	3
	hermano	57	85,2	75,0	74,3	40,4	57,9	45,6	54,8	65,3	56,5	47,4	77,2	61,4	3	2	3	3
	masa	41	80,0	53,8	80,0	9,8	17,1	9,8	17,4	25,9	17,4	12,2	31,7	12,2	4	2	3	2
	naturaleza	56	100,0	70,0	75,0	12,5	50,0	10,7	22,2	58,3	18,8	12,5	71,4	14,3	6	2	4	4
	operación	47	40,0	40,0	41,2	17,0	17,0	14,9	23,9	23,9	21,9	42,6	42,6	36,2	4	4	4	4
	órgano	81	94,4	90,2	92,3	42,0	45,7	29,6	58,1	60,7	44,9	44,4	50,6	32,1	3	3	3	3
	partido	57	90,0	95,7	90,9	31,6	38,6	17,5	46,8	55,0	29,4	35,1	40,4	19,3	2	2	2	2
	pasaje	41	63,6	52,4	54,5	17,1	26,8	14,6	26,9	35,5	23,1	26,8	51,2	26,8	4	4	4	3
	programa	47	88,9	72,7	100,0	17,0	17,0	10,6	28,6	27,6	19,2	19,1	23,4	10,6	4	3	3	2
	tabla	41	100,0	70,6	100,0	17,1	29,3	17,1	29,2	41,4	29,2	17,1	41,5	17,1	3	2	2	2
V	actuar	55	33,3	48,3	33,3	5,5	25,5	5,5	9,4	33,3	9,4	16,4	52,7	16,4	6	4	6	4
	apoyar	73	75,8	73,8	91,7	34,2	65,8	15,1	47,2	69,6	25,9	45,2	89,0	16,4	4	4	4	2
	apuntar	49	85,7	55,6	100,0	12,2	30,6	12,2	21,4	39,5	21,8	14,3	55,1	12,2	6	2	6	1
	clavar	44	78,6	78,6	78,6	25,0	25,0	25,0	37,9	37,9	37,9	31,8	31,8	31,8	7	4	4	4
	conducir	54	84,6	72,2	84,6	20,4	24,1	20,4	32,8	36,1	32,8	24,1	33,3	24,1	7	4	6	4
	copiar	53	50,0	66,7	66,7	18,9	22,6	15,1	27,4	33,8	24,6	37,7	34,0	22,6	5	3	3	3
	coronar	74	75,9	84,2	76,2	29,7	21,6	21,6	42,7	34,4	33,7	39,2	25,7	28,4	4	3	3	3
	explotar	41	63,6	61,1	63,6	17,1	26,8	17,1	26,9	37,3	26,9	26,8	43,9	26,8	5	5	5	5
	saltar	37	84,6	76,5	50,0	29,7	35,1	2,7	44,0	48,1	5,1	35,1	45,9	5,4	11	5	5	1
	tocar	74	93,8	84,6	88,2	20,3	29,7	20,3	33,3	44,0	33,0	21,6	35,1	23,0	10	4	5	4
	usar	56	75,0	81,3	81,3	69,6	69,6	23,2	72,2	75,0	36,1	92,9	85,7	28,6	3	3	3	2
	vencer	65	77,8	77,8	77,8	21,5	21,5	21,5	33,7	33,7	33,7	27,7	27,7	27,7	6	3	3	3
	todos	1474	79,4	74,8	**80,1**	26,7	**37,3**	21,0	40,0	**49,8**	33,3	33,6	**49,9**	26,3	137	91	**107**	84
	nombres	799	**84,3**	77,0	82,2	27,5	**40,3**	24,3	41,5	**52,9**	37,5	32,7	**52,3**	29,5	63	47	**54**	48
	verbos	675	74,0	71,9	**76,8**	25,8	**33,8**	17,2	38,2	**46,0**	28,1	34,8	**47,0**	22,4	74	44	**53**	36

Cuadro 6.24. Evaluación configuraciones reentrenamiento por palabra sobre corpus test SENSEVAL-2 español: precisión, cobertura y F1

El promedio de errores, PE, aventaja ligeramente al criterio de precisión, PRE, (80,1 % frente a 79,4 % de precisión)[11], por delante los dos del criterio precisión ponderada, PP (74,8 %). Sin embargo, la mayor precisión para los nombres la obtiene PRE (84,3 %) mientras que para los verbos vuelve a ser PE el mejor (76,8 %).

Sin embargo, cuando se tienen en cuenta los valores de cobertura (estándar y absoluta) y de cobertura de sentidos, es el criterio PP el que aventaja a los otros dos. Así, el valor de F1 es claramente favorable a PP. Dicho en otras palabras, el criterio PP es menos restrictivo a la hora de clasificar pero también menos fiable.

No olvidemos que nuestro objetivo es clasificar con la mayor fiabilidad por lo que, aún siendo la F1 del criterio PP claramente mejor que la de los otros dos, no creemos que sea la mejor opción aun cuando su cobertura sea mayor.

11 En realidad, las selecciones de atributos resultan muy parecidas.

En todos los casos, la cobertura de sentidos es buena, lo que indica que no sólo las clases más frecuentes son las detectadas, sino que también son capaces de aportar clasificaciones de los otros sentidos.

Podemos concluir que la diferencia de precisión entre PRE y PE puede no ser significativa y que, siendo tan pequeña, por el valor de F1 de ambas es preferible el criterio PRE que, además, es más sencillo de calcular.

Llaman la atención palabras concretas, como el nombre operación o el verbo actuar, que obtienen precisiones muy bajas con cualquier criterio. Interpretamos que estas palabras forman parte del grupo de palabras cuyo análisis del corpus de entrenamiento es erróneo tal y como se ha planteado, y necesitan otro tipo de información que, ahora mismo, no somos capaces de determinar cuál es.

El nombre circuito, por contra, tiene precisiones muy altas con cualquier criterio y la cobertura de sentidos es total (se detectan ejemplos de los cuatro posibles sentidos de la palabra), y el CNA del verbo usar se clasifica casi por entero con precisiones también altas, otra vez con cualquiera de los tres criterios propuestos.

Esto viene a corroborar, una vez más, la diferencia que existe en el aprendizaje de unas y otras palabras, y que estos criterios de selección de atributos, aún siendo eficaces en cuanto a la precisión global obtenida, posiblemente necesitan más información.

Finalmente, la diferencia en los resultados entre nombres y verbos ya no es tan grande como en otras evaluaciones (expuestas en el capítulo 5), lo que entendemos es un hecho más a favor del método de reentrenamiento propuesto por nosotros.

Selección global secuencial

El siguiente experimento está más enfocado a la aplicación en competiciones tipo SENSEVAL-2, es decir, queremos comprobar si un esquema de reentrenamiento aportaría ventajas frente a un proceso normal de aprendizaje-clasificación.

La hipótesis es que cada grupo de atributos es capaz de detectar la clase de un pequeño conjunto de contextos y, además, con una alta

precisión, si nos guiamos por los datos obtenidos del 3FCV sobre el corpus de entrenamiento (véase el cuadro 6.18).

Siendo el corpus de entrenamiento el CA y el de test el CNA, se plantean cuatro reentrenamientos sucesivos de tal forma que la salida de uno es la entrada del otro; el corpus de aprendizaje del siguiente reentrenamiento está engrosado con los contextos clasificados por el anterior, al mismo tiempo que el CNA va disminuyendo de tamaño. Finalmente, se entrena un clasificador ME con el corpus de aprendizaje resultado de los anteriores y se clasifican el resto de instancias no cubiertas.

PARCIAL				ACUMULADO								
a	e	t	pre	a	t	**pre**	cob	rec	f1		at1	at2
311	26	337	92,3	311	337	**92,3**	15,1	14,0	24,3	R1:	C	
302	87	389	77,6	613	726	**84,4**	32,6	27,6	41,5	R2:	W	
172	92	264	65,2	785	990	**79,3**	44,5	35,3	48,8	R3:	Lbc	lBC
16	5	21	76,2	801	1011	**79,2**	45,4	36,0	49,5	R4:	SWr	LCd
691	523	1214	56,9	1492	2225	**67,2**	100,0	67,1	67,1	F:	sk5(*)	

*clasificación estándar, sin reentrenamiento

Cuadro 6.25. Evaluación reentrenamiento secuencial sobre corpus test SENSEVAL-2 español: precisión, cobertura y F1

En el cuadro 6.25 se muestran los resultados del experimento. Cada reentrenamiento viene identificado por *Rx* y la clasificación final por *F*, y se acompaña de los conjuntos de atributos utilizados en cada ejecución. Este cuadro detalla los valores parciales obtenidos en cada fase de reentrenamiento (*PARCIAL*), presentando los aciertos (*a*), errores (*e*), la suma de ambos (*t*) y, finalmente, la precisión alcanzada (*pre*); los valores acumulados (*ACUMULADO*) muestran, además, la cobertura absoluta (*cob*) y la cobertura (*rec*, de *recall*).

Los dos primeros reentrenamientos, $R1$ y $R2$, se configuran con un único clasificador parcial, aprovechando la alta precisión del grupo de atributos, en nuestro experimento C y W, respectivamente. No se ha utilizado el B por ser muy similar al C. Los otros dos reentrenamientos, que se supone deben clasificar contextos de una mayor dificultad, se configuran con dos selecciones de grupos de atributos atendiendo a su precisión y al tipo de información que procesa. Concretamente, el tercer reentrenamiento se basa en las mínimas diferencias entre atributos relajados y no relajados (Lbc - lBC), bus-

cando nuevamente los atributos con una buena precisión al tiempo que asegurar una cierta cobertura absoluta. El cuarto ya presenta diferencias notables en el tipo de información tratada (SWr - LCd).

La clasificación final pretende cubrir ya el total de contextos todavía en el CNA, aprendiendo con el corpus generado por el cuarto reentrenamiento.

Efectivamente, el primer reentrenamiento (sólo el grupo C) consigue un 92 % de precisión, aunque con una cobertura muy baja del 15 %. Sin embargo, el segundo reentrenamiento (W) consigue clasificar casi la misma cantidad de contextos, siendo en este caso la precisión del 77 %. El tercer reentrenamiento ya utiliza clasificadores parciales diferenciados (Lbc - lBC) pero obtiene tan sólo un 65 % de precisión y clasifica menos contextos. El último reentrenamiento se configura de manera aún más restrictiva (SWr - LCd) alcanzando el 76 % de precisión pero clasificando sólo 21 instancias nuevas.

En todos los experimentos realizados hasta ahora sobre reentrenamiento se ha visto un decremento de la precisión al aumentar la cobertura, hecho éste que se repite aquí. El objetivo era clasificar tantos contextos como pudiéramos pero asegurando una mínima precisión.

Observemos, ahora, los valores acumulados en el mismo cuadro 6.25. Al llegar al cuarto reentrenamiento, la menor precisión de las últimas ejecuciones se compensa precisamente con esa menor cobertura, y la precisión y cobertura acumuladas, es decir, teniendo en cuenta las clasificaciones efectuadas hasta entonces, son del 79 % (precisión) y 45 % (cobertura).

El CNA restante es considerado como difícil de clasificar y, efectivamente, aplicado un entrenamiento normal la precisión acumulada decae hasta el 67 %.

Podemos comparar estos valores con los que se obtienen de la evaluación de un clasificador entrenado con $sk5$ a partir de únicamente el corpus de aprendizaje, como se puede consultar en el cuadro 6.26

El reentrenamiento secuencial supone una mejora sobre el clasificador $sk5$ de casi un 5 %. Todas las categorías ven incrementada su tasa de acierto, siendo los adjetivos en los que se observa una diferencia menor. Es obvio, pues, que el efectuar varios reentrenamien-

	sk5	reent	dif
Todos	62,5	67,1	+4,58
Nombres	61,2	67,0	+5,76
Verbos	52,1	57,2	+5,10
Adjetivos	75,3	78,2	+2,64

Cuadro 6.26. Comparación de un clasificador entrenado con $sk5$ y los reentrenamientos secuenciales: tasas de acierto

tos previos a la clasificación normal aumenta la cantidad de aciertos finales.

Sin embargo, el hecho remarcable es que somos capaces de asegurar un precisión cercana al 80 % con reentrenamientos sucesivos, a falta de concretar qué hacer con estas clasificaciones. Puede que determinadas tareas del PLN precisen de una determinada precisión aún cuando la cobertura absoluta no sea total. También es plausible la posibilidad de que la última fase de clasificación se haga con otros métodos (si es que se pretende clasificar el 100 % de los contextos de test, que con el reentrenamiento es difícil de conseguir).

En el cuadro 6.27 se puede ver el detalle del reentrenamiento secuencial por categorías.

TODOS			NOMBRES			VERBOS			ADJETIVOS				
pre	rec	F1	**pre**	rec	F1	**pre**	rec	F1	**pre**	rec	F1	at1	at2
92,3	14,0	24,3	**90,2**	13,8	23,9	**88,2**	11,0	19,6	**97,5**	17,5	29,6	R1: C	
84,4	27,6	41,5	**85,0**	24,2	37,6	**74,8**	21,1	32,9	**91,0**	38,6	54,2	R2: W	
79,3	35,3	48,8	**79,6**	31,2	44,8	**68,3**	27,8	39,5	**88,0**	48,3	62,4	R3: Lbc	IBC
79,2	36,0	49,5	**79,3**	32,0	45,6	**68,0**	27,9	39,6	**88,2**	49,5	63,4	R4: SWr	LCd
67,1	67,1	67,1	**67,0**	67,0	67,0	**57,2**	57,2	57,2	**78,0**	78,0	78,0	F: sk5(*)	

*clasificación estándar, sin reentrenamiento

Cuadro 6.27. Evaluación reentrenamiento secuencial sobre corpus test SENSEVAL-2 español: detalle por categorías

Los adjetivos y los nombres son los que mantienen unos valores de precisión altos durante los reentrenamientos, pero también se destacan los verbos por conseguir precisiones por encima del 68 %. En todas las categorías, la última clasificación, la que debe procesar los contextos que aún no han podido ser clasificados, es la que hace caer la precisión. Cómo se ha hecho anteriormente, podríamos di-

ferenciar los reentrenamientos por categorías, buscando esa mejora final en los verbos.

El porqué realizar esa clasificación final se encuentra en nuestro deseo de llegar a cubrir el 100 % de los contextos de test. No obstante, los reentrenamientos anteriores han dificultado la tarea de este último clasificador puesto que ya han detectado y procesado los contextos *más fáciles de clasificar*.

Como información adicional, el cuadro D.13 (anejo D) muestra los valores acumulados de precisión, cobertura y F1 para cada palabra en los sucesivos reentrenamientos y clasificación final.

6.9.3 Discusión

Hemos reconfirmado la teoría que venimos sosteniendo desde el *capítulo anterior*, que cada palabra tiene unas características propias que hace que el aprendizaje deba particularizarse.

La explicación a este comportamiento diferenciado de las palabras ante el aprendizaje está en la propia selección de atributos. Mientras que un tipo de atributos es muy preciso para algunas palabras, en otras no consigue etiquetar ningún ejemplo (lo que no sería malo en sí ya que, simplemente, no contribuiría a la clasificación) o, lo que es peor, genera errores de anotación.

Es sabido que ciertas palabras tienen sentidos fuertemente relacionados con la forma en que se escribe la palabra, o por pertenecer a expresiones comunes (*"tasa de interés"*). La esperanza de los métodos de aprendizaje automático es que sea el propio proceso de entrenamiento el que detecte las características relevantes y diferenciadoras y descarte las que deriven en confusión. Hay demasiada información, por una parte, y escasea por otro. Un corpus contendrá cientos, miles, o incluso decenas de miles de ejemplos, pero habrá sido confeccionado por "alguien" y puede estar descompensado, con un excesivo número de ocurrencias de ciertos atributos que, en la realidad, no son tan relevantes. Por otro lado, hace falta más información que la puramente estadística para discernir el sentido apropiado, conocimiento que no le estamos suministrando a nuestro método.

Por todo ello, dado el corpus en el que estemos trabajando, cada palabra tiene su propia clasificación de atributos, del mejor al peor,

y la inclusión de uno "malo" puede ser perniciosa. Pero también habrá que estudiar no ya sólo la palabra en su conjunto sino sus sentidos. Si un sentido se relaciona habitualmente con una composición de palabras, las funciones de atributo relacionadas serán muy precisas, mientras que para otros sentidos no tendrá relevancia.

Se ha visto que, globalmente, los atributos con mayor especialización sintáctica (palabras compuestas, dependencias, roles, ...) alcanzan una mayor precisión que los tradicionales de ocurrencias de palabras en una determinada posición. Esta clasificación, en el marco que estamos defendiendo, el reentrenamiento, nos puede dar una pista de como combinar las definiciones de los clasificadores.

Si hablamos de cada palabra en particular, bien sea por los resultados calculados globalmente para ellas, bien por un estudio más complejo del comportamiento de sus sentidos, hay algunas que son "notorias" por el éxito o el fracaso obtenido. No hay una correlación clara entre, por ejemplo, número de sentidos y precisiones altas, ya que hay palabras con pocos sentidos que se clasifican mal en general y, al contrario, hay palabras con muchos sentidos que obtienen esas altas precisiones.

6.10 Conclusiones

Los resultados del método propuesto de reentrenamiento son modestos pero no desechables. El método proporciona un límite dentro del cual podemos estar seguros del porcentaje de acierto en la incorporación de ejemplos nuevos a partir de un corpus no anotado.

Otra cosa es su aplicación a ejercicios comparativos como SENSEVAL. ¿Un corpus más grande ayudará a aumentar el acierto frente a otros sistemas? ¿Cómo de grande ha de ser ese corpus?

Cada palabra presenta un comportamiento distinto tanto en el proceso de clasificación estándar como en el reentrenamiento. Hasta la fecha, es un problema abordado por muy pocos investigadores y con soluciones discutibles dado que siempre nos tenemos que basar en un corpus concreto.

A continuación se resumen los detalles sobre el método propuesto.

Cómo mejorar el reentrenamiento

Del estudio del corpus de entrenamiento de la muestra léxica en español del SENSEVAL-2, se ha visto que un proceso más pormenorizado de palabras y sentidos puede mejorar la precisión, nuestro objetivo básico. Como principal desventaja tenemos que su aplicación es más compleja, puesto que primero hay que obtener las ordenaciones de grupos de atributo por cada palabra.

No obstante, quedan cuestiones pendientes como es su aplicación a entornos concretos tipo SENSEVAL, donde la consecución de un corpus de aprendizaje mayor no asegura un mayor acierto en la evaluación. Todos los datos obtenidos del estudio de atributos deberían poder aplicarse a este fin, junto con las modificaciones necesarias del método para adecuarse a ejercicios de este tipo.

Cómo aumentar la precisión

El elegir los atributos adecuados y diferenciados por palabra nos asegura una alta precisión. El problema, insistimos, consiste en determinar una forma eficiente de realizar esa elección. Hemos demostrado que el estudio previo de los atributos permite obtener precisiones muy altas para ciertas palabras.

Otro parámetro que influye es lo que hemos denominado umbral, un valor de diferencia mínimo de probabilidad para decidir clasificar (en la primera fase, la de propuestas parciales) un ejemplo como positivo o negativo. Un umbral lo suficientemente alto contribuye a mejorar la precisión. Nuevamente nos encontramos con un obstáculo que son las propias palabras, puesto que no todas tienen el mismo umbral de inicio, pero esta deficiencia es subsanable.

Otra forma de asegurar la precisión es limitar el número de iteraciones, que combinado con el umbral de inicio puede asegurar la precisión más alta.

Cómo aumentar la cobertura

Todo lo dicho en el punto anterior es cierto pero también lo es que, en muchos casos, la cobertura es ridícula, lo que deriva en un aumento insignificante del CA.

En primer lugar hay que anotar que ciertas palabras tienden a resistirse al proceso de reentrenamiento ya que ejecutan muy pocas iteraciones. Quiere esto decir que los conjuntos de atributos de los clasificadores parciales generan clasificaciones tan dispares que se ponen de acuerdo en muy pocos casos.

Seleccionar conjuntos de atributos solapados genera gran cantidad de clasificaciones positivas que, por un lado, incrementan la posibilidad de anotar y, por otro, puede llevar a más competencia puesto que pueden ser varios los sentidos que se disputen un mismo ejemplos. Sin embargo, la elección de atributos ha de ser cuidadosa para no generar demasiados errores que "hundan" la precisión ya que, en realidad, estamos relajando uno de los filtros del reentrenamiento. Es mejor un reparto de atributos que estén diferenciados, aunque sea mínimamente. Incluso la limitación a dos clasificadores binarios por sentido no es tal, sino que se podría plantear un número mayor que diera respuestas por consenso, votación o cualquier otro método, de tal forma que se suavizara la influencia de un conjunto de atributos concreto.

El umbral de confianza dinámico ayuda a no detener el proceso en una iteración demasiado temprana. El método ideal, a priori, es iniciar el proceso con el umbral más alto y permitir que sean las iteraciones las que decidan si necesitan bajarlo o no en busca de una mayor facilidad de clasificación. Se espera que, al incorporar primero los ejemplos que cumplen este mayor umbral, el reentrenamiento irá siendo dirigido por estos. Lo cierto es que permitir una variación muy grande del umbral no siempre es bueno, permite una mayor cobertura pero la precisión decae por debajo de niveles aceptables.

Depende de cada palabra, pero parece adecuado limitar la variación del umbral a 0.2 o a 0.4 todo lo más, perjudicando una cobertura mayor pero asegurando la precisión necesaria. El método propuesto en este trabajo tiene en cuenta la cantidad de ejemplos que se han clasificado para cada sentido, de forma que las variaciones de umbrales se hacen en los sentidos que han obtenido menos ejemplos

nuevos. Se pueden producir, entonces, diferencias muy grandes entre lo que se permite a unos sentidos y a otros. Si partimos, por ejemplo, de una configuración 0.9-0.6, puede darse el caso de un sentido a 0.9 y todos los demás a 0.3, con lo que el primero no compite con los demás, y serle los filtros del reentrenamiento exageradamente adversos.

Visto que bajar por igual los umbrales tampoco genera resultados espectaculares, se podría hacer una modificación tal que las diferencias de umbrales estuvieran limitadas también, algo así como 0.9-0.6(0.2), es decir, que el proceso parta de un umbral de inicio 0.9, que puede bajar hasta 0.3, pero que en ningún caso permita diferencias entre sentidos de más de 0.2.

Otra forma de aumentar la cobertura es incrementar el número de iteraciones. Otra vez chocamos con palabras concretas, aquellas ya mencionadas que no agotan todas las iteraciones, pero peor aún las que a partir de cierto número comienzan a perder precisión de forma exagerada. Depende del tamaño del corpus y del propio proceso de reentrenamiento ya que la tendencia es a clasificar durante un buen número de iteraciones sólo para unos pocos sentidos.

El método está diseñado para evitar las clasificaciones erróneas por el simple establecimiento de un límite a partir del cuál ya no estamos tan seguros del acierto. No olvidemos que por debajo está el propio método de ME así que los mismos errores que éste comete en el proceso normal de WSD los comete en el reentrenamiento. La esperanza es que los sucesivos filtros y la propia estructuración del flujo de ejecución detecten esos errores, cosa que, por otro lado, se ha demostrado factible.

No se ha encontrado una relación clara entre todos los parámetros y la precisión final obtenida. Son las palabras y los sentidos (o puede que fuera mejor decir los corpus de los que disponemos) los que marcan una ejecución u otra.

Uso de otros métodos

Nos referimos a si incorporar a los entrenamientos parciales no ya dos clasificadores basados en ME sino n clasificadores basados en métodos diversos. La respuesta ha de ser forzosamente positiva,

Volver al índice/Tornar a l'índex

sobre todo si atendemos a lo visto en el SENSEVAL-2: muchos de los sistemas presentados combinan varios métodos en un único clasificador final, y que creemos que representa una fuerte tendencia que se verá refrendada en el próximo SENSEVAL. También es posible plantear el uso de varios sistemas de alta precisión en paralelo o en cascada, buscando una mayor cobertura al tiempo que se mantiene la precisión.

Inclusive, en el capítulo 5 hemos experimentado con la cooperación entre ME y un método basado en el conocimiento, marcas de especificidad, con resultados satisfactorios.

7

Conclusiones finales

7.1 Conclusiones sobre el trabajo presentado

Se ha presentado un método de desambiguación semántica automática basado en los modelos de probabilidad condicional de máxima entropía. Se ha evaluado bajo similares condiciones a las definidas para la muestra léxica del español para el SENSEVAL-2.

Una vez diseñado e implementado, ha sido aplicado en un esquema de semilla con reentrenamiento iterativo. El objetivo es alcanzar una alta precisión de desambiguación aún a costa de la cobertura. Puesto que el resultado final ha de ser la clasificación de ejemplos no anotados de fuentes no restringidas para la confección automática de corpus, la cobertura no es tan importante pero la precisión es crucial.

El sistema está a la altura de los mejores, dado el actual estado tecnológico. Su desarrollo actual no incluye ningún sistema de suavizado de atributos (salvo la propia naturaleza del método) y la selección se basa en grupos completos de atributos, y no hace uso de otros recursos o preprocesos como, por ejemplo, reconocimiento de entidades, que pudieran aportar una mayor calidad de la información a tratar. Por todo ello, el sistema está abierto a numerosas mejoras, entre ellas la exploración de nuevas fuentes de información. El reentrenamiento se plantea como un punto de inicio válido y prometedor para la generación automática de corpus anotados semánticamente.

Resumiendo:

Es un método basado en corpus y esto conlleva ciertas ventajas y muchas servidumbres. Se depende de lo que hay, muchas de las conclusiones que se extraen de un determinado conjunto de datos no son trasladables a otros dominios, a otros corpus. El éxito

se mide contra otro corpus y no estamos seguros de que su aplicación a una situación real sea tan satisfactoria como pensamos.

Sin embargo, si de trabajar con conjuntos anotados semánticamente se trata, los modelos de máxima entropía son candidatos perfectos para la tarea. Se ha visto que son capaces de rivalizar con cualquier otro método similar y, por supuesto, con métodos basados en conocimiento.

Pero cada palabra es distinta porque, aún dentro del mismo corpus recopilado por cierta persona o equipo, parece que la forma de aprender sus características más relevantes, las que nos ayuden a identificar sus sentidos, cambian fundamentalmente de una a otra.

Se puede afinar más, son los propios sentidos los que se apoyan más en un conjunto de hechos estadísticamente significativos que en otro.

Se habla de un atributo como la función concreta que nos informa de un hecho de la palabra (o del sentido) muy particular. Nosotros proponemos agrupar esos atributos, hablar más de selección de fuentes de información relevantes que de selección de atributos, hablar más de «*palabras a la izquierda de interés*» que de «*'tasa de' está a la izquierda de interés*».

No hay suficientes corpus, ni cubren todas las palabras, ni tampoco el tamaño parece el adecuado. Siendo nuestra aproximación tan dependiente de este recurso es un handicap especialmente grave, más si pretendemos trabajar en lenguas que no sean el inglés.

Es curioso que WSD (supervisado) necesite grandes cantidades de ejemplos y que precisamente WSD pueda ser la solución a este problema. La realidad es que no se busca el "gran corpus" como ventaja para ganar en SENSEVAL, sino que la asignatura pendiente es como ayudar a las demás tareas.

Así, se propone un método de los denominados de semilla. Partiendo de un conjunto limitado de ejemplos, atacamos un conjunto mucho mayor pero desconocido, buscando las clasificaciones más probablemente acertadas. El método se basa (y se diferencia de) en el *coentrenamiento*, intentando superar sus inconvenientes y forzando su aplicación concreta, la clasificación en sen-

tidos. El coentrenamiento ha sido seguido con interés entre la comunidad científica por lo que de prometedor tiene. Aquí se propone un método basado en sucesivos filtros que intentan dificultar la clasificación errónea y, por ende, asegurar la corrección.

Los resultados, en experimentación sobre varios corpus ha dado resultados muy interesantes que ofrecen múltiples líneas a explorar. Sobre una evaluación de tipo SENSEVAL los resultados han sido discretos y dispares. No obstante, el origen de los conjuntos de ejemplos puede ser determinante a la hora de aprovechar información no anotada para el aprendizaje.

Y al final nadie duda de la ventajosa y teórica aportación de WSD para, por ejemplo, traducción automática, pero aún no podemos contribuir. El gran reto de WSD es demostrar su utilidad. Es más, ya no se trata de ayudar a una tarea concreta, vamos más allá, queremos la comprensión por una máquina en términos aún sin determinar pero parecidos a los de un humano. En eso estamos.

7.2 Trabajos en progreso y líneas futuras

El sistema de WSD basado en máxima entropía se ha diseñado como vehículo para otros fines como pueda ser, precisamente, la confección de corpus anotados semánticamente. Por ello, la mejora del sistema es el beneficio de la aplicación.

En el aspecto técnico, sería interesante comprobar la influencia de un algoritmo de optimización diferente al empleado, el *generalized iterative scaling*. Tanto el propio procedimiento de optimización como el criterio de convergencia se han mostrado suficientes al comparar los resultados obtenidos con otro sistemas. No obstante, haría falta comprobar si una mayor sofisticación matemática generaría un incremento de la tasa de acierto.

De la propia tarea, la definición de más tipos de atributos es una mejora obvia. Se pueden buscar nuevas fuentes de información lingüística que se puedan incorporar al sistema como atributos o como preproceso.

Respecto de este último aspecto, si de un sistema completo y funcional de WSD se trata, lo aquí mostrado ha de convertirse en un

módulo más de una estructura más compleja. Reconocedores de entidades, expresiones comunes, anáfora, patrones, redes semánticas, etc., pueden ser componentes de ese sistema final.

Otro aspecto no tratado aquí sería medir la influencia de las distintas herramientas externas que se han utilizado en la clasificación final. En particular, nos parece interesante comprobar y comparar las salidas de los analizadores sintácticos. Podríamos llevarnos alguna sorpresa.

Otra forma de abordar WSD podría ser una estrategia descendente basada en la generalidad de las clases. Si disponemos de una jerarquía de conceptos en forma de árbol cuya raíz es un concepto que engloba a todos los demás, y se establecen niveles dentro de ese árbol hasta llegar a los sentidos de WN, podemos realizar varias fases de desambiguación.

Supongamos que establecemos tres niveles de detalle de conceptos: categorías lexicográficas de WN, *WN Domains*, y *synsets*. Es claro que la reducción del conjunto de clases mejora la precisión. Así, un determinado contexto sería clasificado según el primer nivel, lo que descartaría algunos de los conceptos del segundo; la segunda desambiguación tendría en cuenta esta "poda" y, a su vez, obligaría a descartar ciertos sentidos, lo que supone, finalmente, clasificar entre unos pocos sentidos de todos los posibles. Obviamente, los niveles no están constreñidos a únicamente los mencionados.

A este respecto, y que nosotros conozcamos, ha habido un intento parecido con resultados parcialmente negativos (Escudero et al., 2001), pero incluso sus autores reconocen su sorpresa ante este comportamiento. Nuestra intención es explorar, igualmente, esta vía.

Nótese que estamos induciendo, aunque de forma un poco relajada, *conocimiento*. Estamos convencidos de que los métodos no supervisados, si no suficientes, son absolutamente complementarios a los supervisados. No estamos descubriendo nada, la actualidad es una mezcla de técnicas cuya frontera está lo bastante borrosa como para no pensar en ella. La plena integración de las dos aproximaciones que estamos desarrollando en nuestro grupo de investigación, la que fundamenta este trabajo y la de Montoyo y Palomar (2000), es una meta no muy distante aunque dificultosa. También pretendemos incorporar métodos ajenos, supervisados y no supervisados, ayuda-

dos por la Universidad Politécnica de Valencia y por la Universidad de Jaen, en sistemas combinados o cooperativos.

En cuanto al reentrenamiento queda mucho por hacer. En primer lugar, ciertos aspectos del coentrenamiento (evitados por nosotros) merecen ser estudiados, como pueda ser la proporción entre positivos y negativos. Aunque no se ha mencionado aquí, sí se ha experimentado con este parámetro, pero la dificultad de combinarlo con los otros nos hizo desistir. Lo cierto es que para encuentros como el SENSEVAL la distribución de sentidos sí es un problema a tener en cuenta.

En la clasificación de ejemplos sin anotación influye la mayor cantidad de ejemplos anotados pero también hay sentidos con pocos ejemplos de partida que consiguen clasificar. El porqué en unos casos se da este hecho y en otros no es otra tarea a abordar.

Para evitar la degradación de la precisión a medida que se superan iteraciones, se puede realizar, como otros autores proponen, una supervisión del corpus generado hasta ese momento, corrigiendo los errores cometidos por el proceso. Parece el método más adecuado dada la diferencia de comportamiento de las palabras entre si.

La selección automática de fuentes de información está, en realidad, por desarrollar. Así mismo, falta comprobar que las palabras mantienen ese mismo comportamiento diferenciador en corpus distintos, para establecer definitivamente qué fuentes de información son las adecuadas para cada palabra, o mejor para cada sentido.

En nuestro grupo de investigación se está trabajando en muchas de esas tareas que están esperando la aportación de un sistema de WSD.

Y, por supuesto, nuestra intención es mostrar nuestras propuestas en el próximo SENSEVAL-3.

7.3 Producción científica

Las siguientes referencias son publicaciones previas del autor, directamente relacionadas con esta Tesis Doctoral:

- SUÁREZ, ARMANDO y ANDRÉS MONTOYO (**2001**). «Estudio de cooperación entre métodos de desambiguación léxica: Marcas de especificidad vs. máxima entropía», *Procesamiento Lenguaje Natural*, **27**(1), 207–214.

  En este artículo hicimos nuestro primer estudio sobre una posible cooperación entre los métodos marcas de especificidad, no supervisado, y máxima entropía. El estudio consistió en valorar el techo de acierto que se podría alcanzar si se construyera un clasificador óptimo. Este clasificador óptimo sería el que anota únicamente con las etiquetas correctas asignadas por uno u otro método.

- SUÁREZ, ARMANDO y MANUEL PALOMAR (**2001**). «Desambiguación del sentido y del dominio de las palabras con modelos de probabilidad de Máxima Entropía», *Revista Procesamiento del Lenguaje Natural*, **28**(1), 45–54.

  Uno de nuestros primeros trabajos sobre la ganancia en precisión al utilizar etiquetas de dominio en lugar de synsets.

- MONTOYO, ANDRÉS y ARMANDO SUÁREZ (**2001**). «The University of Alicante word sense disambiguation system», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 131–134, ACL-SIGLEX, Toulouse, France.

  Este trabajo describe el sistema combinado presentado en el SENSEVAL-2 que se basaba, como ha sido comentado en la sección 2.5.1, en utilizar máxima entropía para verbos y adjetivos, y marcas de especificidad para nombres.

- MONTOYO, ANDRÉS, ARMANDO SUÁREZ y MANUEL PALOMAR (**2002**). «Combining supervised-unsupervised methods for word sense disambiguation», en Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002*, vol. 2276 de *Lecture Notes in Computer Science*, págs. 156–164, Springer, Mexico City, Mexico

  Este artículo es una ampliación del anteriormente comentado (Montoyo y Suárez, 2001) en el que también se incluyó el algoritmo de densidad conceptual. Coincidió en la misma publicación con otro de Pedersen (2002b) en el que uno de sus resultados venía a confirmar nuestra experimentación.

- SUÁREZ, ARMANDO y MANUEL PALOMAR (**2002**). SUÁREZ, ARMANDO y MANUEL PALOMAR (2002). «Feature selection analysis for maximum entropy-based wsd», en Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002*, vol. 2276 de *Lecture Notes in Computer Science*, págs. 146–155, Springer, Mexico City, Mexico.

  Es un primer acercamiento a nuestra hipótesis de que distintas palabras necesitan distintos conjuntos de atributos de aprendizaje y una demostración empírica, utilizando el corpus DSO.

- SUÁREZ, ARMANDO y MANUEL PALOMAR (**2002**). «Improving feature selection analysis for maximum entropy-based wsd», en Elisabete Ranchhod y Nuno J. Mamede, editores, *PorTAL*, vol. 2389 de *Lecture Notes in Artificial Intelligence*, págs. 15–24, Springer.

  SUÁREZ, ARMANDO y MANUEL PALOMAR (**2002**). «Best Feature Selection for Maximum Entropy-Based Word Sense Disambiguation», en *Natural Language Processing and Information Systems: 7th International Conference on Applications of Natural Language to Information Systems, NLDB 2002*, vol. 2553 de *Lecture Notes in Computer Science*, págs. 213–217, Springer-Verlag Heidelberg, Stockholm, Sweden.

  Estos dos artículos son sucesivas ampliaciones del estudio anterior que nos permitieron reforzar nuestras conclusiones.

- SUÁREZ, ARMANDO y MANUEL PALOMAR (**2002**). «A maximum entropy-based word sense disambiguation system», en Hsin-Hsi Chen and Chin-Yew Lin, editors, *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, pages 960–966, Taipei, Taiwan, August 2002.

  Finalmente, la aplicación de la selección de atributos por prueba sistemática de conjuntos de característica. La selección parte de pruebas de validación cruzada sobre el conjunto de entrenamiento de SENSEVAL-2 para la muestra léxica en español, diferenciando por palabras y por POS. Los resultados se combinaron con marcas de especificidad para los nombres, por votación, alcanzando en esta categoría un empate en la primera posición, además de una segunda posición en la puntuación con todas las palabras. Es una evaluación posterior al propio SENSEVAL-2 y se comenta ampliamente en la sección 5.2.

- SUÁREZ, ARMANDO y MANUEL PALOMAR (**2002**). «Word sense vs. word domain disambiguation: a maximum entropy approach», en Petr Sojka, Ivan Kopecek y Karel Pala, editores, *TSD 2002*, vol. 2448 de *Lecture Notes in Artificial Intelligence*, págs. 131–138, Springer.

  Este trabajo amplía el ya comentado de Suárez y Palomar (2002a). Un estudio empírico más completo del impacto de la reducción de la polisemia en la clasificación.

- MONTOYO, ANDRÉS, ARMANDO SUÁREZ, GERMAN RIGAU y MANUEL PALOMAR (**2004**). «Combining Knowledge & corpus-based Word Sense Disambiguation Methods», informe interno, enviado al *Journal of Artificial Intelligence Research, JAIR* (en revisión).

  Este artículo resume los experimentos descritos en la sección 5.5, que consisten en establecer fases de predesambiguación entre marcas de especificidad y máxima entropía de forma que en la fase de desambiguación propiamente dicha, uno aproveche la información suministrada por el otro, como poda de sentidos posibles en el primero y como atributos en el segundo.

Otras publicaciones, donde los resultados de nuestro sistema de máxima entropía son utilizados como apoyo o como propuesta son enunciados a continuación:

- PALOMAR, MANUEL, MAXIMILIANO SAIZ-NOEDA, RAFAEL MUÑOZ, ARMANDO SUÁREZ y PATRICIO MARTÍNEZ-BARCO (**2000**). «PHORA: A system to solve the Anaphora in Spanish», en *Proceedings of Third Colloquium on Discurse Anaphora and Anaphor Resolution (DAARC'2000)*, págs. 206–211, Lancaster, UK.

- PALOMAR, M., M. SAIZ-NOEDA, R. MUÑOZ, A. SUÁREZ, P. MARTÍNEZ-BARCO y A. MONTOYO (**2001**). «PHORA: A NLP aystem for Spanish», en A. Gelbukh, editor, *Proceedings of 2nd International conference on Intelligent Text Processing and Computational Linguistics (CICLing-2001). Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, págs. 126–139, Springer-Verlag, Mexico City.

- SAIZ-NOEDA, MAXIMILIANO, ARMANDO SUÁREZ y MANUEL PALOMAR (**2001**). «Semantic pattern learning through maximum entropy-based wsd technique», en *Proceedings of CoNLL-2001*, págs. 23–29, Toulouse, France.

- MUÑOZ, RAFAEL, RUSLAN MITKOV, MANUEL PALOMAR, JESÚS PERAL, RICHARD EVANS, LIDIA MORENO, CONSTANTIN ORASAN, MAXIMILIANO SAIZ-NOEDA, ANTONIO FERRÁNDEZ, CATALINA BARBÚ, PATRICIO MARTÍNEZ-BARCO y ARMANDO

SUÁREZ (**2002**). «Bilingual Alignment of Anaphoric Expressions», en *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'2002)*, Las Palmas, Canary Islands, Spain.

A

Sistemas seleccionados de Senseval-2

Volver al Índice/Tornar a l'índex

Sistema	tipo	métodos	meta-métodos	recursos extra	info	comentarios
TALP	supervisado	LazyBoosting		WordNet WN Domains	local (+3-3) WNDomains WN lex-files	
Antwerp	supervisado	memory-based learning (TiMBL) rule induction (Ripper)	votación	SemCor		
JHU	supervisado	decision lists cosine-based model naive-Bayes			local gramatical	
CS224N	supervisado	naive-Bayes vector space memory based	votación votación ponderada máxima entropía			
duluth	supervisado	decision trees naive-Bayes			local	
KUNLP	supervisado	classification information model (CIM)				
UMD-SST	supervisado	support vector machines				
ehu-dlist	supervisado	decision lists		SemCor	local género selección atributos	
U. California	supervisado	Probabilistic Network Models One-sense-per-discourse Bayesian Networks* Maximum Likelihood Estimation*		SemCor Internet search engines	estructural (no posicional)	*se utilizan en la construcción de la red y el suavizado de parámetros
LIA-Sinequa	supervisado	Semantic Classification Trees (binary decision trees) similarity distance Hidden Markov Models		WordNet Semantic Classes (lexicographer files)	local + global	HMM para All-words y SCT para lexical-sample
SMU	mixto	instance-based pattern learning active feature selection	supervisado si suficiente cantidad de ejemplos	WordNet SemCor GenCor		
UA	mixto	Specification Marks Maximum Entropy	ME *verbos y adjetivos* SM nombres	WordNet	local keywords sintáctica WN relations	
UNED	No supervisado	mutual information hiponimia		Gutenberg Project WordNet		Uno de los sistemas es supervisado
Sussex	No supervisado	Selectional preferences Bayes rule One-sense-per-discourse Anaphora resolution		BNC WordNet		
IIT	No supervisado	Pattern matching		WordNet examples WordNet relations		
DIMAP	No supervisado			WordNet NODE	collocation patterns contextual clue words contextual overlap with definitions and examples topical area matches	es una herramienta para la creación y mantenimiento de lexicones para PLN
IRST	No supervisado	Domain vectors		SemCor WordNet Domains		
U. Sheffield	No supervisado	Anaphora resolution Simulated annealing	majority voting	WordNet ANLT lexicon		

Cuadro A.1. Descripción de sistemas seleccionados de SENSEVAL-2 (datos porcentuales)

B

Muestras de corpus

SEMCOR

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

DSO

```
ca01.db \#004 `` Only a relative handful of such reports was
received '' , the jury said , `` considering the widespread >>
interest 1 << in the election , the number of voters and the size
of this city '' .

ca01.db \#007 The grand jury commented on a number of other topics
 , among them the Atlanta and Fulton County purchasing departments
which it said `` are well operated and follow generally accepted
practices which inure to the best >> interest 5 << of both
governments '' .

ca06.db \#045 Sheets said that his proposed law would offer state
financing aid for the purchase of voting machines , enabling
counties to repay the loan over a 10-year period without >>
interest 4 << or charge .
```

SENSEVAL-2 *Testing data for English Lexical Sample*

```
<?xml version="1.0" encoding="iso-8859-1" ?> <!DOCTYPE corpus
SYSTEM
  "lexical-sample.dtd">
  <corpus lang='english'>

 <lexelt item="art.n">

 <instance id="art.40003" docsrc="bnc_A04_1181">
 <context>
 Whatever flickerings of potential this young tyro possesses, they cannot cover up the
 fact that he is a painter with the imagination of a retarded adolescent; no technical
 mastery; no intuitive feeling for pictorial space; no sensitivity towards, or grasp
 of, tradition; and a colour sense rather less than that of Congo, the chimpanzee who
 was taught (among other things) a crude responsiveness to colour harmonies by Desmond
 Morris in the late 1950s. However, potentially educable as a painter Schnabel may or
 may not be, his work is just not worthy of serious attention by anyone with a
 developed taste in this particular art form. [/p] [/quote] [/p] [p] Readers need also
 to be wary of the existence of special markets. The explosive prices for Teddy Bears
 in the last few years indicate how a [pb] market can be created, in this case by a
 mix of merit and nostalgia. What is clearly a dealers' market is often signalled by
 the invention of a brand name to group together a variety of material, perhaps rather
 disparate. Pop <head>Art</head>is an example.
 </context>
 </instance>
```

SENSEVAL-2 *Testing data for English All words task*

```
<text id="d00">
The
<head id="d00.s00.t01">art</head>
of
<head id="d00.s00.t03">change-ringing</head>
<head id="d00.s00.t04">is</head>
<head id="d00.s00.t05">peculiar</head>
to the <head id="d00.s00.t08">English</head>
,
and
,
like
<head id="d00.s00.t13">most</head>
<head id="d00.s00.t14">English</head>
<head id="d00.s00.t15">peculiarities</head>
,
<head id="d00.s00.t17">unintelligible</head>
to
the
<head id="d00.s00.t20">rest</head>
of
the
<head id="d00.s00.t23">world</head>
.
```

SENSEVAL-2 ***Training data for Spanish Lexical Sample***

```
<instance id="actuar.000000">
<answer instance="actuar.000000" senseid="2" />
<context>
Es que las majors <head>actúan</head> por
razones comerciales .
</context>
</instance>
<instance id="actuar.000001">
<answer instance="actuar.000001" senseid="2"/>
<context>
Ha dicho el reverendo : " Son una minoría que mete mucho
ruido y entre ellos hay sindicalistas , por los modos de
<head>actuar</head> " .
</context>
</instance>
```

Interest Corpus

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE corpus SYSTEM  "lexical-sample.dtd">
<corpus lang="en">
<lexelt item="interest-n">

<instanceid="interest-n.int1"> <answer instance="interest-n.int1" senseid="6"/>
<context>
<s> yields  on  money-market mutual funds continued to slide , amid  signs  that  portfolio managers
expect  further declines  in   <head>interest</head>  rates  . </s>

</context>
</instance>

<instance id="interest-n.int2">
<answer instance="interest-n.int2" senseid="6"/>
<context>
<s> longer maturities  are thought to indicate  declining <head>interest</head>  rates  because
they  permit  portfolio managers  to retain relatively  higher rates  for  a longer period .</s>

</context>
</instance>
```

Este corpus fue convertido al formato de SENSEVAL-2 por Ted Pedersen.

Serve corpus

```
<corpus lang="en">
<lexelt item="serve-v">
<instance id="serve-v.aphb_51905969_4083">
<answer instance="serve-v.aphb_51905969_4083" senseid="10"/>
<context>
<s>Some tart fruits mixed with greens make a nice contrast with rich meat dishes ( see Orange and Onion
Salad , page 111 ) , but if you like to follow the meat course with sweet fruit , it seems wiser to
<head>serve</head>  it plain with a good sharp cheese and let it take the place of a sweet or dessert
course . </s> <s>If you insist on serving fruit as a salad , don't cut it into cubes and mix it up .
</s>

</context>
</instance>
<instance id="serve-v.aphb_27702628_1678">
<answer instance="serve-v.aphb_27702628_1678" senseid="10"/>
<context>
<s>To increase amount of stuffing , preserve ratio of half as much mushrooms as bread . </s>
<s>Capon stuffed in this manner would most likely be  <head>served</head>  with pan-fried potatoes ,
broccoli , or cauliflower browned in oil .   </s>

</context>
</instance>
```

Este corpus fue convertido al formato de SENSEVAL-2 por Ted Pedersen.

Hard corpus

```
<corpus lang="en">
<lexelt item="hard-a">
<instance id="hard-a.sjm-274_1:">
<answer instance="hard-a.sjm-274_1:" senseid="HARD1"/>
<context>
<s> `` He may lose all popular support ,  but someone has to kill him to defeat him and that 's
<head>HARD</head>  to do. '' </s>

</context>
</instance>
<instance id="hard-a.sjm-014_1:">
<answer instance="hard-a.sjm-014_1:" senseid="HARD1"/>
<context>
<s> Clever White House `` spin doctors '' are having a  <head>HARD</head>  time helping President
Bush explain away the economic bashing that low-and middle-income workers are taking these days . </s>

</context>
</instance>
<instance id="hard-a.sjm-128_1:">
<answer instance="hard-a.sjm-128_1:" senseid="HARD1"/>
<context>
<s> I find it  <head>HARD</head>  to believe that the Sacramento River will ever be quite the
same , although I certainly wish that I'm wrong . </s>

</context>
</instance>
```

Este corpus fue convertido al formato de SENSEVAL-2 por Ted Pedersen.

Line corpus

```
<corpus lang="en">
<lexelt item="line-n">
<instance id="line-n.w7_010:888:">
<answer instance="line-n.w7_010:888:" senseid="cord"/>
<context>
<s> The company argued that its foreman needn't have told the worker not to move the plank to
which his lifeline was tied because "that comes with common sense." </s> <@> </p> <@> <p> <@>
<s> The commission noted, however, that Dellovade hadn't instructed its employees on how to
secure their lifelines and didn't heed a federal inspector's earlier suggestion that the company
install special safety  <head>lines</head>  inside the A-frame structure it was building. </s>
</context>
</instance>
<instance id="line-n.w7_034:5894:">
<answer instance="line-n.w7_034:5894:" senseid="cord"/>
<context>
<s> The set, designed by Mr. Hall's longtime associate Eugene Lee, has the audience divided in
half, facing a central playing area. </s> <@> <s> Off to one side -- representing the "have-nots"
of Louisiana -- is a broken-down shack with a woodpile and a wash  <head>line</head> . </s>
</context>
</instance>
<instance id="line-n.w7_038:7434:">
<answer instance="line-n.w7_038:7434:" senseid="cord"/>
<context>
<s> The new technology represents a considerable savings over conventional offshore production
technology. </s> <@> </p> <@> <p> <@> <s> The foundation of the project, a "tension-leg well
platform," will be a floating structure that is anchored to the seabed by steel mooring
<head>lines</head> , a design that requires far less steel than conventional platforms do. </s>
</context>
</instance>
```

Este corpus fue convertido al formato de SENSEVAL-2 por Ted Pedersen.

C

Muestras de salidas de analizadores sintácticos

Tree-tagger: este etiquetador suministra información sobre categoría gramatical y lema (segunda y tercera columnas, respectivamente)

```
The       DT   the
cat       NN   cat
is        VBZ  be
sleeping  VBG  sleep
on        IN   on
the       DT   the
zinc      NN   zinc
tile      NN   tile
roof      NN   roof
.         SENT .
```

Connexor: aparte del lema y su categoría (tercera y quinta columna), representa el árbol sintáctico en la cuarta columna mediante referencias a la numeración de cada línea, además de ofrecer información sobre el rol gramatical de la palabra dentro de la oración: si núcleo verbal (main), sujeto (subj), determinante (det), etc. Las etiquetas finales ofrecen información sobre género, número, etc.

```
1  El       el       det:>2  &DN> DET MSC SG
2  gato     gato     subj:>3 &NH N MSC SG
3  duerme   dormir   main:>0 &+FM V IND PRES SG3
4  sobre    sobre    pm:>6   &PM> PREP
5  el       el       det:>6  &DN> DET MSC SG
6  tejado   tejado   ha:>3   &NH N MSC SG
7  de       de       pm:>8   &PM> PREP
8  zinc     zinc     mod:>6  &NH N MSC SG
9  .        .
10 <s>      <s>
```

Minipar: al igual que el anterior, el árbol sintáctico se representa con referencias a la numeración de la primera columna, y también muestra información sintáctica compleja. Cuando el lema coincide con la palabra tal y como está escrita en la frase, en la columna

del lema nos encontramos el carácter '~'. Además, segmenta la frase en cláusulas (numeradas con Ex), y es capaz de detectar ciertas palabras compuestas (es el ejemplo de tile roof que se muestra como lema de roof).

```
E0(# fin C   *                                  (atts (3sg +)
                                                (tense present)))
1 (The       ~ Det          2  det              (gov cat))
2 (cat       ~ N            4  s                (gov sleep))
3 (is        be be          4  be               (gov sleep)
                           (atts (3sg +) (plu -) (per 3)
                                                (tense present)))
4 (sleeping  sleep V        E0 i                (gov fin)
                                          (atts (3sg+) (plu -)
                                                (tense present)))
E2 (#        cat N          4  subj             (gov sleep)
                                                (antecedent 2))
5 (on        ~ Prep         4  mod              (gov sleep))
6 (the       ~ Det          9  det              (gov "tile roof"))
7 (zinc      ~ N            9  nn               (gov "tile roof"))
8 (tile      ~ U            9  lex-mod          (gov "tile roof"))
9 (roof      "tile roof" N  5  pcomp-n          (gov on)) )
```

Las dos siguientes referencias se exponen aquí puesto que dentro del subgrupo de WSD del GPLSI también han sido utilizados.

Maco+: (Carmona et al., 1998) *Morphological Analizer Corpus-Oriented.* La herramienta tokeniza el texto y realiza y produce como salida todas las posibles interpretaciones morfológicas para cada token. Es capaz de reconocer y tratar números, nombres propios, puntuación, fechas, abreviaturas y palabras compuestas, y acepta texto no restringido como entrada.

```
El      el TDMS0
gato    gato NCMS000
duerme  dormir VMIP3S0 dormir VMMP2S0
sobre   sobrar VMMP3S0 sobrar VMSP1S0
        sobrar VMSP3S0 sobre NCMS000
        sobre SPS00
el      el TDMS0
tejado  tejado NCMS000 tejar VMP00SM
de      de NCFS000 de SPS00
zinc    zinc NCMS000
.       . Fp
```

Tacat Parser: (Atserias y Rodríguez, 1998). Este analizador toma como entrada la salida de MACO+ (o la de cualquier etiquetador) y produce un análisis sintáctico. Usa gramáticas CFG, las

cuales producen tanto un análisis completo como parcial y reconocimiento de *chunks*. Hay versiones para catalán, español e inglés.

Volver al índice/Tornar a l'índex

D

Cuadros adicionales

Los cuadros que aquí se ubican contienen información adicional que se ha considerado conveniente separar del texto, por su extensión, ya que no son cruciales para seguir la exposición del capítulo correspondiente.

		0.8-0.0						0.8–0.4					
atrib1	atrib2	25	50	75	100	125	150	25	50	75	100	125	150
0mCDdv	McBbrW	90,6						87,0	84,2	82,7	81,5	79,4	77,9
0mCDdv	—	88,1	87,8	87,9	87,8	87,5		69,8	68,1	66,0	65,5	65,6	65,6
—	McBbrW	90,6						73,2	70,5	70,1	69,8	70,1	70,1
0mMCBDW	0mMCcBDbdrW	98,3	93,9	91,2	91,5	91,3	91,5	80,4	78,1	76,8	76,5	74,9	74,4
0mMCBDW	—	94,6	95,1	95,5	92,5	92,5	92,1	77,6	76,4	76,6	73,9	74,3	74,1
—	0mMCcBDbdrW	95,3	95,9	92,3	91,5	91,5	91,7	78,8	74,6	70,4	67,5	65,9	65,5
LSBPDr	0WSCMv	95,0	95,4	94,9	93,3	93,9	93,5	81,2	79,7	77,7	74,9	73,8	72,0
LSBPDr	—	93,0	91,1	88,6	87,8	88,3	87,2	74,1	69,8	65,2	63,1	62,0	60,9
—	0WSCMv	93,0	93,2	93,7	94,5	94,3	94,4	72,4	71,5	69,5	67,5	66,1	65,2
mcBSp	0MCbls	98,2	95,6	94,1	93,2	93,1	92,7	76,8	77,5	75,7	73,3	71,5	71,3
mcBSp	—	94,9	93,2	93,2	93,4	93,3	92,9	67,9	67,0	66,1	64,6	62,8	60,5
—	0MCbls	93,7	94,1	92,9	92,7	92,7	92,5	76,1	73,3	71,0	69,6	68,5	66,9

Cuadro D.1. Validez del método: precisiones

El cuadro D.13 muestra los valores acumulados de precisión (P), cobertura (C) y F1 para cada palabra en los sucesivos reentrenamientos (Rx) y clasificación final (F). La tabla está ordenada por categoría y por precisión descendente de la clasificación final.

D Cuadros adicionales

atrib1	atrib2	0.8-0.0 25	50	75	100	125	150	0.8–0.4 25	50	75	100	125	150
0mCDdv	McBbrW	12,5						32,0	32,0	32,0	32,8	32,8	32,8
0mCDdv	—	29,7	29,7	29,7	29,7	29,7		52,3	57,0	57,8	57,8	58,6	58,6
—	McBbrW	19,5						44,5	45,3	45,3	45,3	45,3	45,3
0mMCBDW	0mMCcBDbdrW	14,1	14,1	14,1	14,1	14,1	14,1	43,0	43,0	43,0	43,0	43,0	43,0
0mMCBDW	—	17,2	17,2	17,2	17,2	17,2	17,2	46,1	46,1	46,1	46,1	46,1	46,9
—	0mMCcBDbdrW	18,8	18,8	18,8	18,8	18,8	18,8	44,5	45,3	46,9	46,9	46,9	46,9
LSBPDr	0WSCMv	11,7	11,7	11,7	11,7	11,7	11,7	43,0	43,0	43,0	43,0	43,0	43,0
LSBPDr	—	17,2	17,2	17,2	17,2	17,2	17,2	43,0	43,8	43,8	43,8	43,8	43,8
—	0WSCMv	13,3	13,3	13,3	13,3	13,3	13,3	45,3	46,9	46,9	46,9	47,7	47,7
mcBSp	0MCbls	16,4	16,4	16,4	16,4	16,4	16,4	46,1	46,9	46,9	46,9	46,9	46,9
mcBSp	—	21,1	21,1	21,1	21,1	21,1	21,1	50,0	51,6	51,6	51,6	52,3	52,3
—	0MCbls	21,1	21,1	21,1	21,1	21,1	21,1	48,4	50,8	50,8	51,6	51,6	51,6

Cuadro D.2. Validez del método: cobertura de sentidos

atrib1	atrib2	0.8-0.0 25	50	75	100	125	150	0.8–0.4 25	50	75	100	125	150
0mCDdv	McBbrW	0,5						5,9	9,6	11,9	14,7	16,5	17,7
0mCDdv	—	9,3	11,9	12,3	12,7	13,8		22,7	40,4	52,7	59,6	65,1	65,8
—	McBbrW	0,7						13,9	24,2	29,5	30,6	31,1	31,1
0mMCBDW	0mMCcBDbdrW	1,0	1,4	1,8	2,5	2,8	3,1	9,1	17,2	23,2	30,1	34,0	36,8
0mMCBDW	—	1,6	1,9	2,3	2,7	3,8	4,1	11,4	20,6	28,9	35,2	39,1	41,8
—	0mMCcBDbdrW	2,4	4,0	5,0	5,3	5,6	6,0	11,1	22,7	35,0	44,0	49,8	55,1
LSBPDr	0WSCMv	1,4	2,1	2,5	3,0	3,4	3,7	5,9	9,6	11,9	14,7	16,5	17,7
LSBPDr	—	2,0	3,1	4,4	5,3	6,4	7,2	7,3	14,5	21,9	29,1	36,2	43,5
—	0WSCMv	2,0	3,1	4,0	5,2	5,9	6,5	8,2	15,9	23,6	31,6	38,8	46,6
mcBSp	0MCbls	1,4	2,0	2,2	2,4	2,6	2,8	7,0	13,0	18,7	24,5	30,0	34,9
mcBSp	—	1,9	2,9	4,2	5,1	5,8	6,2	7,9	15,0	22,4	29,5	36,4	43,4
—	0MCbls	2,3	3,9	5,0	5,7	6,4	7,2	8,5	16,5	24,0	31,3	38,2	45,3

Cuadro D.3. Validez del método: cobertura absoluta

atrib1	atrib2	0.8-0.0 25	50	75	100	125	150	0.8–0.4 25	50	75	100	125	150
0mCDdv	McBbrW	0,8						9,7	14,7	17,6	20,9	22,4	23,5
0mCDdv	—	15,0	18,6	19,2	19,7	21,2		25,8	39,2	45,5	48,9	51,7	52,1
—	McBbrW	1,3						17,8	27,5	31,9	32,7	33,3	33,3
0mMCBDW	0mMCcBDbdrW	2,0	2,6	3,2	4,5	4,9	5,4	13,4	22,9	28,9	35,4	38,0	40,0
0mMCBDW	—	3,0	3,6	4,3	4,8	6,8	7,2						
—	0mMCcBDbdrW	4,4	7,3	8,9	9,2	9,7	10,3	15,7	27,6	36,5	41,2	43,8	46,5
LSBPDr	0WSCMv	2,6	3,9	4,7	5,4	6,2	6,7	9,7	16,8	22,8	27,1	31,3	34,8
LSBPDr	—	3,6	5,5	7,5	8,9	10,5	11,6	10,1	17,7	23,5	28,4	32,9	37,0
—	0WSCMv	3,6	5,5	7,1	9,3	10,5	11,6	11,0	19,6	26,5	32,4	37,0	41,4
mcBSp	0MCbls	2,8	3,7	4,0	4,4	4,8	5,1	10,1	17,8	23,9	28,9	33,0	36,9
mcBSp	—	3,5	5,3	7,5	9,0	10,3	10,9	9,9	17,5	24,2	29,5	33,5	36,6
—	0MCbls	4,3	7,1	8,8	10,0	11,1	12,4	11,9	20,8	27,5	33,2	37,9	41,8

Cuadro D.4. Validez del método: F1

atrib1	atrib2	0.8-0.0 25	50	75	100	125	150	0.8-0.4 25	50	75	100	125	150
0mCDdv	McBbrW	86,6	86,9	84,7	83,4	83,3	83,5	69,0	65,3	62,0	59,8	59,3	58,9
0mMCBDW	0mMCcBDbdrW	95,0	92,5	91,2	92,2	92,8	91,8	77,8	74,3	71,8	71,0	69,4	68,6
LSBPDr	0WSCMv	91,9	92,7	90,1	89,4	88,7	88,9	71,9	71,3	67,7	65,6	63,5	63,1
mcBSp	0MCbls	91,6	92,1	92,3	92,3	91,2	90,2	70,1	69,0	66,6	64,3	62,6	61,5

Cuadro D.5. Coentrenamiento: precisiones

atrib1	atrib2	0.8-0.0 25	50	75	100	125	150	0.8-0.4 25	50	75	100	125	150
0mCDdv	McBbrW	32,8	32,8	32,8	32,8	32,8	32,8	51,6	57,8	58,6	60,2	61,7	63,3
0mMCBDW	0mMCcBDbdrW	19,5	19,5	19,5	19,5	19,5	19,5	47,7	48,4	48,4	48,4	48,4	48,4
LSBPDr	0WSCMv	17,2	17,2	17,2	17,2	17,2	17,2	43,0	44,5	45,3	45,3	45,3	45,3
mcBSp	0MCbls	21,9	21,9	21,9	21,9	21,9	21,9	50,0	53,9	53,9	53,9	53,9	53,9

Cuadro D.6. Coentrenamiento: coberturas de sentidos

atrib1	atrib2	0.8-0.0 25	50	75	100	125	150	0.8-0.4 25	50	75	100	125	150
0mCDdv	McBbrW	7,8	11,0	14,5	15,8	16,1	16,7	15,0	29,0	43,0	54,6	66,2	73,5
0mMCBDW	0mMCcBDbdrW	2,1	2,5	2,9	3,4	3,8	4,2	12,2	22,5	31,9	40,5	46,3	51,5
LSBPDr	0WSCMv	2,0	3,2	4,4	5,6	6,5	7,4	7,4	14,4	21,8	28,6	35,2	42,3
mcBSp	0MCbls	2,4	3,8	4,9	5,8	6,8	7,7	7,6	15,3	22,3	29,3	36,2	42,9

Cuadro D.7. Coentrenamiento: coberturas absolutas

D Cuadros adicionales

			Precisión				F1			
			0.8-0.0		0.8-0.4		0.8-0.0		0.8-0.4	
atrib1	atrib2	TIPO	25	150	25	150	25	150	25	150
0mCDdv	McBbrW	COENT	86,6	83,5	69,0	58,9	12,5	23,9	18,0	49,9
		REENT	90,6	90,6	87,0	77,9	0,8	0,8	9,7	23,5
0mMCBDW	0mMCcBDbdrW	COENT	95,0	91,8	77,8	68,6	3,9	7,4	17,0	46,6
		REENT	98,3	91,5	80,4	74,4	2,0	5,4	13,4	40,0
LSBPDr	0WSCMv	COENT	91,9	88,9	71,9	63,1	3,6	12,3	9,9	37,5
		REENT	95,0	93,5	81,2	72,0	2,6	6,7	9,7	34,8
mcBSp	0MCbls	COENT	91,6	90,2	70,1	61,5	4,2	12,9	9,9	36,9
		REENT	98,2	92,7	76,8	71,3	2,8	5,1	10,1	36,9
			Cobertura sentidos				Cobertura absoluta			
			0.8-0.0		0.8-0.4		0.8-0.0		0.8-0.4	
atrib1	atrib2	TIPO	25	150	25	150	25	150	25	150
0mCDdv	McBbrW	COENT	32,8	32,8	51,6	63,3	7,8	16,7	15,0	73,5
		REENT	12,5	12,5	32,0	32,8	0,5	0,5	5,9	17,7
0mMCBDW	0mMCcBDbdrW	COENT	19,5	19,5	47,7	48,4	2,1	4,2	12,2	51,5
		REENT	14,1	14,1	43,0	43,0	1,0	3,1	9,1	36,8
LSBPDr	0WSCMv	COENT	17,2	17,2	43,0	45,3	2,0	7,4	7,4	42,3
		REENT	11,7	11,7	35,9	37,5	1,4	3,7	6,4	31,8
mcBSp	0MCbls	COENT	21,9	21,9	50,0	53,9	2,4	7,7	7,6	42,9
		REENT	16,4	16,4	46,1	46,9	1,4	2,8	7,0	34,9

Cuadro D.8. Coentrenamiento y reentrenamiento: comparación de ejecuciones en las iteraciones 25 y 150

pos	palabra	0	b	B	c	C	d	D	l	L	o	p	P	r	s	S	v	W
N	autoridad	90,0	100,0	100,0	100,0	100,0	40,0	40,0	100,0	100,0	58,5	48,4	55,7	65,1	58,4	61,6	57,1	100,0
	bomba	71,1	100,0	100,0	100,0	100,0	88,9	88,9	100,0	100,0	64,9	63,2	67,1	57,4	56,0	66,7	72,7	100,0
	canal	31,8	93,9	93,9	100,0	100,0	55,6	50,0	65,3	67,3	47,6	38,8	0,0	21,4	48,3	48,2	41,2	94,3
	circuito	0,0	75,0	75,0	100,0	100,0	0,0	0,0	66,7	66,7	45,8	31,9	34,8	20,0	41,8	36,8	42,9	72,7
	corazón	59,8	100,0	100,0	100,0	100,0	75,0	78,3	73,3	71,0	65,9	54,0	60,6	52,2	69,3	67,9	52,9	100,0
	corona	77,2	50,0	50,0	50,0	50,0	0,0	0,0	40,0	40,0	65,2	56,1	56,8	70,8	62,0	58,1	50,0	55,6
	gracia	82,1	100,0	100,0	100,0	100,0	87,5	87,1	56,4	60,0	65,9	36,4	52,6	80,6	59,0	53,9	81,5	76,2
	grano	64,3	75,0	75,0	75,0	75,0	0,0	0,0	53,3	53,3	64,0	43,6	51,9	66,7	66,0	56,0	0,0	66,7
	hermano	56,9	100,0	100,0	100,0	100,0	71,4	70,0	50,0	50,0	53,8	63,5	57,1	75,6	57,6	61,2	40,0	100,0
	masa	64,0	100,0	100,0	100,0	100,0	90,5	90,5	84,0	84,0	74,3	48,7	40,8	39,3	69,3	65,3	66,7	94,7
	naturaleza	100,0	100,0	100,0	100,0	100,0	0,0	0,0	57,7	50,0	55,1	44,7	57,5	46,3	53,6	48,4	10,0	71,4
	operación	0,0	63,6	63,6	100,0	100,0	60,0	60,0	76,7	76,7	41,3	29,6	44,0	0,0	39,5	41,9	52,9	71,4
	órgano	58,4	88,9	88,9	100,0	100,0	33,3	55,6	87,8	83,3	56,2	42,2	50,4	72,5	57,3	56,8	44,4	93,1
	partido	54,3	88,9	88,9	100,0	100,0	50,0	50,0	82,4	82,4	75,8	72,5	61,8	58,6	77,5	69,6	78,3	100,0
	pasaje	68,4	50,0	50,0	100,0	100,0	66,7	66,7	83,3	87,5	52,8	49,1	52,7	50,0	52,8	61,8	75,0	83,3
	programa	41,7	83,3	78,6	100,0	100,0	0,0	0,0	71,4	84,0	50,0	44,6	50,0	48,6	57,5	50,0	68,8	87,5
	tabla	60,3	100,0	100,0	100,0	100,0	92,9	92,9	92,9	92,9	56,2	40,5	55,3	48,0	56,3	45,3	100,0	100,0
V	actuar	39,4	25,0	25,0	50,0	50,0	37,5	37,5	30,6	32,4	38,7	35,4	28,6	0,0	48,7	45,8	33,3	52,6
	apoyar	66,4	68,8	68,8	71,4	71,4	58,3	57,1	77,6	77,6	67,5	62,4	63,5	72,5	65,8	60,9	62,5	75,9
	apuntar	64,9	54,5	54,5	100,0	100,0	0,0	0,0	69,0	66,7	59,2	51,1	44,3	52,3	68,2	65,7	71,4	83,3
	clavar	38,2	83,3	83,3	100,0	100,0	100,0	100,0	65,4	64,0	38,1	46,8	47,8	50,0	43,8	46,0	66,7	84,2
	conducir	41,7	100,0	100,0	100,0	100,0	80,0	80,0	79,3	81,5	55,4	37,5	40,9	22,2	51,5	60,0	12,5	76,0
	copiar	29,2	100,0	100,0	0,0	0,0	11,1	11,1	28,1	32,3	36,4	23,7	0,0	28,6	32,0	34,8	11,1	63,6
	coronar	48,7	89,6	89,6	96,9	96,9	68,4	68,4	77,8	71,6	54,4	46,2	22,2	53,6	62,9	70,3	70,6	82,1
	explotar	53,3	100,0	100,0	100,0	100,0	100,0	100,0	60,0	57,9	50,0	34,5	33,3	33,3	48,5	52,0	100,0	70,4
	saltar	0,0	75,0	75,0	60,0	60,0	28,6	28,6	46,5	55,6	33,9	20,0	30,8	0,0	39,6	36,6	16,7	80,0
	tocar	34,0	97,5	94,7	93,5	96,7	27,3	33,3	56,1	61,1	47,5	54,4	0,0	51,2	49,6	43,8	36,4	78,7
	usar	70,0	46,7	46,7	50,0	50,0	60,0	60,0	61,4	65,1	60,9	67,0	68,5	64,2	63,0	67,3	50,0	77,8
	vencer	61,5	77,8	77,8	90,0	90,0	89,5	85,0	72,3	72,3	66,7	56,6	60,7	56,3	68,3	69,7	81,0	74,3

Cuadro D.9. 3fcv corpus entrenamiento SENSEVAL-2 español: precisión por grupos de atributos y palabras

pos	palabra	0	b	B	c	C	d	D	l	L	o	p	P	r	s	S	v	W
N	autoridad	30,7	2,3	2,3	2,3	2,3	2,3	2,3	5,7	5,7	54,5	35,2	38,6	31,8	51,1	51,1	4,5	4,5
	bomba	71,1	25,0	25,0	19,7	19,7	10,5	10,5	35,5	35,5	63,2	63,2	67,1	35,5	55,3	65,8	10,5	22,4
	canal	12,2	27,0	27,0	24,3	24,3	8,7	8,7	27,8	28,7	33,9	22,6	0,0	2,6	36,5	35,7	6,1	28,7
	circuito	0,0	8,1	8,1	8,1	8,1	0,0	0,0	16,2	16,2	36,5	20,3	10,8	2,7	31,1	18,9	4,1	10,8
	corazón	58,6	9,1	9,1	8,1	8,1	15,2	18,2	22,2	22,2	60,6	47,5	57,6	24,2	61,6	57,6	9,1	21,2
	corona	77,2	2,5	2,5	2,5	2,5	0,0	0,0	5,1	5,1	57,0	46,8	53,2	43,0	55,7	54,4	2,5	6,3
	gracia	23,2	21,2	21,2	16,2	16,2	28,3	27,3	22,2	24,2	54,5	24,2	10,1	25,3	46,5	41,4	22,2	16,2
	grano	64,3	10,7	10,7	10,7	10,7	0,0	0,0	14,3	14,3	57,1	42,9	48,2	50,0	58,9	50,0	0,0	14,3
	hermano	42,3	2,6	2,6	2,6	2,6	6,4	9,0	5,1	5,1	44,9	60,3	56,4	39,7	48,7	52,6	2,6	5,1
	masa	53,3	14,4	14,4	14,4	14,4	21,1	21,1	23,3	23,3	61,1	42,2	22,2	12,2	57,8	52,2	8,9	20,0
	naturaleza	6,3	4,5	4,5	1,8	1,8	0,0	0,0	13,5	11,7	44,1	30,6	20,7	17,1	40,5	39,6	0,9	9,0
	operación	0,0	7,4	7,4	4,2	4,2	12,6	12,6	24,2	24,2	32,6	16,8	11,6	0,0	31,6	27,4	9,5	10,5
	órgano	39,7	12,2	12,2	10,7	10,7	1,5	3,8	27,5	26,7	51,9	37,4	50,4	28,2	54,2	51,1	3,1	20,6
	partido	43,1	15,7	15,7	11,8	11,8	6,9	6,9	27,5	27,5	73,5	72,5	61,8	40,2	77,5	69,6	17,6	15,7
	pasaje	18,3	2,8	2,8	2,8	2,8	5,6	5,6	7,0	9,9	39,4	36,6	40,8	32,4	39,4	47,9	8,5	7,0
	programa	10,5	10,5	11,6	14,7	14,7	0,0	0,0	21,1	22,1	38,9	34,7	47,4	18,9	48,4	44,2	11,6	14,7
	tabla	60,3	5,1	5,1	3,8	3,8	16,7	16,7	16,7	16,7	52,6	38,5	53,8	30,8	51,3	37,2	16,7	11,5
V	actuar	13,0	2,0	2,0	2,0	2,0	3,0	3,0	11,0	11,0	29,0	17,0	2,0	0,0	37,0	27,0	3,0	10,0
	apoyar	51,8	8,0	8,0	7,3	7,3	5,1	5,8	27,7	27,7	59,1	56,9	63,5	36,5	56,2	56,9	7,3	16,1
	apuntar	44,4	4,2	4,2	4,2	4,2	0,0	0,0	20,4	19,7	40,8	32,4	35,9	16,2	51,4	48,6	3,5	17,6
	clavar	24,1	11,5	11,5	11,5	11,5	8,0	8,0	19,5	18,4	27,6	33,3	37,9	9,2	32,2	33,3	6,9	18,4
	conducir	10,4	17,7	17,7	17,7	17,7	8,3	8,3	24,0	22,9	42,7	15,6	9,4	2,1	35,4	28,1	1,0	19,8
	copiar	7,4	3,2	3,2	0,0	0,0	1,1	1,1	9,6	10,6	25,5	9,6	0,0	4,3	25,5	25,5	1,1	7,4
	coronar	22,4	25,3	25,3	18,2	18,2	7,6	7,6	32,9	28,2	43,5	14,1	1,2	8,8	48,8	41,8	7,1	27,1
	explotar	17,4	10,9	10,9	6,5	6,5	7,6	7,6	22,8	23,9	38,0	20,7	7,6	4,3	35,9	28,3	7,6	20,7
	saltar	0,0	6,0	6,0	3,0	3,0	2,0	2,0	20,0	20,0	19,0	5,0	4,0	0,0	19,0	15,0	2,0	20,0
	tocar	9,9	24,1	22,2	17,9	17,9	1,9	2,5	28,4	27,2	34,6	19,1	0,0	13,0	35,2	30,2	2,5	22,8
	usar	56,8	6,3	6,3	3,6	3,6	5,4	5,4	24,3	25,2	50,5	64,0	68,5	30,6	56,8	59,5	5,4	18,9
	vencer	40,7	17,8	17,8	15,3	15,3	14,4	14,4	28,8	28,8	59,3	50,8	60,2	22,9	58,5	58,5	14,4	22,0

Cuadro D.10. 3fcv corpus entrenamiento SENSEVAL-2 español: cobertura por grupos de atributos y palabras

pos	palabra	0	b	B	c	C	d	D	l	L	o	p	P	r	s	S	v	W
N	autoridad	45,8	4,4	4,4	4,4	4,4	4,3	4,3	10,8	10,8	56,5	40,8	45,6	42,7	54,5	55,9	8,4	8,7
	bomba	71,1	40,0	40,0	33,0	33,0	18,8	18,8	52,4	52,4	64,0	63,2	67,1	43,9	55,6	66,2	18,4	36,6
	canal	17,6	41,9	41,9	39,2	39,2	15,0	14,8	39,0	40,2	39,6	28,6	0,0	4,7	41,6	41,0	10,6	44,0
	circuito	0,0	14,6	14,6	15,0	15,0	0,0	0,0	26,1	26,1	40,6	24,8	16,5	4,8	35,7	25,0	7,4	18,8
	corazón	59,2	16,7	16,7	15,0	15,0	25,2	29,5	34,1	33,8	63,2	50,5	59,1	33,1	65,2	62,3	15,5	35,0
	corona	77,2	4,8	4,8	4,8	4,8	0,0	0,0	9,0	9,0	60,8	51,0	54,9	53,5	58,7	56,2	4,8	11,4
	gracia	36,2	35,0	35,0	27,8	27,8	42,7	41,5	31,9	34,5	59,7	29,1	16,9	38,5	52,0	46,9	34,9	26,7
	grano	64,3	18,8	18,8	18,8	18,8	0,0	0,0	22,5	22,5	60,4	43,2	50,0	57,1	62,3	52,8	0,0	23,5
	hermano	48,5	5,0	5,0	5,0	5,0	11,8	15,9	9,3	9,3	49,0	61,8	56,8	52,1	52,8	56,6	4,8	9,8
	masa	58,2	25,2	25,2	25,2	25,2	34,2	34,2	36,5	36,5	67,1	45,2	28,8	18,6	63,0	58,0	15,7	33,0
	naturaleza	11,9	8,6	8,6	3,5	3,5	0,0	0,0	21,9	19,0	49,0	36,4	30,5	25,0	46,2	43,6	1,7	16,0
	operación	0,0	13,2	13,2	8,1	8,1	20,9	20,9	36,8	36,8	36,5	21,5	18,3	0,0	35,1	33,1	16,1	18,3
	órgano	47,3	21,5	21,5	19,3	19,3	2,9	7,1	41,9	40,5	54,0	39,7	50,4	40,7	55,7	53,8	5,7	33,8
	partido	48,1	26,7	26,7	21,1	21,1	12,1	12,1	41,2	41,2	74,6	72,5	61,8	47,7	77,5	69,6	28,8	27,1
	pasaje	28,9	5,3	5,3	5,5	5,5	10,4	10,4	13,0	17,7	45,2	41,9	46,0	39,3	45,2	54,0	15,2	13,0
	programa	16,8	18,7	20,2	25,7	25,7	0,0	0,0	32,5	35,0	43,8	39,1	48,6	27,3	52,6	46,9	19,8	25,2
	tabla	60,3	9,8	9,8	7,4	7,4	28,3	28,3	28,3	28,3	54,3	39,5	54,5	37,5	53,7	40,8	28,6	20,7
V	actuar	19,5	3,7	3,7	3,8	3,8	5,6	5,6	16,2	16,4	33,1	23,0	3,7	0,0	42,0	34,0	5,5	16,8
	apoyar	58,2	14,4	14,4	13,2	13,2	9,4	10,6	40,9	40,9	63,0	59,5	63,5	48,5	60,6	58,9	13,1	26,5
	apuntar	52,7	7,8	7,8	8,1	8,1	0,0	0,0	31,5	30,4	48,3	39,7	39,7	24,7	58,6	55,9	6,7	29,1
	clavar	29,6	20,2	20,2	20,6	20,6	14,9	14,9	30,1	28,6	32,0	38,9	42,3	15,5	37,1	38,7	12,5	30,2
	conducir	16,7	30,1	30,1	30,1	30,1	15,1	15,1	36,8	35,8	48,2	22,1	15,3	3,8	42,0	38,3	1,9	31,4
	copiar	11,9	6,2	6,2	0,0	0,0	1,9	1,9	14,3	16,0	30,0	13,6	0,0	7,4	28,4	29,4	1,9	13,3
	coronar	30,6	39,4	39,4	30,7	30,7	13,8	13,8	46,3	40,5	48,4	21,6	2,2	15,2	55,0	52,4	12,8	40,7
	explotar	26,2	19,6	19,6	12,2	12,2	14,1	14,1	33,1	33,8	43,2	25,9	12,4	7,7	41,3	36,6	14,1	31,9
	saltar	0,0	11,1	11,1	5,7	5,7	3,7	3,7	28,0	29,4	24,4	8,0	7,1	0,0	25,7	21,3	3,6	32,0
	tocar	15,3	38,6	36,0	30,1	30,2	3,5	4,6	37,7	37,6	40,0	28,3	0,0	20,7	41,2	35,8	4,6	35,4
	usar	62,7	11,1	11,1	6,7	6,7	9,9	9,9	34,8	36,4	55,2	65,4	68,5	41,5	59,7	63,2	9,8	30,4
	vencer	49,0	29,0	29,0	26,1	26,1	24,8	24,6	41,2	41,2	62,8	53,6	60,4	32,5	63,0	63,6	24,5	34,0

Cuadro D.11. 3fcv corpus entrenamiento SENSEVAL-2 español: F1 por grupos de atributos y palabras

D Cuadros adicionales

pos	palabra	0	b	B	c	C	d	D	l	L	o	p	P	r	s	S	v	W
N	autoridad	34,1	2,3	2,3	2,3	2,3	5,7	5,7	5,7	5,7	93,2	72,7	69,3	48,9	87,5	83,0	8,0	4,5
	bomba	100,0	25,0	25,0	19,7	19,7	11,8	11,8	35,5	35,5	97,4	100,0	100,0	61,8	98,7	98,7	14,5	22,4
	canal	38,3	28,7	28,7	24,3	24,3	15,7	17,4	42,6	42,6	71,3	58,3	0,9	12,2	75,7	73,9	14,8	30,4
	circuito	0,0	10,8	10,8	8,1	8,1	1,4	1,4	24,3	24,3	79,7	63,5	31,1	13,5	74,3	51,4	9,5	14,9
	corazón	98,0	9,1	9,1	8,1	8,1	20,2	23,2	30,3	31,3	91,9	87,9	94,9	46,5	88,9	84,8	17,2	21,2
	corona	100,0	5,1	5,1	5,1	5,1	2,5	2,5	12,7	12,7	87,3	83,5	93,7	60,8	89,9	93,7	5,1	11,4
	gracia	28,3	21,2	21,2	16,2	16,2	32,3	31,3	39,4	40,4	82,8	66,7	19,2	31,3	78,8	76,8	27,3	21,2
	grano	100,0	14,3	14,3	14,3	14,3	0,0	0,0	26,8	26,8	89,3	98,2	92,9	75,0	89,3	89,3	0,0	21,4
	hermano	74,4	2,6	2,6	2,6	2,6	9,0	12,8	10,3	10,3	83,3	94,9	98,7	52,6	84,6	85,9	6,4	5,1
	masa	83,3	14,4	14,4	14,4	14,4	23,3	23,3	27,8	27,8	82,2	86,7	54,4	31,1	83,3	80,0	13,3	21,1
	naturaleza	6,3	4,5	4,5	1,8	1,8	3,6	3,6	23,4	23,4	80,2	68,5	36,0	36,9	75,7	82,0	9,0	12,6
	operación	0,0	11,6	11,6	4,2	4,2	21,1	21,1	31,6	31,6	78,9	56,8	26,3	3,2	80,0	65,3	17,9	14,7
	órgano	67,9	13,7	13,7	10,7	10,7	4,6	6,9	31,3	32,1	92,4	88,5	100,0	38,9	94,7	90,1	6,9	22,1
	partido	79,4	17,6	17,6	11,8	11,8	13,7	13,7	33,3	33,3	97,1	100,0	100,0	68,6	100,0	100,0	22,5	15,7
	pasaje	26,8	5,6	5,6	2,8	2,8	8,5	8,5	8,5	11,3	74,6	74,6	77,5	64,8	74,6	77,5	11,3	8,5
	programa	25,3	12,6	14,7	14,7	14,7	1,1	0,0	29,5	26,3	77,9	77,9	94,7	38,9	84,2	88,4	16,8	16,8
	tabla	100,0	5,1	5,1	3,8	3,8	17,9	17,9	17,9	17,9	93,6	94,9	97,4	64,1	91,0	82,1	16,7	11,5
V	actuar	33,0	8,0	8,0	4,0	4,0	8,0	8,0	36,0	34,0	75,0	48,0	7,0	7,0	76,0	59,0	9,0	19,0
	apoyar	78,1	11,7	11,7	10,2	10,2	8,8	10,2	35,8	35,8	87,6	91,2	100,0	50,4	85,4	93,4	11,7	21,2
	apuntar	68,3	7,7	7,7	4,2	4,2	1,4	1,4	29,6	29,6	69,0	63,4	81,0	31,0	75,4	73,9	4,9	21,1
	clavar	63,2	13,8	13,8	11,5	11,5	8,0	8,0	29,9	28,7	72,4	71,3	79,3	18,4	73,6	72,4	10,3	21,8
	conducir	25,0	17,7	17,7	17,7	17,7	10,4	10,4	30,2	28,1	77,1	41,7	22,9	9,4	68,8	46,9	8,3	26,0
	copiar	25,5	3,2	3,2	0,0	0,0	9,6	9,6	34,0	33,0	70,2	40,4	5,3	14,9	79,8	73,4	9,6	11,7
	coronar	45,9	28,2	28,2	18,8	18,8	11,2	11,2	42,4	39,4	80,0	30,6	5,3	16,5	77,6	59,4	10,0	32,9
	explotar	32,6	10,9	10,9	6,5	6,5	7,6	7,6	38,0	41,3	76,1	59,8	22,8	13,0	73,9	54,3	7,6	29,3
	saltar	6,0	8,0	8,0	5,0	5,0	7,0	7,0	43,0	36,0	56,0	25,0	13,0	2,0	48,0	41,0	12,0	25,0
	tocar	29,0	24,7	23,5	19,1	18,5	6,8	7,4	50,6	44,4	72,8	35,2	0,0	25,3	71,0	69,1	6,8	29,0
	usar	81,1	13,5	13,5	7,2	7,2	9,0	9,0	39,6	38,7	82,9	95,5	100,0	47,7	90,1	88,3	10,8	24,3
	vencer	66,1	22,9	22,9	16,9	16,9	16,1	16,9	39,8	39,8	89,0	89,8	99,2	40,7	85,6	83,9	17,8	29,7

Cuadro D.12. 3fcv corpus entrenamiento SENSEVAL-2 español: cobertura absoluta por grupos de atributos y palabras

			R1			R2			R3			R4			F		
palabra	pos	total	P	C	F1	P	C	F1	P	C	F1	P	C	F1	P	C	F1
órgano	N	81	88,9	19,8	32,3	93,5	35,8	51,8	92,9	48,1	63,4	93,0	49,4	64,5	85,2	85,2	85,2
bomba	N	37	90,9	27,0	41,7	93,3	37,8	53,8	94,1	43,2	59,3	94,4	45,9	61,8	83,8	83,8	83,8
canal	N	41	100,0	29,3	45,3	100,0	41,5	58,6	89,5	41,5	56,7	89,5	41,5	56,7	82,9	82,9	82,9
corona	N	40	100,0	27,5	43,1	93,3	35,0	50,9	94,4	42,5	58,6	94,4	42,5	58,6	77,5	77,5	77,5
partido	N	57	87,5	12,3	21,5	94,1	28,1	43,2	85,2	40,4	54,8	85,7	42,1	56,5	77,2	77,2	77,2
gracia	N	61	90,9	16,4	27,8	89,5	27,9	42,5	92,6	41,0	56,8	92,6	41,0	56,8	77,0	77,0	77,0
tabla	N	41	100,0	17,1	29,2	90,0	22,0	35,3	80,0	29,3	42,9	75,0	29,3	42,1	70,7	70,7	70,7
hermano	N	57	100,0	8,8	16,1	78,6	19,3	31,0	72,7	28,1	40,5	73,9	29,8	42,5	70,2	70,2	70,2
naturaleza	N	56	85,7	10,7	19,0	81,8	16,1	26,9	83,3	17,9	29,4	83,3	17,9	29,4	64,3	64,3	64,3
circuito	N	49	100,0	14,3	25,0	92,3	24,5	38,7	81,0	34,7	48,6	81,0	34,7	48,6	61,2	61,2	61,2
masa	N	41	100,0	4,9	9,3	57,1	9,8	16,7	57,1	19,5	29,1	60,0	22,0	32,1	61,0	61,0	61,0
autoridad	N	34	50,0	2,9	5,6	83,3	14,7	25,0	77,8	20,6	32,6	77,8	20,6	32,6	58,8	58,8	58,8
corazón	N	47	66,7	8,5	15,1	77,8	29,8	43,1	69,6	34,0	45,7	69,6	34,0	45,7	55,3	55,3	55,3
grano	N	22	100,0	13,6	24,0	100,0	18,2	30,8	100,0	18,2	30,8	80,0	18,2	29,6	54,5	54,5	54,5
programa	N	47	100,0	6,4	12,0	87,5	14,9	25,5	66,7	17,0	27,1	66,7	17,0	27,1	53,2	53,2	53,2
pasaje	N	41	100,0	4,9	9,3	66,7	9,8	17,0	50,0	12,2	19,6	53,8	17,1	25,9	43,9	43,9	43,9
operación	N	47	57,1	8,5	14,8	43,8	14,9	22,2	42,9	19,1	26,5	42,9	19,1	26,5	38,3	38,3	38,3
vencer	V	65	80,0	12,3	21,3	77,8	21,5	33,7	79,2	29,2	42,7	79,2	29,2	42,7	76,9	76,9	76,9
apoyar	V	73	100,0	11,0	19,8	87,0	27,4	41,7	78,4	39,7	52,7	76,3	39,7	52,3	74,0	74,0	74,0
usar	V	56	50,0	1,8	3,4	83,3	17,9	29,4	56,5	23,2	32,9	56,5	23,2	32,9	71,4	71,4	71,4
apuntar	V	49	100,0	8,2	15,1	100,0	14,3	25,0	90,0	18,4	30,5	90,0	18,4	30,5	61,2	61,2	61,2
clavar	V	44	100,0	20,5	34,0	81,3	29,5	43,3	76,5	29,5	42,6	76,5	29,5	42,6	56,8	56,8	56,8
coronar	V	74	92,3	16,2	27,6	65,5	25,7	36,9	59,0	31,1	40,7	58,5	32,4	41,7	55,4	55,4	55,4
tocar	V	74	100,0	20,3	33,7	84,0	28,4	42,4	73,5	33,8	46,3	73,5	33,8	46,3	54,1	54,1	54,1
conducir	V	54	100,0	16,7	28,6	84,6	20,4	32,8	76,5	24,1	36,6	76,5	24,1	36,6	53,7	53,7	53,7
actuar	V	55	20,0	1,8	3,3	45,5	9,1	15,2	52,4	20,0	28,9	52,4	20,0	28,9	50,9	50,9	50,9
tratar	V	70	57,1	5,7	10,4	45,0	12,9	20,0	50,0	20,0	28,6	50,0	20,0	28,6	50,0	50,0	50,0
explotar	V	41	100,0	7,3	13,6	63,6	17,1	26,9	65,0	31,7	42,6	65,0	31,7	42,6	43,9	43,9	43,9
copiar	V	53	100,0	7,5	14,0	75,0	17,0	27,7	75,0	22,6	34,8	75,0	22,6	34,8	43,4	43,4	43,4
saltar	V	37	100,0	10,8	19,5	92,3	32,4	48,0	76,5	35,1	48,1	76,5	35,1	48,1	35,1	35,1	35,1
claro	A	66	100,0	21,2	35,0	100,0	43,9	61,1	97,6	60,6	74,8	97,6	60,6	74,8	90,9	90,9	90,9
local	A	55	100,0	5,5	10,3	100,0	29,1	45,1	87,5	38,2	53,2	87,5	38,2	53,2	87,3	87,3	87,3
popular	A	204	98,1	25,0	39,8	95,1	48,0	63,8	91,9	60,8	73,2	91,9	60,8	73,2	85,8	85,8	85,8
vital	A	79	100,0	16,5	28,3	88,6	39,2	54,4	87,2	51,9	65,1	87,5	53,2	66,1	79,7	79,7	79,7
simple	A	57	84,6	19,3	31,4	83,9	45,6	59,1	82,9	50,9	63,0	82,9	50,9	63,0	73,7	73,7	73,7
brillante	A	87	100,0	6,9	12,9	80,8	24,1	37,2	75,7	32,2	45,2	78,6	37,9	51,2	71,3	71,3	71,3
verde	A	33	100,0	12,1	21,6	85,7	36,4	51,1	85,7	36,4	51,1	85,7	36,4	51,1	69,7	69,7	69,7
ciego	A	42	100,0	16,7	28,6	92,3	28,6	43,6	93,3	33,3	49,1	93,8	35,7	51,7	66,7	66,7	66,7
natural	A	58	100,0	17,2	29,4	81,8	31,0	45,0	76,9	34,5	47,6	77,8	36,2	49,4	51,7	51,7	51,7

Cuadro D.13. Evaluación reentrenamiento secuencial sobre corpus test SENSEVAL-2 español: detalle por palabras

Bibliografía

ABNEY, STEVEN (2002). «Bootstrapping», en *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, págs. 360–367.

AGIRRE, E., O. ANSA, X. ARREGI, J. ARRIOLA, A. DÍAZ DE ILARRAZA, E. POCIELLO y L. URIA (2002). «Methodological issues in the building of the basque wordnet: quantitative and qualitative analysis», en *Proceedings of First International WordNet Conference*, Mysore, India.

AGIRRE, ENEKO y DAVID MARTÍNEZ (2000). «Exploring automatic word sense disambiguation with decision lists and the web», en *Proceedings of the Semantic Annotation and Intelligent Annotation workshop organized by COLING*, Luxembourg.

AGIRRE, ENEKO y DAVID MARTÍNEZ (2001). «Knowledge sources for word sense disambiguation», en Vaclav Matousek, Pavel Mautner, Roman Moucek y Karel Tauser, editores, *Proceedings of the Fourth International Conference TSD 2001*, Lecture Notes in Computer Science, Springer-Verlag, Plzen (Pilsen), Czech Republic.

AGIRRE, ENEKO y GERMAN RIGAU (1995). «A proposal for word sense disambiguation using Conceptual Distance», en *Proceedings of the International Conference Recent Advances in Natural Language Processing"(RANLP´95)*.

AGIRRE, ENEKO y GERMAN RIGAU (1996). «Word sense disambiguation using Conceptual Density», en *Proceedings of the 16th International Conference on Computational Linguistic (COLING´96*, Copenhagen, Denmark.

AHA, DAVID, DENNIS KIBLER y MARC ALBERT (1991). «Instance-based learning algorithms», *Machine Learning*, **6**(1), 37–66.

BIBLIOGRAFÍA

ARGAMON-ENDELSON, S. y I. DAGAN (1999). «A committee-based sample selection for probabilistic classifiers», *Journal of Artificial Intelligence Research*, **11**, 335–460.

ATSERIAS, JORDI y HORACIO RODRÍGUEZ (1998). «Tacat: Tagged corpus analizer tool», *inf. téc.*, LSI, Universidad Politécnica de Cataluña.

BAEZA-YATES, RICARDO (2004). «Challenges in the interaction of information retrieval and natural language processing», en Alexander Gelbukh, editor, *Proceedings of the Fifth International Conference CICling 2004*, Lecture Notes in Computer Science, págs. 445–456, Springer-Verlag Heidelberg, Seoul, Korea.

BAUM, L. E. (1972). «An inequality and associated maximization technique in statistical estimation for probabilistic functions of a markov process», *Inequalities*, págs. 1–8.

BENÍTEZ, L., S. CERVELL, G. ESCUDERO, M. LÓPEZ, G. RIGAU y M. TAULÉ (1998). «Methods and tools for building the catalan wordnet», en *Proceedings of ELRA Workshop on Language Resources for European Minority Languages*, Granada, Spain.

BENSON, STEVEN J. y JORGE J. MORÉ (2001). «A limited memory variable metric method for bound constrained minimization», *inf. téc.*, Argonne National Laboratory.

BERGER, ADAM L., STEPHEN A. DELLA PIETRA y VINCENT J. DELLA PIETRA (1996). «A maximum entropy approach to natural language processing», *Computational Linguistics*, **22**(1), 39–71.

BLUM, AVRIM y TOM MITCHELL (1998). «Combining labeled and unlabeled data with co-training», en *Proceedings of the 11th Annual Conference on Computational Learning Theory*, págs. 92–100, ACM Press, Madison, Wisconsin.

BORTHWICK, A., J. STERLING, E. AGICHTEIN y R. GRISHMAN (1998a). «Description of the MENE Named Entity System used for MUC-7», en *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

BORTHWICK, ANDREW, JOHN STERLING, EUGENE AGICHTEIN y RALPH GRISHMAN (1998b). «Exploiting diverse knowledge sources via maximum entropy in named entity recognition», en Eugene Charniak, editor, *Proceedings of the ACL's SIGDAT 6th Workshop*

on Very Large Corpora, págs. 152–160, Montreal, Canada.

BROWN, PETER F., STEPHEN A. DELLA PIETRA y VINCENT J. DELLA PIETRA (1991). «Word sense Disambiguation using statistical methods», en *29th Annual Meeting of the Association for Computational Linguistics*, págs. 264–270.

BRUCE, REBECCA y JANYCE WIEBE (1994). «Word sense disambiguation using decomposable models», en *Proceedings of the ACL-94, 32nd Annual Meeting of the Association for Computational Linguistics*, págs. 139–145, Las Cruces, US.

CABEZAS, CLARA, PHILIP RESNIK y JESSICA STEVENS (2001). «Supervised sense tagging using support vector machines», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 59–62, ACL-SIGLEX, Toulouse, France.

CARMONA, J., S. CERVELL, L. MÀRQUEZ, M. A. MARTÍ, L. PADRÓ, R. PLACER, H. RODRÍGUEZ, M. TAULÉ y J. TURMO (1998). «Morphosyntactic Analysis and Parsing of Unrestricted Spanish Text», en *Proceedings of First International Conference on Language Resources and Evaluation (LREC'98)*, Granada, Spain.

CHAO, GERALD y MICHAEL G. DYER (2002). «Maximum Entropy Models for Word Sense Disambiguation», en Hsin-Hsi Chen y Chin-Yew Lin, editores, *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, págs. 155–161, Taipei, Taiwan.

CHEN, STANLEY F. y RONALD ROSENFELD (1999). «A gaussian prior for smoothing maximum entropy models», *inf. téc.*, Carnegie Mellon Univesity, Pittsburgh, PA.

CHEN, STANLEY F. y RONALD ROSENFELD (2000). «A survey of smoothing techniques for maximum entropy models», *IEEE Transactions on Speech and Audio Processing*, **8**(1), 37–50.

CHKLOVSKI, TIMOTHY y RADA MIHALCEA (2002). «Building a sense tagged corpus with Open Mind Word Expert», en *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, págs. 116–122, Association for Computational Linguistics, Philadelphia.

BIBLIOGRAFÍA

COHN, DAVID, LES ATLAS y RICHARD LADNER (1994). «Improving generalization with active learning», *Machine Learning*, **15**(2), 201–221.

COLLINS, MICHAEL y YORAM SINGER (1999). «Unsupervised models for named entity classification», en Pascale Fung y Joe Zhou, editores, *Proceedings of 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, págs. 100–110, ACL, Maryland, USA.

COLLINS, MICHAEL y YORAM SINGER (2002). «Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods», *To appear as a book chapter. http://www.ai.mit.edu/people/mcollins/papers/book.ps.*

COTTRELL, GARRISON y STEVEN SMALL (1983). «A connectionist scheme for modelling word sense disambiguation», *Cognition and Brain Theory*, **6**, 89–120.

COWIE, JIM, JOE GUTHRIE y LOUISE GUTHRIE (1992). «Lexical disambiguation using simulated annealing», en *Proceedings of the 14th International Conference on Computational Linguistic, COLING´92*, págs. 359–365, Nantes, France.

CURRAN, JAMES y STEPHEN CLARK (2003). «Investigating GIS and smoothing for maximum entropy taggers», en *Proceedings of the 10th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL03)*, págs. 91–98, Budapest, Hungary.

DAGAN, IDO, FERNANDO PEREIRA y LILLIAN LEE (1994). «Word sense disambiguation using a second language monolingual corpus», *Computational Linguistics*, **20**(4), 563–596.

DARROCH, J.N. y D. RATCLIFF (1972). «Generalized Iterative Scaling for log-linear models», *The annals of mathematical statistics*, **43**(5), 1470–1480.

DAUDÉ, JORDI, LLUIS PADRO y GERMAN RIGAU (2000). «Mapping wordnets using structural information», en *Proceedings of the 38th Anual Meeting of the Association for Computational Linguistics (ACL'2000)*, Hong Kong.

DEHASPE, L. (1997). «Maximum entropy modeling with clausal constraints», en S. Džeroski y N. Lavrač, editores, *Proceedings of the*

7th International Workshop on Inductive Logic Programming, vol. 1297, págs. 109–124, Springer-Verlag.

DIAB, MONA y PHILIP RESNIK (2002). «An unsupervised method for word sense tagging using parallel corpora», en *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, págs. 255–262.

DIAB, MONA TALAT (2003). *Word Sense Disambiguation within a Multiligual Framework*, Tesis Doctoral, University of Maryland.

DIETTERICH, THOMAS G. (1998). «Approximate statistical test for comparing supervised classification learning algorithms», *Neural Computation*, **10**(7), 1895–1923.

DIETTERICH, THOMAS G. y GHULUM BAKIRI (1995). «Solving multiclass learning problems via error-correcting output codes», *Journal of Artificial Intelligence Research*, **2**, 263–286.

EDMONDS, PHIL (2001). «Senseval-2: Overview», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 1–5, ACL-SIGLEX, Toulouse, France.

ESCUDERO, G., L. MÀRQUEZ y G. RIGAU (2000a). «Boosting Applied to Word Sense Disambiguation», en *Proceedings of the 11th European Conference on Machine Learning, ECML-2000*, Barcelona, Spain.

ESCUDERO, GERARD, LLUÍS MÀRQUEZ y GERMAN RIGAU (2000b). «A comparison between supervised learning algorithms for word sense disambiguation», en Claire Cardie, Walter Daelemans, Claire Nedellec y Erik Tjong Kim Sang, editores, *Proceedings of CoNLL-2000 and LLL-2000*, págs. 31–36, Lisbon, Portugal.

ESCUDERO, GERARD, LLUÍS MÀRQUEZ y GERMAN RIGAU (2000c). «On the portability and tuning of supervised word sense disambiguation systems», en Hinrich Schütze y Keh-Yih Su, editores, *Proceedings of the Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, Hong Kong, China.

ESCUDERO, GERARD, LLUÍS MÀRQUEZ y GERMAN RIGAU (2001). «Using LazyBoosting for Word Sense Disambiguation», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd Interna-*

*tional Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 71–76, ACL-SIGLEX, Toulouse, France.

FELLBAUM, CHRISTIANE (1998). *WordNet, an electronic lexical database*, MIT Press.

FERNÁNDEZ-AMORÓS, DAVID, JULIO GONZALO y FELISA VERDEJO (2001). «The UNED systems at SENSEVAL-2», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 75–78, ACL-SIGLEX, Toulouse, France.

FLORIAN, RADU y DAVID YAROWSKY (2002). «Modeling consensus: Classifier combination for word sense disambiguation», en *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, págs. 25–32, Association for Computational Linguistics, Philadelphia.

FUJII, A., K. INUI, T. TOKUNAGA y H. TANAKA (1998). «Selective sampling for example-based word sense disambiguation», *Computational Linguistics*, **24**(4), 573–598.

GALE, WILLIAM, KENNETH CHURCH y DAVID YAROWSKY (1992a). «One sense per discourse», en Morgan Kaufmann, editor, *Proceedings of the speech and Natural Language Workshop*, págs. 233–237, San Francisco, USA.

GALE, WILLIAM A., KENNETH W. CHURCH y DAVID YAROWSKY (1992b). «One sense per discourse», en *Proceedings of the ARPA Workshop on Speech and Natural Language Processing*, págs. 233–237.

GALE, WILLIAM A., KENNETH W. CHURCH y DAVID YAROWSKY (1993). «A method for disambiguating word senses in a large corpus», *Computers and the Humanities*, **26**(5), 415–439.

GARCÍA-VEGA, MANUEL, MARÍA TERESA MARTÍN-VALDIVIA y LUIS ALFONSO UREÑA (2003). «Aprendizaje competitivo lvq para la desambiguación léxica», *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural*, **31**, 125–132.

GARCÍA-VAREA, ISMAEL, FRANZ J. OCH, HERMANN NEY y FRANCISCO CASACUBERTA (2001). «Refined lexicon models for statistical machine translation using a maximum entropy approach», en *Pro-*

ceedings of 39th Annual Meeting of the Association for Computational Linguistics, págs. 204–211.

GHANI, RAYID (2001). *Using error-correcting codes for efficient text classification with a large number of categories*, Tesis Doctoral, Center for Automated Learning and Discovery, Carnegie Mellon University.

GONZALO, JULIO, IRINA CHUGUR y FELISA VERDEJO (2003). «Resources for word sense disambiguation», *Word Sense Disambiguation: Algorithms, applications and Trends*, eds. Eneko Agirre and Phil Edmonds, Kluwer AP (to appear).

GOODMAN, JOSHUA (2001). «Classes for fast maximum entropy training», en *Proceedings of ICASSP-2001*, Utah, USA.

GOODMAN, JOSHUA (2002). «Sequential conditional generalized iterative scaling», en *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, págs. 9–16.

HOSTE, VERONIQUE, WALTER DAELEMANS, IRIS HENDRICKX y ANTAL VAN DEN BOSCH (2002). «Evaluating the results of a memory-based word-expert approach to unrestricted word sense disambiguation», en *Proceedings of the SIGLEX/SENSEVAL Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, págs. 95–101, Association for Computational Linguistics, Philadelphia.

HOSTE, VÉRONIQUE, ANNE KOLL y WALTER DAELEMANS (2001). «Classifier optimization and combination in the English all words task», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 83–86, ACL-SIGLEX, Toulouse, France.

HWA, REBECCA (2000). «Sample selection for statistical grammar induction», en *Proceedings of the 2000 Joint SIGDAT Conference on EMNLP and VLC*, págs. 45–52, Hong Kong, China.

IDE, N. y J. VÉRONIS (1998). «Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art», *Computational Linguistics*, **24**(1), 1–40.

ILHAN, H. TOLGA, SEPANDAR D. KAMVAR, DAN KLEIN, CHRISTOPHER D. MANNING y KRISTINA TOUTANOVA (2001). «Combining He-

terogeneous Classifiers for Word-Sense Disambiguation», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 87–90, ACL-SIGLEX, Toulouse, France.

ITTYCHERIAH, ABRAHAM, MARTIN FRANZ, WEI-JING ZHU y ADWAIT RATNAPARKI (2000). «IBM's statistical question answering system», en *9th Text REtrieval Conference*, Gaithersburg, MD, 2000.

ITTYCHERIAH, ABRAHAM, MARTIN FRANZ, WEI-JING ZHU, ADWAIT RATNAPARKI y RICHARD J. MAMMONE (2001). «Question answering using maximum entropy components», en *Proceedings of the NAACL Conference*, págs. 33–39, Pittsburgh, PA.

JAYNES, E.T. (1990). «Notes on present status and future prospects», en W.T. Grandy y L.H. Schick, editores, *Maximum Entropy and Bayesian Methods*, págs. 1–13, Kluwer.

JOACHIMS, T. (1998). «Text catogorization with support vector machines», en *Proceedings of the 11th European Conference on Machine Learning, ECML'98*, págs. 3–12, Chemnitz, Germany.

KAZAMA, JUN'ICHI y JUN'ICHI TSUJII (2003). «Evaluation and extension of maximum entropy models with inequality constraints», en Michael Collins y Mark Steedman, editores, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, págs. 137–144.

KHUDANPUR, S. (1995). «A method of ME estimation with relaxed constraints», en *Proceedings of the Johns Hopkins University Language Modeling Workshop*, págs. 1–17.

KILGARRIFF, A. y J ROSENZWEIG (2000). «Framework and results for English SENSEVAL», *Computers and the Humanities*, **34**(1-2), 15–48.

KILGARRIFF, ADAM y MARTHA PALMER (2000). «Introduction to the Special Issue on SENSEVAL.», *Computers and the Humanities*, **34**(1-2), 1–13.

KUDO, T. y Y. MATSUMOTO (2001). «Chunking with support vector machines», en Preiss y Yarowsky (2001a).

KUČERA, HENRY y W.ÑELSON FRANCIS (1997). «The standard corpus of present-day edited American English (The Brown Corpus)»,

Electronic Database. Providence, Rode Island (last revision and amplification 1979). Brown University.

LAFFERTY, JOHN, ANDREW MCCALLUM y FERNANDO PEREIRA (2001). «Conditional random fields: Probabilistic models for segmenting and labeling sequence data», en *Proc. 18th International Conf. on Machine Learning*, págs. 282–289, Morgan Kaufmann, San Francisco, CA.

LAU, R., R. ROSENFELD y S. ROUKOS (1993). «Adaptative statistical language modeling using the maximum entropy principle», en *Proceedings of the Human Language Technology Workshop, ARPA.*

LAU, RAYMOND (1994). *Adaptative statistical language modeling*, proyecto fin de carrera, MIT.

LEACOCK, CLAUDIA, M. CHODOROW y G. MILLER (1998). «Using corpus statistics and WordNet relations for sense identification», *Computational Linguistics*, **1**(24), 147–165.

LEACOCK, CLAUDIA, GEOFFREY TOWELL y ELLEN VOORHEES (1993). «Corpus based statistic sense resolution», en *Proceedings of the ARPA Workshop on Human Language Technology*, págs. 260–265, San Francisco, Morgan Kaufman.

LEE, YOONG KEOK y HWEE TOU NG (2002). «An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation», en *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, págs. 41–48, Association for Computational Linguistics, Philadelphia.

LEWIS, D. y D. GALE (1994). «Training text classifiers by uncertainty sampling», en *Proceedings of the International ACM Conference on Research and Development in Information Retrieval*, págs. 3–12.

LIN, DEKANG (1997). «Using syntactic dependency as local context to resolve word sense ambiguity», en *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistic and the 8th Conference of the European Chapter of the Association for Computational Linguistics*, págs. 64–71.

LIN, DEKANG (1998). «Dependency-based evaluation of minipar», en *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*, Granada, Spain.

BIBLIOGRAFÍA

MAGNINI, BERNARDO y GABRIELA CAVAGLIA (2000). «Integrating Subject Field Codes into WordNet», en M. Gavrilidou, G. Crayannis, S. Markantonatu, S. Piperidis y G. Stainhaouer, editores, *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*, págs. 1413–1418, Athens, Greece.

MAGNINI, BERNARDO y C. STRAPPARAVA (2000). «Experiments in Word Domain Disambiguation for Parallel Texts», en *Proceedings of the ACL Workshop on Word Senses and Multilinguality*, Hong Kong, China.

MAGNINI, BERNARDO, CARLO STRAPPARAVA, GIOVANNI PEZZULO y ALFIO GLIOZZO (2001). «Using Domain Information for Word Sense Disambiguation», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 111–114, ACL-SIGLEX, Toulouse, France.

MALOUF, ROBERT (2002). «A comparison of algorithms for maximum entropy parameter estimation», en *Proceedings of CoNLL-2002*, págs. 49–55, Taipei, Taiwan.

MANNING, CHRISTOPHER D. y HINRICH SCHÜTZE (1999). *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, Massachusetts.

MARKOV, A. (1913). «An example of statistical investigation in the text of 'eugene onyegin' illustrating coupling of 'tests' in chains», en *Proceedings of the Academy of Science*, vol. 7, págs. 152–162, St. Petersburg.

MÀRQUEZ, LLUÍS (2000). «Machine learning and natural language processing», *inf. téc.*, Centre de recerca TALP, Departament de Llenguatges i Sistemes Informàtics, U. Politècnica de Catalunya.

MÀRQUEZ, LLUÍS, FCO. JAVIER RAYA, JOHN CARROLL, DIANA MCCARTHY, ENEKO AGIRRE, DAVID MARTÍNEZ, CARLO STRAPPARAVA y ALFIO GLIOZZO (2003). «Experiment A: several all-words WSD systems for English», *inf. téc. WP6.2*, MEANING project (IST-2001-34460), http://www.lsi.upc.es/~nlp/meaning/meaning.html.

MARTÍNEZ, DAVID y ENEKO AGIRRE (2000). «One sense per collocation and genre/topic variations», en Hinrich Schütze y Keh-Yih Su, editores, *Proceedings of the Joint Sigdat Conference on Empirical*

Methods in Natural Language Processing and Very Large Corpora, Hong Kong, China.

MARTÍNEZ, DAVID y ENEKO AGIRRE (2001). «Decision Lists for English and Basque», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 115–118, ACL-SIGLEX, Toulouse, France.

MARTÍNEZ, DAVID, ENEKO AGIRRE y LLUÍS MÀRQUEZ (2002). «Syntactic features for high precision Word Sense Disambiguation», en Hsin-Hsi Chen y Chin-Yew Lin, editores, *Proceedings of the 19th International Conference on Computational Linguistics*, págs. 626–632, Taipei, Taiwan.

MCCALLUM, ANDREW, DAYNE FREITAG y FERNANDO PEREIRA (2000). «Maximum Entropy Markov Models for Information Extraction and Segmentation», en *Machine Learning: Proceedings of the Seventeenth International Conference (ICML 2000)*, págs. 591–598, Stanford, California.

MCCARTHY, DIANA, JOHN CARROL y JUDITA PREISS (2001). «Disambiguating Noun and Verb Senses Using Automatically Acquired Selectional Preferences», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 119–122, ACL-SIGLEX, Toulouse, France.

MCROY, SUSAN (1992). «Using multiple knowledge sources for word sense discrimination», *Computational Linguistics*, **18**(1), 1–30.

MIHALCEA, RADA (2002). «Instance based learning with automatic feature selection applied to word sense disambiguation», en Hsin-Hsi Chen y Chin-Yew Lin, editores, *Proceedings of the 19th International Conference on Computational Linguistics*, págs. 660–666, Taipei, Taiwan.

MIHALCEA, RADA (2003). «Unsupervised natural language disambiguation: the role of non-ambiguous words», en *Proceedings of the RANLP'03*.

MIHALCEA, RADA y DAN MOLDOVAN (1999). «A method for word sense disambiguation of unrestricted text», en *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistic*,

págs. 152–158, College Park, Maryland, USA.

MIHALCEA, RADA y DAN MOLDOVAN (2000). «An iterative approach to word sense disambiguation», en *Proceedings of FLAIRS-2000*, págs. 219–223, Orlando, FL.

MIHALCEA, RADA y DAN MOLDOVAN (2001a). «Automatic generation of a coarse grained WordNet», en *Proceedings of the SIGLEX workshop "WordNet and Other Lexical Resources: Applications, Extensions and Customizations" held in conjunction with NAACL*, Pittsburgh, USA.

MIHALCEA, RADA y DAN MOLDOVAN (2001b). «eXtended WordNet: Progress report», en Preiss y Yarowsky (2001a), págs. 95–100.

MIHALCEA, RADA F. y DAN I. MOLDOVAN (2001c). «Pattern Learning and Active Feature Selection for Word Sense Disambiguation», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 127–130, ACL-SIGLEX, Toulouse, France.

MIKHEEV, ANDREI (1998). «Feature lattices for maximum entropy modelling», en ACL, editor, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, págs. 848–854, Montreal, Canada.

MILLER, GEORGE A., RICHARD BECKWITH, CHRISTIANE FELLBAUM, DEREK GROSS y KATHERINE MILLER (1990). «WordNet: An on-line lexical database», *International journal of lexicography*, **3**(4), 235–244.

MOLINA, ANTONIO (2004). *Desambiguación en Procesamiento del Lenguaje Natural mediante Técnicas de Aprendizaje Automático*, Tesis Doctoral, Universidad Politécnica de Valencia.

MONTOYO, ANDRÉS y MANUEL PALOMAR (2000). «Word Sense Disambiguation with Specification Marks in Unrestricted Texts.», en *Proceedings of 11th International Workshop on Database and Expert Systems Applications (DEXA 2000)*, págs. 103–107, IEEE Computer Society, Greenwich, London, UK.

MONTOYO, ANDRÉS y MANUEL PALOMAR (2001). «Specification Marks for Word Sense Disambiguation: New Development», en Ale-

xander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, Second International Conference, CICLing 2001*, vol. 2004 de *Lecture Notes in Computer Science*, págs. 182–191, Springer, Mexico-City, Mexico.

MONTOYO, ANDRÉS, MANUEL PALOMAR y GERMAN RIGAU (2001). «Wordnet enrichment with classification systems», en *Proceedings of WordNet and Other Lexical Resources: Applications, Extensions and Customisations Workshop. The Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, págs. 101–106, Carnegie Mellon University. Pittsburgh, PA, USA.

MONTOYO, ANDRÉS y ARMANDO SUÁREZ (2001). «The University of Alicante word sense disambiguation system», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 131–134, ACL-SIGLEX, Toulouse, France.

MONTOYO, ANDRÉS, ARMANDO SUÁREZ y MANUEL PALOMAR (2002). «Combining supervised-unsupervised methods for word sense disambiguation», en Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002*, vol. 2276 de *Lecture Notes in Computer Science*, págs. 156–164, Springer, Mexico City, Mexico.

MONTOYO, ANDRÉS (2002). *Desambiguación léxica mediante Marcas de Especificidad*, Tesis Doctoral, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante.

MONTOYO, ANDRÉS, SONIA VAZQUEZ y GERMAN RIGAU (2003). «Método de desambiguación léxica basada en el recurso léxico Dominios Relevantes», *Procesamiento del Lenguaje Natural*, **30**.

MOONEY, RAYMOND J. (1996). «Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning», en Eric Brill y Kenneth Church, editores, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, págs. 82–91, Association for Computational Linguistics, Somerset, New Jersey.

MOONEY, RAYMOND J. (2003). *Machine Learning*, cap. 20, págs. 376–394, Oxford Handbook of Computational Linguistics, Oxford

University Press, Ruslan Mitkov ed.

MURATA, MASAKI, MASAO UTIYAMA, KIYOTAKA UCHIMOTO, QING MA y HITOSHI ISAHARA (2001). «Japanese word sense disambiguation using the simple Bayes and support vector machine methods», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 135–138, ACL-SIGLEX, Toulouse, France.

NAKAGAWA, TETSUJI, TAKU KUDO y YUJI MATSUMOTO (2002). «Revision learning and its application to part-of-speech tagging», en *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, págs. 497–504.

NEWMANN, W. (1977). «Extension to ME method», en *IEEE Trans. on Information Theory*, vol. IT-23, págs. 89–93.

NG, HWEE TOU (1997). «Getting Serious about Word Sense Disambiguation», en *Proceedings of the SIGLEX Workshop "Tagging Text with Lexical Semantics: Why, What and How?*

NG, HWEE TOU y HIANG BENG LEE (1996). «Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach», en *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistic*, págs. 40–47, University of California, Santa Cruz, CA.

NG, HWEE TOU, BIN WANG y YEE SENG CHAN (2003). «Exploiting parallel texts for word sense disambiguation: An empirical study», en Erhard Hinrichs y Dan Roth, editores, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, págs. 455–462.

NG, VINCENT y CLAIRE CARDIE (2003). «Weakly supervised natural language learning without redundant views», en Marti Hearst y Mari Ostendorf, editores, *HLT-NAACL 2003: Main Proceedings*, págs. 173–180, Association for Computational Linguistics, Edmonton, Alberta, Canada.

NICA, IULIA, Mª ANTONIA MARTÍ y ANDRÉS MONTOYO (2004a). «Colaboración entre información paradigmática y sintagmática en la desambiguación semántica automática», *Procesamiento del Lenguaje Natural*, **31**, 133–140.

NICA, IULIA, Mª ANTONIA MARTÍ, ANDRÉS MONTOYO y SONIA VAZQUEZ (2004b). «Combining EWN and Sense-Untagged Corpus for WSD», en A. Gelbukh, editor, *Proceedings of 5th International conference on Intelligent Text Processing and Computational Linguistics (CICLing-2004). Computational Linguistics and Intelligent Text Processing*, Lecture Notes in Computer Science, págs. 188–200, Springer-Verlag, Seul, Corea del Sur.

NIGAM, KAMAL y RAYID GHANI (2000a). «Analyzing the effectiveness and applicability of co-training», en *Proceedings of the 9th International Conference on Information and Knowledge Management*, págs. 86–93, ACM Press.

NIGAM, KAMAL y RAYID GHANI (2000b). «Understanding the behavior of co-training», en *Proceedings of the 6th ACM SIGKDD Internation Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*, págs. 105–106.

NIGAM, KAMAL, JOHN LAFFERTY y ANDREW MCCALLUM (1999). «Using maximum entropy for text classification», en *Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, págs. 61–67.

OCH, FRANZ JOSEF y HERMANN NEY (2002). «Discriminative training and maximum entropy models for statistical machine translation», en *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, págs. 295–302.

PALMER, MARTHA, HOA DANG y CHRISTIANE FELLBAUM (2004). «Making fine-grained and coarse-grained sense distinctions, both manually and automatically», *Natural Language Engineering*, to appear.

PEDERSEN, TED (2001a). «A decision tree of bigrams is an accurate predictor of word sense», en *Proceedings of the Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, págs. 79–86, Pittsburgh.

PEDERSEN, TED (2001b). «Machine Learning with Lexical Features: The Duluth Approach to SENSEVAL-2», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 139–142, ACL-SIGLEX, Toulouse, France.

BIBLIOGRAFÍA

PEDERSEN, TED (2002a). «Assessing System Agreement and Instance Difficulty in the Lexical Sample Tasks of Senseval-2», en *Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia, USA.

PEDERSEN, TED (2002b). «A baseline methodology for word sense disambiguation», en Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, Third International Conference, CICLing 2002*, vol. 2276 de *Lecture Notes in Computer Science*, págs. 126–135, Springer, Mexico City, Mexico.

PHILIPS, WILLIAMS y ELLEN RILOFF (2002). «Exploiting strong syntactic heuristics and co-training to learn semantic lexicons», en *Proceedings of Conference on Empirical Methods in Natural Language Processing EMNLP'02*.

PIERCE, DAVID y CLAIRE CARDIE (2001). «Limitations of co-training for natural language learning from large datasets», en *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

PLA, F. (2000). *Etiquetado Léxico y Análisis Sintáctico Superficial basado en Modelos Estadísticos*, Tesis doctoral, Departamento de Sistemas Informáticos y Computación. Universidad de Politécnica de Valencia.

PREISS, J. y D. YAROWSKY, editores (2001a). *Proceedings of NAACL 2001*, Pittsburgh, PA, USA.

PREISS, JUDITA (2001). «Local versus global context for WSD of nouns», en *Proceedings of the 4th Annual CLUK Research Colloquium*, págs. 1–8, University of Sheffield.

PREISS, JUDITA y DAVID YAROWSKY, editores (2001b). *Proceedings of SENSEVAL-2*, Toulouse, France, ACL-SIGLEX.

RATNAPARKHI, ADWAIT (1998). *Maximum Entropy Models for Natural Language Ambiguity Resolution*, Tesis Doctoral, University of Pennsylvania.

RESNIK, PHILIP y DAVID YAROWSKY (1999). «Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation», *Natural Language Engineering*, **5**(2), 113–134.

RIGAU, GERMAN, BERNARDO MAGNINI, ENEKO AGIRRE, PIEK VOSSEN y JOHN CARROL (2002). «Meaning: a roadmap to knowledge technologies», en *Proceedings of COLING Workshop "A Roadmap for Computational Linguistics"*, Taipei, Taiwan.

ROSENFELD, R. (1997). «A whole sentence maximum entropy language model», en *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding.*

ROSENFELD, RONALD (1996). «A Maximum Entropy Approach to Adaptive Statistical Language Modeling», *Computer, Speech and Language*, **10**, 187– 228, longer version: Carnegie Mellon Tech. Rep. CMU-CS-94-138.

SAIZ-NOEDA, MAXIMILIANO, ARMANDO SUÁREZ y MANUEL PALOMAR (2001). «Semantic pattern learning through maximum entropy-based wsd technique», en *Proceedings of CoNLL-2001*, págs. 23–29, Toulouse, France.

SAMPSON, GEOFFREY (1995). *English for the Computer: The Susanne Corpus and Analytic Scheme*, Clarendon Press, Oxford.

SANTAMARÍA, C., JULIO GONZALO y FELISA VERDEJO (2003). «Automatic association of web directories to word senses», *Computational Linguistics, Special Issue on the Web as Corpus*, **29**(3), 485–502.

SARKAR, ANOOP (2001). «Applying co-training methods to statistical parsing», en *Proceedings of the 2nd Annual Meeting of the NAACL*, págs. 95–102, Pittsburgh, PA.

SCHAPIRE, E. y Y. SINGER (1999). «Improved boosting algorithms using confidence-rated predictions», *Machine Learning*, **37**(3), 297–336.

SCHMID, HELMUT (1994). «Probabilistic part-of-speech tagging using decision trees», en *Proceedings International Conference on New Methods in Language Processing.*, págs. 44–49, Manchester, UK.

SCHMID, HELMUT (1995). «Improvements in part-of-speech tagging with an application to german», en Feldweg y Hinrichs, editores, *EACL SIGDAT Workshop*, págs. 47–50.

SEO, HEE-CHEOL, SANG-ZOO LEE, HAE-CHANG RIM y HO LEE (2001). «KUNLP system using Classification Information Model at

SENSEVAL-2», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 147–150, ACL-SIGLEX, Toulouse, France.

SLATOR, B. M. y Y. WILKS (1987). «Towards semantic structures from dictionary entries», en *Proceedings of the 2nd annual rocky mountain conference on Artificial Intelligence*, págs. 85–96.

STEEDMAN, MARK, REBECCA HWA, STEPHEN CLARK, MILES OSBORNE, ANOOP SARKAR, JULIA HOCKENMAIER, PAUL RUHLEN, STEVEN BAKER y JEREMIAH CRIM (2003). «Example selection for bootstrapping statistical parsers», en Marti Hearst y Mari Ostendorf, editores, *HLT-NAACL 2003: Main Proceedings*, págs. 236–243, Association for Computational Linguistics, Edmonton, Alberta, Canada.

SUÁREZ, ARMANDO y MANUEL PALOMAR (2002a). «Desambiguación del sentido y del dominio de las palabras con modelos de probabilidad de máxima entropía», *Procesamiento Lenguaje Natural*, **28**(1), 45–54.

SUÁREZ, ARMANDO y MANUEL PALOMAR (2002b). «A maximum entropy-based word sense disambiguation system», en Hsin-Hsi Chen y Chin-Yew Lin, editores, *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002*, págs. 960–966, Taipei, Taiwan.

SUTTON, CHARLES, KHASHAYAR ROHANIMANESH y ANDREW MCCALLUM (2004). «Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data», en *ICML 2004*.

TANG, MIN, XIAOQIANG LUO y SALIM ROUKOS (2002). «Active learning for statistical natural language parsing», en *Proceedings of the 40th Annual Meeting of the ACL*, págs. 120–127.

TAPANAINEN, PASI y TIMO JÄRVINEN (1997). «A non-projective dependency parser», en *Proceedings of the Fifth Conference on Applied Natural Language Processing*, págs. 64–71.

THOMPSON, CYNTHIA A., MARY ELAINE CALIFF y RAYMOND J. MOONEY (1999). «Active learning for natural language parsing and information extraction», en *Proceedings of ICML-99*, págs. 406–414,

Bled, Slovenia.

VAN HALTEREN, H., J. ZAVREL y W. DAELEMANS (2001). «Improving accuracy in wordclass tagging through combination of machine learning systems», *Computational Linguistics*, **27**(2), 199–230.

VAPNIK, VLADIMIR (1995). *The Nature of Statistical Learning Theory*, Springer Verlag.

VAPNIK, VLADIMIR (1998). *Statistical Learning Theory*, John Wiley and Sons, inc.

VERONIS, JEAN y NANCY IDE (1990). «Word Sense Disambiguation with very large neural networks extracted from machine readable dictionaries», en *Proceedings of the 13th International Conference on Computational Linguistics, COLING´90,volume 2*, págs. 389–394, Helsinki, Finland.

VOORHEES, E. M. y LORI P. BUCKLAND, editores (2002). *NIST Special Publication: SP 500-251, The 11th Text Retrieval Conference (TREC 2002)*, Department of Commerce, National Institute of Standards and Technology.

VOSER, B., I. GUYON y V. VAPNIK (1992). «A training algorithm for optimal margin classifiers», en *Proceedings of 5th Annual Workshop on Computational Learning Theory, CoLT'92*, ACM Press.

VOSSEN, P., L. BLOKSMA, H. RODRIGUEZ, S. CLIMENT, N. CALZOLARI, A. ROVENTINI, F. BERTAGNA, A. ALONGE y W. PETERS (1997). «The eurowordnet base concepts and top ontology», *Deliverable d017, d034, d036, eurowordnet (le 4003)*, University of Amsterdam.

VOSSEN, P., P., L. BLOKSMA, S. CLIMENT, M. ANTONIA MARTI, G. OREGGIONI, G. ESCUDERO, G. RIGAU, H. RODRIGUEZ, A. ROVENTINI, F. BERTAGNA, A. ALONGE, C. PETERS y W. PETERS (1998). «The Restructured Core wordnets in EuroWordNet: Subset1. EuroWordNet», en *Deliverable D014, D015, WP3, WP4*, EuroWordNet LE2-4003.

VOSSEN, PIEK (1998). «EuroWordNet: Building a Multilingual Database with WordNets for European Languages», *The ELRA Newsletter*, **3**(1).

WILKS, Y. y M. STEVENSON (1996). «The grammar of sense: Is word sense tagging much more than part-of-speech tagging?», *inf. téc.*,

CS-96-05, University of Sheffield. UK.

WILKS, Y. y M. STEVENSON (1997). «Sense Tagging: Semantic Tagging with a Lexicon», en *SIGLEX Workshop 'Tagging Text with Lexical Semantics'*, págs. 74–78, Washington DC.

WU, JUN y SANJEEV KHUDANPUR (2000). «Efficient training methods for maximum entropy language modeling», en *Proceedings of ICSLP2000*, vol. 3, págs. 114–117, Beijing, China.

YAROWSKY, DAVID (1992). «Word sense disambiguation using statistical models of Roget´s categories trained on large corpora», en *Proceedings of the 14th International Conference on Computational Linguistic, COLING´92*, págs. 454–460, Nantes, France.

YAROWSKY, DAVID (1994). «Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French», en *Proceedings of the 32th Annual Meeting of the Association for Computational Linguistic*, págs. 88–95.

YAROWSKY, DAVID (1995). «Unsupervised word sense disambiguation rivaling supervised methods», en *Proceedings of the 33th Annual Meeting of the Association for Computational Linguistic*, págs. 189–196.

YAROWSKY, DAVID (2000). «Hierarchical decision lists for word sense disambiguation», *Computers and the Humanities*, **34**(2), 179–186.

YAROWSKY, DAVID, SILVIU CUCERZAN, RADU FLORIAN, CHARLES SCHAFER y RICHARD WICENTOWSKI (2001). «The Johns Hopkins SENSEVAL-2 System Description», en Judita Preiss y David Yarowsky, editores, *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems (*SENSEVAL-2*)*, págs. 163–166, ACL-SIGLEX, Toulouse, France.

ZHOU, YAQUIAN, FULIANG WENG, LIDE WU y HAUKE SCHMIDT (2003). «A fast algorithm for feature selection in conditional maximum entropy modeling», en *Proceedings of the 10th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL03)*, págs. 153–159, Budapest, Hungary.

Volver al índice/Tornar a l'índex